IBM® Information Management Software

Reliability and Performance with IBM DB2 Analytics Accelerator Version 4.1

Discover ways to extend your investment in IBM System z for business analytics

Learn how to transparently accelerate complex DB2 queries

Prepare to implement highly available analytics

Paolo Bruni
Jason Arnold
Leticia Cruz
Jeff Feinsmith
Willie Favero
Anna Griner
James Guo
Chris Harlander
Johannes Kern
Ravi Kumar
Ruiping Li
Andy Perkins
Jonathan Sloan
Steve Speller
Dino Tonelli

Redbooks

ibm.com/redbooks

International Technical Support Organization

Reliability and Performance with IBM DB2 Analytics Accelerator Version 4.1

September 2014

SG24-8213-00

Note: Before using this information and the product it supports, read the information in “Notices” on page xvii.

First Edition (September 2014)

This edition applies to Version 4, Release 1 of DB2 Analytics Accelerator for z/OS (product number 5697-DAB) and Version 1 Release 1 of IBM DB2 Analytics Accelerator Loader for z/OS (product number 5639-OLA) for use with Version 11, Release 1 of DB2 for z/OS (program number 5615-DB2) and Version 11, Release 1 of DB2 Utilities Suite for z/OS (program number 5655-W87).

Contents

Tables

Examples

Figures

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information about the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the following website:

http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

CICS®
Cognos®
DataStage®
DB2®
Distributed Relational Database Architecture™
DRDA®
Extreme Blue®
FlashCopy®
GDPS®
Geographically Dispersed Parallel Sysplex™
IBM®
IBM PureData™
IMS™
InfoSphere®
OMEGAMON®
Optim™
Parallel Sysplex®
PureData™
pureXML®
QMF™
Query Management Facility™
RACF®
Redbooks®
Redpaper™
Redbooks (logo) ®
SPSS®
System Storage®
System z®
Tivoli®
TM1®
z/OS®
z/VM®
zEnterprise®

The following terms are trademarks of other companies:

Netezza, Netezza Performance Server, NPS, and N logo are trademarks or registered trademarks of IBM International Group B.V., an IBM Company.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

Preface

The IBM® DB2® Analytics Accelerator for IBM z/OS® is a high-performance appliance that integrates the IBM zEnterprise® infrastructure with IBM PureData™ for Analytics, powered by IBM Netezza® technology. With this integration, you can accelerate data-intensive and complex queries in a DB2 for z/OS highly secure and available environment.

DB2 and the Analytics Accelerator appliance form a self-managing hybrid environment running online transaction processing and online transactional analytical processing concurrently and efficiently. These online transactions run together with business intelligence and online analytic processing workloads.

DB2 Analytics Accelerator V4.1 expands the value of high-performance analytics. DB2 Analytics Accelerator V4.1 opens to static Structured Query Language (SQL) applications and row set processing, minimizes data movement, reduces latency, and improves availability.

This IBM Redbooks® publication provides technical decision-makers with an understanding of the benefits of version 4.1 of the Analytics Accelerator with DB2 11 for z/OS. It describes the installation of the new functions, and the advantages to existing analytical processes as measured in our test environment. This book also introduces the DB2 Analytics Accelerator Loader V1.1, a tool that facilitates the data population of the DB2 Analytics Accelerator.

Authors

This book was produced by a team of specialists from around the world working at the System z Business Analytics Performance Integration Center, Poughkeepsie, NY.

Paolo Bruni is a DB2 Information Management Project Leader at the International Technical Support Organization based in the Silicon Valley Lab. He has authored several IBM Redbooks publications about DB2 for z/OS and related tools, and has conducted workshops and seminars worldwide. During his many years with IBM, in development and in the field, Paolo has worked mostly on database systems.

Jason Arnold has been with IBM as an IBM System z® technical specialist for Information Management software since 2006. Before joining IBM, Jason received a bachelor’s in Computer Science from Northern Illinois University, where he studied mainframe programming. Jason also holds a bachelor’s degree in Mathematics from Northern Illinois University, and a Master’s degree in Computer Science from DePaul University. His focus is on modernization of existing environments.

Leticia Cruz is a Senior IT Specialist and DB2 DBA in the System z Client Center in Poughkeepsie, New York. She works with clients, IBM Business Partners, and independent software vendors (ISVs) on benchmarks, proofs of concepts (POCs), proofs of technologies, client demonstrations, and workshops. Leticia has extensive experience in building very large databases on DB2 for z/OS in a data warehousing environment. She is currently responsible for the development and promotion of DB2 Analytics Accelerator for z/OS assets at the IBM Systems and Technology Group worldwide (STG WW) Client Centers, Mainframe Benchmark Center. She worked as a Systems Analyst and application programmer before she joined IBM 18 years ago. Leticia co-authored *e-Business Intelligence: Data Mart Solutions with DB2 for Linux on zSeries*, SG24-6294 and *50 TB Data Warehouse Benchmark on IBM System z*, SG24-7674. She holds a Bachelor's degree in Management and Industrial Engineering from Mapua Institute of Technology, Philippines.

Jeff Feinsmith is a Principal Analytics Integration Architect in the US. He has 19 years of experience in the data warehousing and analytics field. He has worked at IBM and Netezza for 11 years, with prior roles at a major business intelligence (BI) vendor. In addition, he was a management consultant for a large international consulting firm. Beginning in early 2011, Jeff served as the lead Netezza architect for the DB2 Analytics Accelerator for z/OS. He currently advises worldwide IBM clients on the application of analytics on System z.

Willie Favero is an IBM Senior Certified IT Software Specialist and the DB2 subject matter expert (SME) with the IBM Silicon Valley Lab Data Warehouse on System z SWAT Team. He has 41 years of experience working with databases, including 31 years working with DB2. Willie is a sought-after speaker for international conferences and user groups. He also publishes articles and white papers, contributes to numerous IBM Redbooks publications, and has a top technical blog on the Internet.

Anna Griner is a Technical Specialist Co-op in IBM Software Group, Ecosystems Development, based in Dallas. Anna has recently completed her master's degree in IT Management with a focus on BI. This residency has been part of her System z business intelligence (zBI) mentoring.

James Guo is a Senior Software Engineer in the Silicon Valley Lab, San Jose, California. James has experience with the DB2 for z/OS system, SQL, and Data Warehousing on System z performance analysis and tuning. He has evaluated DB2 performance using several IBM Tivoli® Storage Productivity Center for Replication and client benchmarks. James helps DB2 customers upgrade their applications to new DB2 releases, and is a frequent speaker at International DB2 User Group (IDUG) and DB2 conferences. James co-authored *Hybrid Analytics Solution using IBM DB2 Analytics Accelerator for z/OS V3.1*, SG24-8151.

Chris Harlander is a Senior Project Manager in the Boeblingen Lab, Germany. Chris has more than 20 years of experience in the Information Management field.He started his career as a specialist for Data Warehousing on System z in an IBM internal project, supporting the sales organization with business data. After working as a consultant for data modeling and database implementation, he led several projects in the insurance and automotive industries with DB2 on z/OS as a data server. In 2006, he joined the Boeblingen Lab and led software development projects for DB2 monitoring tools. Currently, he is working as a specialist for DB2 Analytics Accelerator for z/OS in the Center of Excellence for Data Warehousing on System z. Chris holds a Master's degree in Mathematics and Business Administration.

Johannes Kern is a Software Engineer at the Boeblingen Lab, Germany, where for the last three years he has been working on DB2 Analytics Accelerator for z/OS, mainly doing performance engineering and network consulting. He has previously worked on the IBM Extreme Blue® Internship program, IBM Smart Analytics Optimizer for DB2 for z/OS, and the BLINK research project.

Ravi Kumar is an IBM Certified Thought Leader in the IT Specialist profession. He has more than 22 years of experience, and is currently working as a Senior Managing Consultant at IBM Software Group (SWG). He provides technical leadership and consulting services to clients worldwide in DB2 Analytics Accelerator for z/OS. He plays a direct role in increasing the long-term strength, and enhancing the market position and competitive posture, of IBM database products and tools. He holds a Master's degree in Business Administration. Ravi is a co-author of six other IBM Redbooks publications: *Optimizing Restore and Recovery Solutions with DB2 Recovery Expert for z/OS V2.1*, SG24-7606; *DB2 9 for z/OS: Resource Serialization and Concurrency Control*, SG24-4725; *DB2 10 for z/OS Performance Topics*, SG24-7942; *Optimizing DB2 Queries with IBM DB2 Analytics Accelerator for z/OS*, SG24-8005; *DB2 11 for z/OS Technical Overview*, SG24-8180; *Hybrid Analytics Solution using IBM DB2 Analytics Accelerator for z/OS V3.1*, SG24-8151.

Ruiping Li is a Senior Software Engineer in DB2 for z/OS Development at IBM Silicon Valley Lab. She is currently the lead designer for DB2 query acceleration support for the DB2 Analytics Accelerator for z/OS. Over the past decade, Ruiping has led the development for the new DB2 9 feature, *optimistic locking support*, and the new DB2 10 feature, *timestamp*, with time zone data type support. She also designed and implemented several key enhancements in DB2 for z/OS, such as IBM pureXML® index exploitation, index on expression, materialized query tables (MQTs), and multiple coded character set identifier (CCSID) enhancements. Ruiping is also a co-author of the books *The Business Value of DB2 for z/OS: IBM DB2 Analytics Accelerator and Optimizer*, ISBN: 978-1-58347-381-8; and *Hybrid Analytics Solution using IBM DB2 Analytics Accelerator for z/OS V3.1*, SG24-8151.

Andy Perkins is a Data Warehouse and Business Intelligence SME with the IBM Silicon Valley Lab Data Warehouse on System z SWAT Team. He has more than 30 years of experience in the Information Management field. He has worked across platforms, starting his IBM career as an IBM IMS™ database administrator (DBA) on the NASA Space Shuttle project. He spent many years as a technical pre-sales specialist focusing on data warehousing and business intelligence, primarily on distributed platforms, and has trained IBMers worldwide on these solutions. He finally came back home to System z in 2008 with his current position. He has co-authored numerous IBM Redbooks publications on data warehousing and business intelligence topics.

Jonathan Sloan is a Senior Certified Executive Software IT Specialist, with more than two decades of business intelligence experience with multi-dimensional and relational database systems. He has a Bachelor of Science degree in business management and finance from the City University of New York, and a Master's degree in Business Administration from Rutgers University. He joined IBM in 2000, and specializes in helping customers understand the application of IBM Information Management technology to business problems. Jonathan has experience in the healthcare, financial services, and consumer packaged goods industries. He is currently the business intelligence SME and team lead for data warehousing on System z.

Steve Speller is currently a Product Specialist in the Database Tools' business area at Rocket Software, Inc., based in the South of England. Initially a DB2 DBA, he has more than 35 years of experience in the mainframe environment, working with DB2 at IBM and IBM Business Partners. Steve assists IBM Technical Sales with potential and existing customers on DB2 tools, in addition to acting in a client advocate role for the product development teams. Steve has authored many in-depth technical documents, including a variety of hands-on labs focused on leveraging the many features of IBM-branded DB2 tools.

Dino Tonelli is a Senior Software Performance Analyst working in IBM Poughkeepsie. He has over 15 years of experience with System z performance, working on IBM Workload Manager for z/OS (WLM), DB2, data warehousing, and IBM Parallel Sysplex®. Dino currently leads an IBM Systems and Technology Group (STG)/SWG System z Business Analytics Performance Integration (zBAPI) team, which conducts performance evaluations of the IBM zEnterprise Analytics Systems 9700 and 9710 offerings, including the DB2 Analytics Accelerator for z/OS. His main interest is determining best practices and driving optimizations for resource management of System z data warehousing and business analytics. He holds a Master of Science degree in Computer Science from Polytechnic University.

Special thanks to Ute Baumbach, Patric Becker, and Gary Crupi for reviewing the draft of this book.

Thanks to the following people for their contributions to this project:

Anthony Ciabattoni
Ed McCrickard
Rick Huxoll
Tim Willging
Jennifer Nelson
Rocket Software

Ute Baumbach
Patric Becker
Bjoern Broll
Wolfgang Hengstler
Norbert Jenninger
Steffen Knoll
Frank Neumann
Knut Stolze
IBM Boeblingen Lab

Ambica Bhasin
Chih-jieh Chang
Maggie Jin
Maggie Lin
Roy Smith
Bharat Verma
Lingyun Wang
Pierre Washington
Maryela Weihrauch
Binghua Zhen
IBM Silicon Valley Lab

Gary Crupi
IBM Milwaukee

Ernest Balloni
Mike Buechele
Carl Parris
Bruce Smilowitz
Joseph Stein
IBM Poughkeepsie

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Obtain more information about the residency program, browse the residency index, and apply online:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

  ibm.com/redbooks

- Send your comments in an email to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

1

The analytics lifecycle

This chapter introduces the concept and the importance of business analytics, and why IBM System z can be the best platform for its implementation. This chapter also describes the business scenarios for which IBM DB2 Analytics Accelerator for IBM z/OS (DB2 Analytics Accelerator) can provide a possible solution. It concludes with a brief overview of the IBM zEnterprise Analytics System 9700 and IBM zEnterprise Analytics System 9710 offerings.

This chapter contains information about the following topics:

- Why use analytics
- What analytics is
- What this has to do with data and with System z
- What business-critical analytics is
- Why use System z for analytics
- Business usage scenarios
- zEnterprise Analytics Systems 9700 and 9710

1.1 Why use analytics

Analytics has fast become a differentiating factor in our hyper-competitive society. Unless you have completely ignored changes in business and popular culture, you will recognize that analytical processes are playing a greater role in our society.

Analytical processes have entered every domain in our society, including schools, sports, government, insurance, and credit, just to name a few. There are even “Analytics Editions” of magazines. Scarcely a day goes by without articles pertaining to analytics, or articles that use analytics to better explain a topic or issue.

Historically, analytics processes were used to explain *what happened*. Now, analytical processes are more frequently being used to anticipate what *will happen*, and even to identify *what is the best thing to happen*. As you transition from “what happened” to “what will happen,” you move from the historical realm to the here-and-now, which implies that you need to change the way that you think about the underlying infrastructure required to support analytical processes.

This is where System z and the DB2 Analytics Accelerator can help your company compete in the faster-paced analytic world that we live in. When your transaction processing largely occurs on the System z platform, System z becomes the best place to deliver analytic content in near real-time.

1.2 What analytics is

Why would an IBM Redbooks publication be devoted to the topic of analytics? You might have been involved with information technology and reading IBM Redbooks for years, or maybe even decades. If you are a data analyst or a database engineer, you already understand analytics, don’t you?

“What is analytics?” might be a rhetorical question for you. For many technical people, however, this is a question that they are often somewhat reluctant to ask. We often see this at events. When we ask, “Is there anyone who is not comfortable with the concept of analytics?” there is always someone who is willing to say (to everyone else’s relief), “Share your definition of analytics with us so we understand your perspective”.

Therefore, if you are already well-versed in this growing area, feel free to skip to the next section. Perhaps, however, you can read a little and learn about one of the driving factors for improving business results today. Analytics is an area that is growing tremendously, and creating many job opportunities. These opportunities can translate into income growth for your company, but also for those people who have the skills to deliver these capabilities. If this is too simple for you, you can move on to the next section if you want.

Figure 1-1 on page 3 provides an overview of analytics.

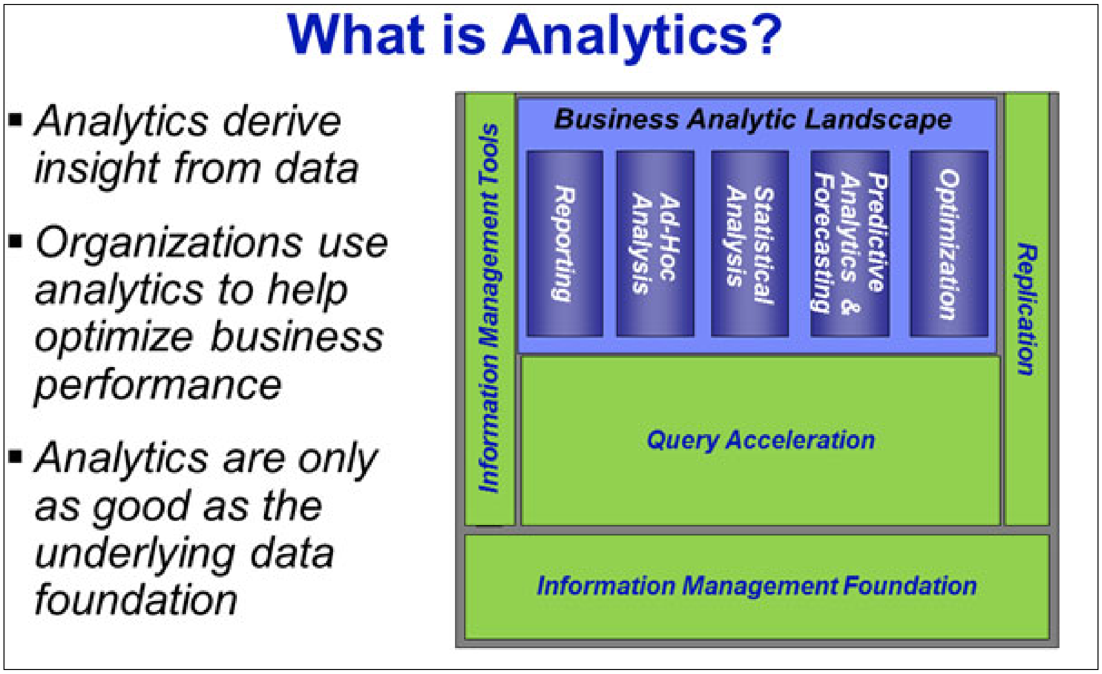


Figure 1-1 What is analytics?

Because you are reading this, you have stayed with us, so this section provides a few definitions. The industry typically defines *analytics* as *the application of mathematical models to data*. We start with the simplest definition of analytics as *data within context*. We begin with this definition because it is a comfortable starting point for most information technology (IT) people, and for many business people, too.

Data within context can describe a report that compares this month's revenue results (sales) to revenue results (sales) the same time last year. Reporting is the simplest level of analytics, and potentially one of the most important.

Several years ago, during the financial crisis, we were working with a bank on a data warehousing solution. We asked them to describe their balance of strategic reporting versus operational reporting. The client quipped, "Today, operational reporting *is* strategic." At a time when companies are failing, operational reporting is absolutely strategic. So, although reporting is not necessarily sexy, it is a foundational and absolutely essential component to analytics.

There are two basic categories of reporting:

- Static reporting
- Exception reporting

The first major category, static reporting, reflects the state of the business at a point in time. We just provided an example with the comparison of year-to-year monthly sales change. The bulk of financial reporting falls in this category.

The second major category of reporting is exception-based reporting. Rather than focusing on the state of the business, exception reporting identifies values that are out of a preplanned range, which enables managers or executives to focus their time on the unusual circumstance, which is called *management by exception*. Exception reporting can also be implemented as an events-driven process that is supported through IBM Cognos® Business Intelligence.

From reporting, we move on to *ad hoc analysis*. Reporting displays a comparison of information, as in the previous example (year-to-year monthly sales change), or identifies an exception condition. It is only a starting point for analysis. This leads to our second classification of analytics, ad hoc analysis. Ad hoc refers to *for this* or *as needed* for a particular purpose. By definition, this type of analysis cannot be pre-planned, nor can it be anticipated, because by its very nature it is for a specific purpose at the moment.

For data people like us, this is a challenge. This is where we traditionally spend a large amount of time thinking through the approach that we need to use to ensure that ad hoc type analysis can be readily accommodated. Any known workload can be tuned through Structured Query Language (SQL) tuning, adding indexes and building aggregate data stores (materialized query tables).

Ad hoc analysis, however, is by its very nature unknown, so we can't tune well for it in advance. This is where we typically spent (notice the past tense) a large amount of time. Later in this book, you will see how DB2 Analytics Accelerator helps to solve this challenge.

Following ad hoc analysis is another type of analysis or analytic process called *statistical analysis*. Statistical analysis breaks down into two separate but dependent categories:

- Descriptive statistics
- Inferential statistics

Descriptive statistics is the area that most of us are familiar with. Most of us know the terms *mean*, *mode*, *median*, *variance*, and *standard deviation*. An entire area of statistics called inferential statistics exists that leverages the descriptive values to infer or anticipate data values. Inferential statistics helps us to determine if data sets are the same, or if they are truly separate data sets.

Being able to infer values brings us to the world of *prediction*. Predictive analytics is an exciting development in the area of analytics, and means exactly what it sounds like: *using analytics to anticipate or predict behavior*. Here is where we get closest to the industry definition of analytics as the applying of mathematical models to data.

After we have described historical data and applied various statistical techniques to the data (for example, regression analysis), we can then determine what a likely result or outcome of incoming data will be. For instance, based on several factors associated with a client, you might be able to predict whether that client will drop your service (attrite it). Alternatively, you might be able to predict, with a certain level of certainty, that the client might buy a specific new product. Predictive analytics is all about anticipating an event.

This might sound familiar to System z people, because we receive "predictive hardware failure" messages. This is where the platform recognizes that there were hardware errors and a failure is likely to occur. Imagine being able to do this for human behavior.

Lastly, if predictive analytics is about anticipating an event or what will likely happen, *optimization* is all about identifying the best outcome. With optimization, the analyst determines the best thing to happen to optimize for a specific variable, whether that variable is profit, revenue, cost, or acquiring the maximum number of customers. Optimization is all about identifying the best outcome.

1.3 What this has to do with data and with System z

You will notice that we have only mentioned data in passing. The reality is, however, that analytics, the process of presenting data in context, is completely dependent upon the underlying data source. Decisions made based on poor data result in bad outcomes. We all know the term, garbage in garbage out. With analytics, predictive or not, you are betting your business on the data. Separating analytics from the underlying data is nearly impossible.

When the world was focused on what had already happened, the expectation was that data could be updated less frequently, and that decisions could be made from data in the past. This is like "driving a car using the rear view mirror". You know where you're going because you know where you've been.

Today, the business world is focused on anticipating what the next best step *should be*. Businesses are no longer happy with determining why customers have left (or a product offer was rejected), but are looking to anticipate that a client will leave and to prevent them from doing so (or to offer the correct product to the correct client at the correct time).

Being able to anticipate clients' behaviors is only half of the problem; being able to act on that knowledge in a timely fashion is the other important component. This happens at the point of client contact at the system of record, just before, during, or just after a transaction, and most often on System z.

1.4 What business-critical analytics is

Operational applications and analytical applications have changed since the original data warehouses were implemented many years ago. Today's operational applications incorporate informational content (analytical content) often stored in a data warehouse or data mart. Because these informational data stores (data warehouses and data marts) provide content to operational applications, they have taken on qualities of service that they were never designed for.

Business-critical analytics are applications that are tightly integrated with System z book-of-records applications and data, and are critical to the running of the business. Failure of these applications for any length of time can result in lost business, client turnover, and reputational and other risks. Users need access to this data in real time or near real time.

Typically, these applications support a large concurrent user population with high-volume requests. These applications require online transaction processing (OLTP)-like qualities of service, with a high degree of reliability, continuous availability, scalability, and security, and low data latency. System z provides superior support for these business-critical analytics applications.

1.5 Why use System z for analytics

Organizations are using analytics to improve their performance and compete more effectively. Today's operational applications are incorporating analytic content and functions to better service customers through mass customization (better anticipating client requirements), and driving greater revenue and profitability for the organization.

As more applications require access to these business-critical analytic functions, the underlying characteristics of analytic applications have changed. We move from few users to many users. The transactions become client-focused, and they access more granular data rather than aggregate data. Lastly, qualities of service become much more important.

These business-critical applications require proven infrastructure that is scalable, available and reliable. *Scalability* refers to both data scalability and user scalability. *Availability* is the certainty that the platform will be up and running, and able to receive requests and transmit results. Lastly, *reliability* refers to transaction consistency, so that a client service query consistently performs the same way every day.

Data governance and security must also be effective, because as data is exposed to customer service and customers, and they make decisions dependent upon this data, the risks go up. Security is essential in today's environment with the many data breaches. Lastly, analytics must be both timely and effective, so the underlying data must be current.

The traditional approach to analytics has been to move the data from the mainframe environment to a massively parallel database infrastructure, as shown in Figure 1-2. The data model supporting transactional applications, typically in an application-dependent, normalized format, is often different from the type of model that supports query, typically de-normalized dimensional models.

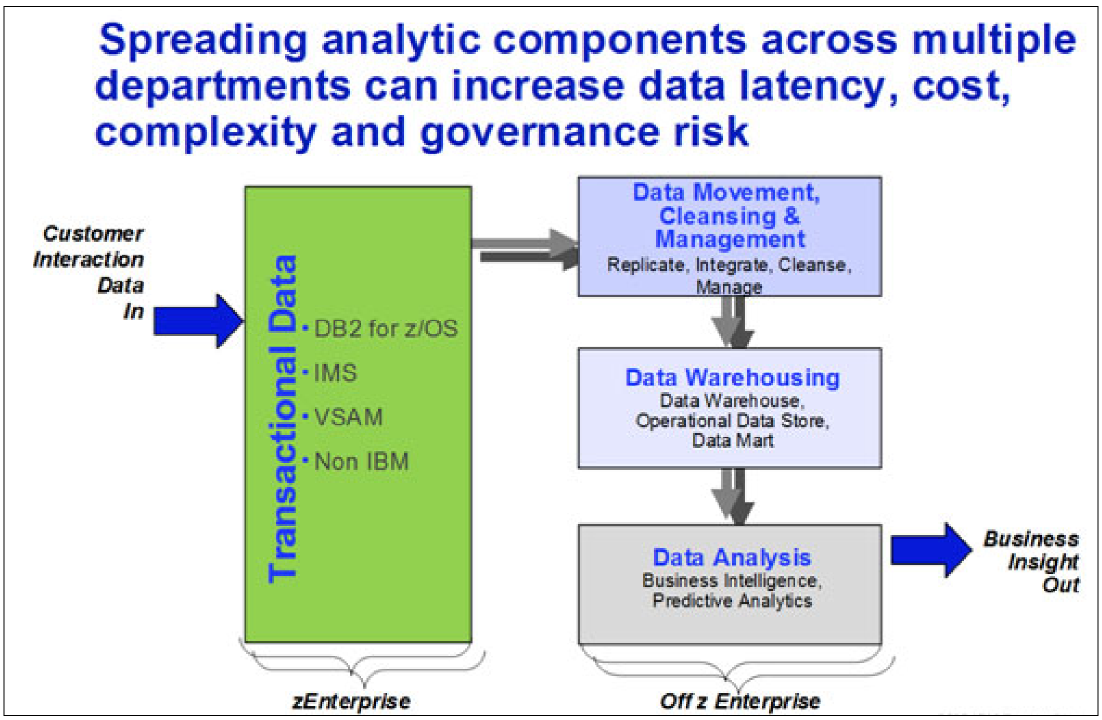


Figure 1-2 Effect on data latency

To move from one data model to another, the data must be transformed, usually through the use of extract, transform, and load (ETL) facilities. Often, these high-performance transformation engines are on a separate platform from both the transactional system and the target data repository (operation data store (ODS), data warehouse or data mart). This then means that the data must be migrated to another, separate platform for ETL processing.

Furthermore, the business intelligence, online analytical processing (OLAP), or predictive analytics platform might be on yet again a separate platform, managed by a different department. Spreading these analytic components across multiple departments and multiple platforms can increase complexity, cost, latency, data governance, and security.

All of these factors can increase the risks associated with decision making.

1.6 Business usage scenarios

Analytics have a direct effect on business performance, and are critical for improved decision making, responding to business situations faster, and improving competitive advantage. The requirement to derive analytics from the most current, accurate data, as it is entering an organization, is a top priority.

With DB2 Analytics Accelerator, clients whose data comes into their business from a zEnterprise mainframe have the distinct advantage of a hybrid solution that enables them to run their analytics on the same platform as their transactional data, as shown in Figure 1-3.

DB2 Analytics Accelerator Usage Scenarios

Understand your workload and data:
On average, 70% of the data that feeds data warehousing and business analytics solutions originates on the System z platform (financial information, customer lists, personal records, manufacturing…)

Understand how the data is being analyzed today:	DB2 Analytics Accelerator Use Case	Benefits
With long running queries (>1 min) in DWH or Operational Data Stores on DB2 for z/OS	Rapid Acceleration of Business Critical Queries	Performance improvements and cost reduction while retaining System z security and reliability
Offloaded to external distributed DWH or Datamarts	Reduce IT Sprawl for analytics	Simplify and consolidate complex infrastructures, low latency, reliability, security and TCO
Not being analyzed yet: • queries are set aside due to performance or cost challenges • new BI applications being considered - new reporting and compliance, operational BI…	Derive business insight from z/OS transaction systems	One integrated, hybrid platform, optimized to run mixed workload. Simplicity and time to value
Analysis, based on a lot of historical data within DB2, without any updates to the data	Improve access to historical data and lower storage costs	Performance improvements and cost reduction

Figure 1-3 The DB2 Analytics Accelerator usage scenarios

This hybrid, single-platform, integrated approach brings analytics closer to the transactional source data, providing the following advantages:

- Minimizing the need to move data across platforms
- Reducing data latency, cost, and complexity
- Improving data governance and reducing risk

Therefore, business users across the enterprise have access to more timely, accurate, and near real-time insight for improved decision making.

Looking at DB2 Analytics Accelerator usage scenarios is looking at where the transactional source data is being analyzed today:

- Rapidly accelerate business-critical queries

  If the data is analyzed on the mainframe, DB2 Analytics Accelerator can be attached to that mainframe. Therefore, the overall client solution will benefit, gaining performance improvements and cost reduction, while retaining System z security and reliability. DB2 Analytics Accelerator is used for rapid acceleration of business-critical queries.

- Derive business insight from z/OS transaction systems

  If the data is not being analyzed yet, because queries are set aside due to performance or cost challenges, or because a new business intelligence (BI) application is being considered, DB2 Analytics Accelerator can be attached to the mainframe to complement DB2 for z/OS, to form a hybrid infrastructure that best hosts this new workload. DB2 Analytics Accelerator is used to derive business insight from z/OS transaction systems.

- Reduce IT sprawl for analytics initiatives

  If the data is off-loaded to a distributed data warehouse or data mart, DB2 Analytics Accelerator could help to simplify and consolidate complex infrastructures, and reduce time-consuming ETL processes. This results in lowering data latency, enhancing reliability, increasing security, and decreasing overall total cost of ownership (TCO). DB2 Analytics Accelerator is used to reduce IT sprawl for analytics.

- Improve access to historical data and lower storage costs

  If the analysis is based on much historical data in DB2 for z/OS, without any updates to that data, DB2 Analytics Accelerator can be used to improve performance and reduce cost for this analytical workload. In this case, DB2 Analytics Accelerator is used to improve access to historical data and lower storage costs.

1.6.1 Rapidly accelerate business-critical queries

Many organizations have created their data warehouses on zEnterprise to take advantage of the platform's quality of service (QoS), and because the majority of their operational data is captured and housed on the System z platform. However, data-intensive queries that characterize data warehouse, BI, and analytics workload are typically complex and long-running.

Historically, running these queries in parallel with OLTP workload presented a challenge. Due to their processor-intensive profile, these queries might require extensive tuning efforts. In some cases, the queries cannot be run and become "forgotten queries" or might need to be scheduled in batch processes overnight so that they do not affect corporate users during the day. But, such overnight schedules do not deliver information in the timely manner required in today's business critical environment.

Extending DB2 for z/OS with DB2 Analytics Accelerator provides dramatic improvements to query response, up to 2000 times faster, to support time-sensitive decisions. Organizations can gain great advantages from this speed:

- Make better business decisions from the most up-to-date information
- Increase user productivity by delivering reports and information much faster
- Meet the growing business demand for analytical reports based on trusted data

This unique approach also removes the need to analyze, tune, and test analytic queries. It can reduce the time and effort required to manage DB2 for z/OS database applications in support of complex queries.

Figure 1-4 shows a client example of a global reinsurance and insurance provider that had claims information data that was spread across multiple international locations, and came from numerous insurance companies. They needed a way to accomplish the following goals:

- Obtain insight from all of that data
- Deliver insight to their business users quickly
- Help those business users better understand risk
- Identify profitable segments

With DB2 Analytics Accelerator, they can now provide fast, accurate analysis from claims transactions in far less time than before. Their users are getting their reports as much as 70% faster (reports that took 10 hours are now available the same day), so user satisfaction has increased dramatically. Because those reports contain key analytics that guide pricing and decision making across their lines of business, the solution has sharpened the company’s competitive edge.

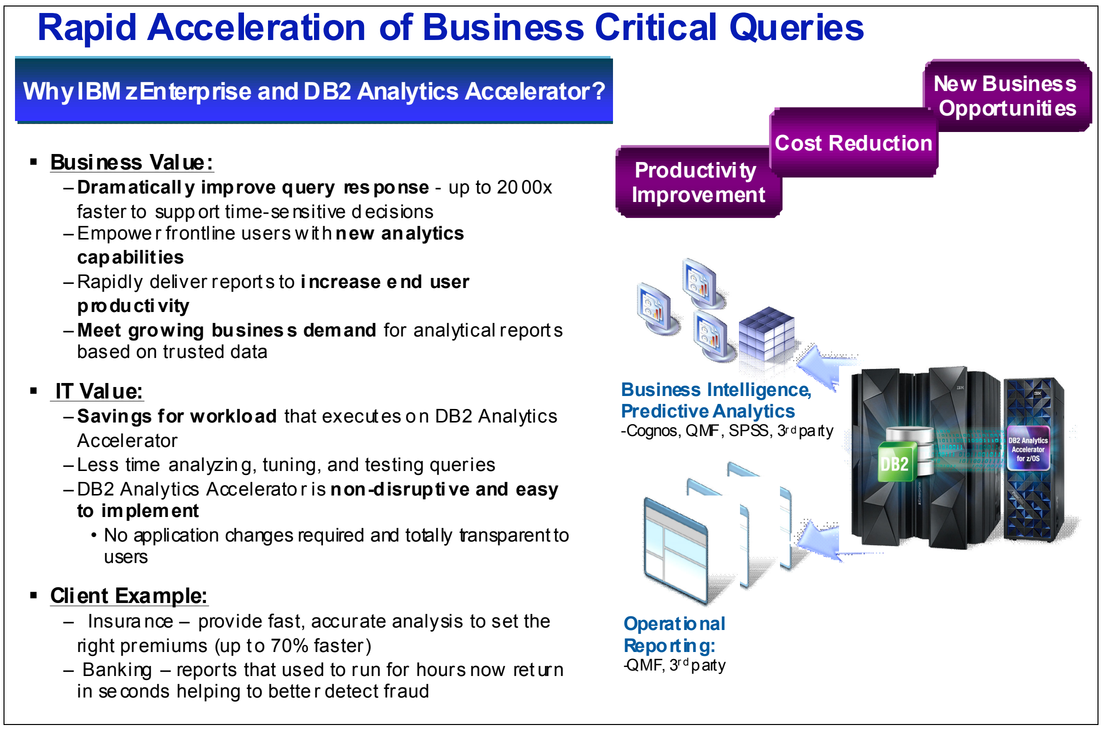


Figure 1-4 Acceleration of business-critical queries

1.6.2 Derive business insight from z/OS transaction systems

More and broader sets of users are now demanding access to analytics, from business users who need timely, accurate insight to do their jobs better, to customers who expect access to information any time, 24x7. These users have high expectations about the quality and time frame in which analytic insight is made available to them. Because analytics now have a direct effect on business performance, near real-time analysis is critical for improved decision making, responding to business situations faster, and improving competitive advantage.

Therefore, the requirement to deliver near real-time analytics is crucial to business success. Clients whose data comes into their business from an zEnterprise mainframe have the distinct advantage of being able to run their analytics on the same platform as their transactional data. This hybrid, single-platform, integrated approach means business users get access to timely, accurate, and near real-time insight for improved business performance.

The introduction of DB2 Analytics Accelerator in conjunction with zEnterprise brought a revolutionary change, resulting in a hybrid computing platform on zEnterprise, as shown in Figure 1-5. This hybrid infrastructure blends the best attributes of DB2 for z/OS, which is built for transactional workloads, with a cost-effective, high-speed query engine to run complex business analytics workloads.

DB2 Analytics Accelerator and DB2 for z/OS form a self-managing, hybrid database management system (DBMS) that transparently runs each workload in the most efficient way, so each transaction (OLTP, batch, or analytic query) is run in its optimal environment for the greatest performance and cost efficiency. DB2 Analytics Accelerator turns DB2 for z/OS into a universal DBMS, capable of handling both transactional and analytical data.

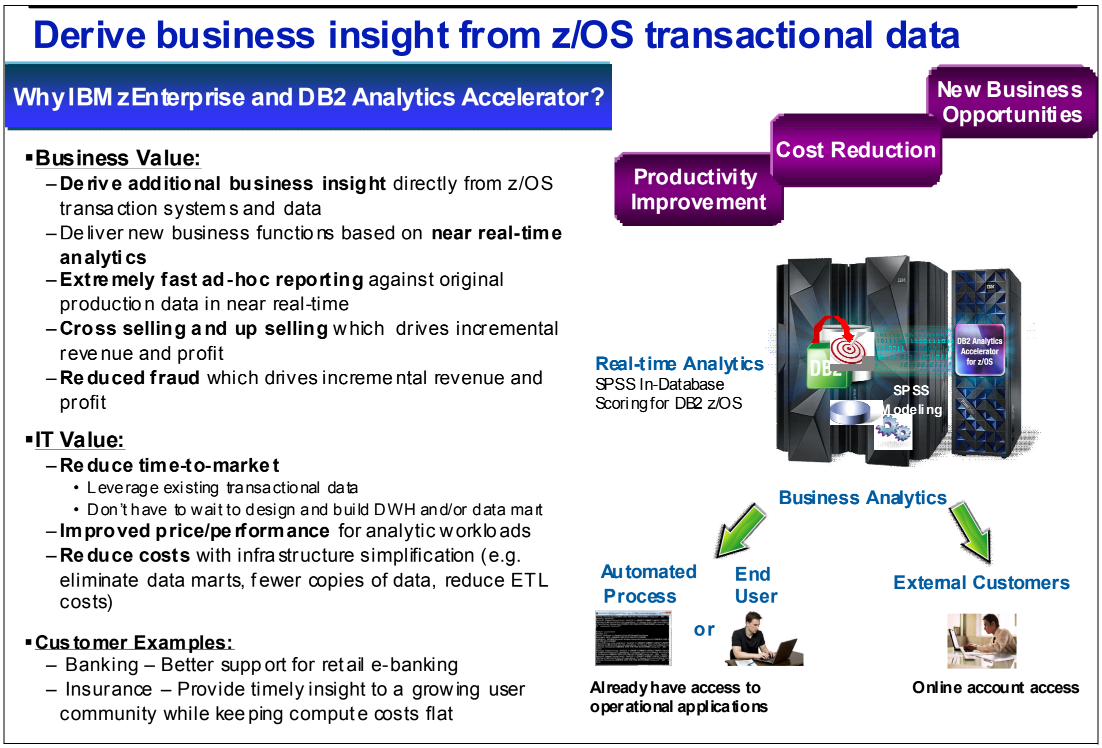


Figure 1-5 Business insight from z/OS transaction systems

Organizations use DB2 Analytics Accelerator to help them gain additional business insight directly from their z/OS transaction systems and data. This enables them to deliver new business functions based on real-time and near real-time analytics, which supports several applications:

- Cross-sell and up-sell to drive sales
- Fraud detection to reduce risk and drive profits

1.6.3 Reduce IT sprawl for analytics initiatives

Today, analytics have a direct effect on business performance, and are critical for improved decision making, faster response to business situations, and improved competitive advantage. Because of the need to support more data and growing numbers of analytics users, infrastructure sprawl can become an inhibitor as IT organizations strive to support new requirements for business-critical analytics that demand the highest QoS.

With traditional analytics approaches, organizations might duplicate and move data from System z to distributed departmental systems for analytic processing. Transferring data from one platform to the next, however, can introduce data quality and security issues, and add data latency. Moreover, departmental analytics environments often struggle to deliver the availability, scalability, and performance levels required.

Figure 1-6 shows how organizations with the majority of their transactional data originating in, and housed on, zEnterprise are rethinking their analytics approach. These organizations are bringing together their transactional processing, data warehousing, and analytic tooling on System z. Implementing DB2 for z/OS and DB2 Analytics Accelerator in a zEnterprise environment can help to meet and exceed business-critical analytics objectives and avoid the drawbacks of traditional analytics approaches.

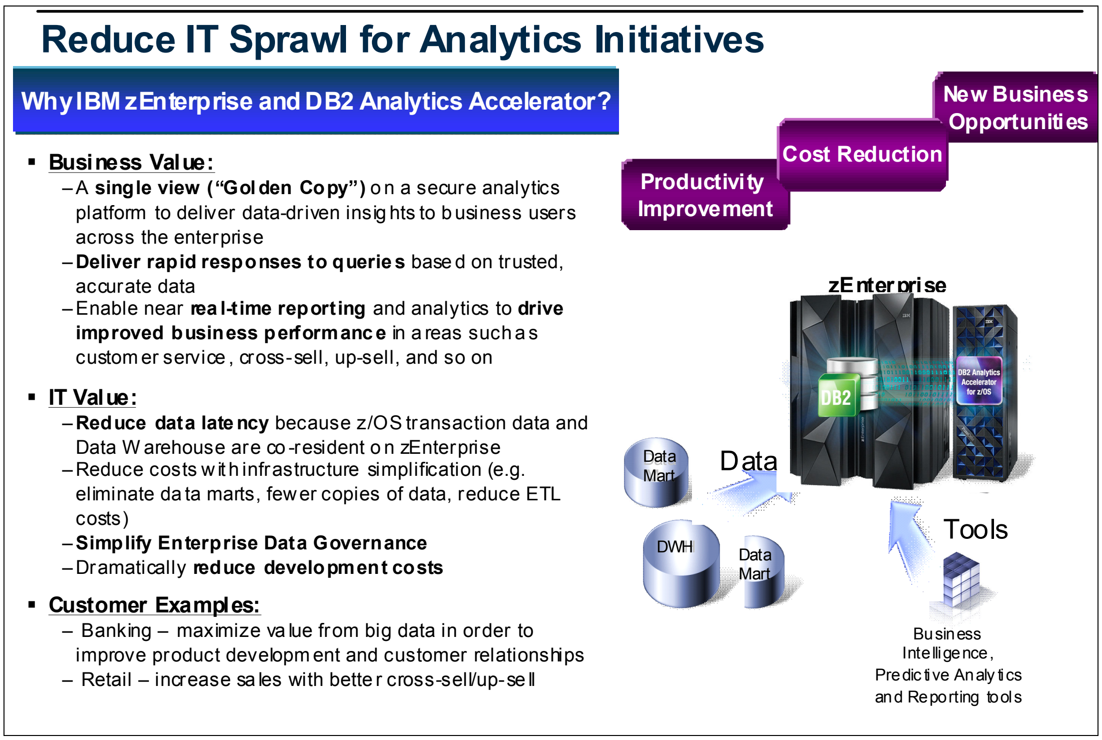


Figure 1-6 Consolidation of analytics initiatives

DB2 for z/OS, DB2 Analytics Accelerator, and zEnterprise offer a hybrid solution that brings together high-volume business transactions, batch reporting, and complex analytic queries running concurrently in a mixed-workload environment. This new approach helps organizations to align their business operations and technology using a single, common platform to drive out complexity and cost while increasing flexibility to support business-critical analytics.

Organizations can use their existing zEnterprise infrastructure, people, and processes to deliver their business-critical analytics across the enterprise, with superior System z QoS: Reliability, availability, and scalability. This hybrid solution provides a single view on a secure analytics platform to deliver data-driven insights to business users across the enterprise, so decisions are made based on easy-to-find, trusted, and accurate data, all driving improved business performance.

1.6.4 Improve access to historical data and lower storage costs

As the use of analytics becomes more pervasive, organizations strive to make more data accessible to users to drive better decision making which can improve customer satisfaction, managerial oversight and support regulatory requirements. Organizations are challenged to balance the need for keeping large volumes of data available for analytic analysis with managing costs and keeping systems running optimally.

Most analytical systems are based on data that is 95% historical, and therefore static. A retailer, for instance, might maintain seven years of past sales histories that contain every transaction for every product sold to every client. Because this data is historical, it generally is not subject to revision or updates.

The High Performance Storage System (HPSS) feature of DB2 Analytics Accelerator helps reduce the cost of storing, managing, and processing this type of data, as shown in Figure 1-7.

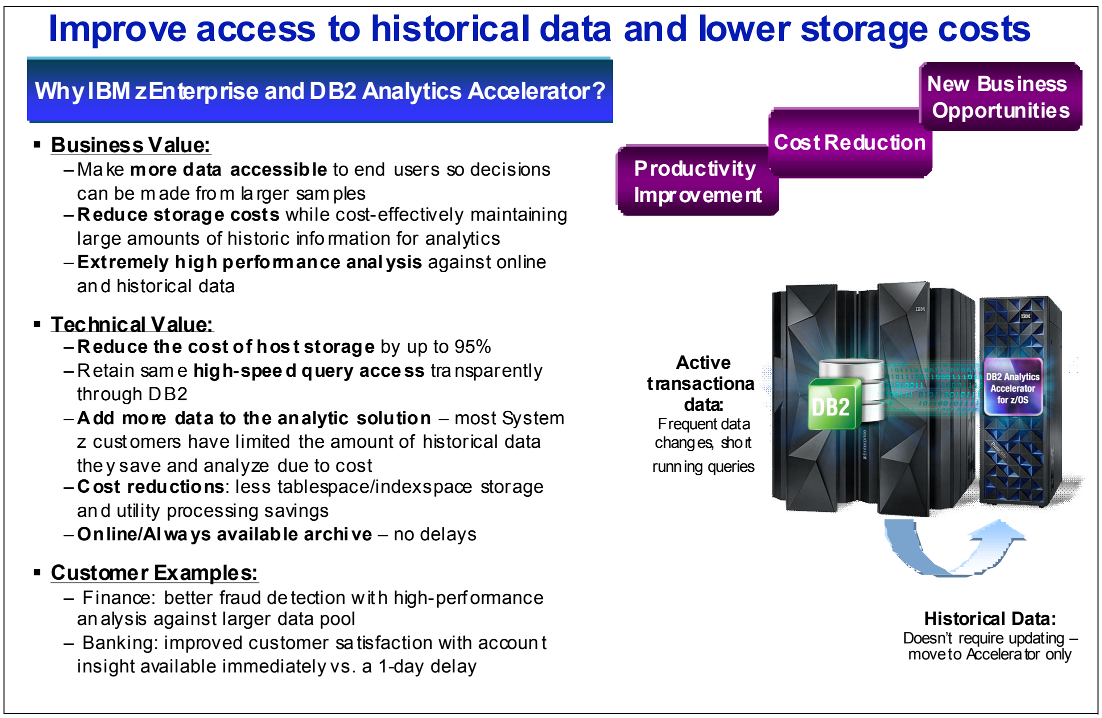


Figure 1-7 Access historical data and lower storage costs

Organizations can move static data in tables or table partitions to the DB2 Analytics Accelerator and remove the data from DB2 for z/OS. All of the data is still managed by, and safeguarded in, the DB2 directory, and all of the queries that target that data are now directed only to the DB2 Analytics Accelerator.

This process can dramatically reduce storage costs on zEnterprise, and enable organizations to substantially increase the amount of history maintained for each subject area, so decisions can be made from much larger data samples. Another great advantage of the HPSS feature is the capability of extremely high-performance analysis against both online and historical data, without noticeable effects to users and applications.

1.7 zEnterprise Analytics Systems 9700 and 9710

The IBM zEnterprise Analytics Systems 9700 and 9710 are a set of fully integrated and scalable data warehouse and analytics solutions designed to give an organization the insight it needs to work more intelligently. They combine software, server, and storage resources to put the correct answers in the hands of the corporate decision makers today, placing your business in the best possible position to answer the questions of tomorrow.

The zEnterprise Analytics Systems 9700 and 9710 are made up of two hardware solutions. The 9700 solution has the following characteristics:

- Is based on the IBM zEnterprise EC12 (zEC12) enterprise-class solution for larger enterprises
- Includes the DB2 Analytics Accelerator for z/OS
- Optionally includes three add-on packs:
  - Data Analytics Pack
  - Data Integration Pack
  - Fast Start Services Pack

Figure 1-8 zEnterprise Analytics System 9700

For clients who require a smaller footprint solution, the zEnterprise Analytics System 9710 provides an entry-level, integrated, business-class offering built on the zEnterprise BC12 (zBC12) footprint. This solution enables clients to cost-competitively, quickly, and securely deliver business reporting and analytics. With the zEnterprise Analytics System 9710, the DB2 Analytics Accelerator and three add-on packs are all optional.

The components that make up the zEnterprise Analytics Systems 9700 and 9710 have been preconfigured:

- Preselected to deliver a comprehensive, end-to-end, flexible solution
- Pretested to meet business-critical analytic demands
- Solution priced to enable customers to deploy a next-generation decision system as an add-on to an existing zEnterprise system, or as a new, stand-alone system

These components consist of mature, tested hardware and software that have set the standards for others to follow. They are all leaders in security, availability, recoverability, and virtualization:

- zEC12 or zBC12

  These include general-purpose processors, the IBM System z Integrated Information Processor (zIIP), and memory, all sized based on workload requirements.

- DB2 Analytics Accelerator for z/OS V4.1

  DB2 Analytics Accelerator is the main subject of this book.

- DB2 for z/OS Value Unit Edition (VUE)

  VUE is a one-time-charge priced component for DB2 for z/OS for eligible net new applications or workloads.

- DB2 Utilities Suite for z/OS

  DB2 Utilities Suite is at the core of managing DB2 for z/OS. The utilities help you to minimize downtime associated with routine DB2 data maintenance, while ensuring the highest degree of data integrity.

- z/OS V1.13 operating system stack, including IBM zSeries/Virtual Machine (z/VM®)

  As of this writing, z/OS V2.1 is the current release of the operating system (OS).

- IBM System Storage® DS8700

  The System Storage DS8870 is designed to manage a broad scope of storage workloads that exist in today's complex data center, and do it effectively and efficiently.

These components have been sized, deeply optimized, and are ready for use in a wide variety of configurations, ranging from modest database sizes and users through petabyte (PB) systems with hundreds to tens of thousands of users.

The three optional add-on packs (Data Analytics Pack, Data Integration Pack, and Fast Start Services Pack) provide the flexibility to meet any client's specific analytic needs:

- The Data Analytics Pack provides a full range of business intelligence, predictive analytics, and performance management capabilities through the use of IBM Cognos Business Intelligence, Cognos TM1®, IBM DB2 Query Management Facility™ (QMF™) Enterprise Edition for z/OS, IBM SPSS® Modeler with Scoring Adapter for zEnterprise, and SPSS Modeler Gold.
- The Data Integration Pack consists of the IBM Information Server for Data Integration and IBM InfoSphere® Data Replication tools.
- The Fast Start Services Pack provides predefined services that can be tailored to meet the unique needs of your environment.

The processor and memory sizing required for the zEnterprise Analytics System 9700 is determined using varying query workload types, and the total data warehouse size based on row data. The sizing for a zEnterprise Analytics System 9700 does assume a combination of general-purpose engines (processors), zIIPs, and the DB2 Analytics Accelerator to process the defined workload successfully. The zEnterprise Analytics System 9710 does not include a DB2 Analytics Accelerator as part of the sizing, although it is optionally available.

When the machine capacity has been estimated, the next decision is to determine whether the hardware and software solution should be stand-alone (new processor) or bolt-on (capacity added to an existing processor).

The stand-alone option sends a new System z Enterprise server, based on the capacity determined during the sizing, that runs nothing but the zEnterprise Analytics System 9700 or 9710 solution. The bolt-on option takes the same capacity (processors, zIIPs, IBM System z Integrated Facility for Linux (IFL), and memory) that would have been sent as part of a corresponding stand-alone system, and adds that capacity to an existing System z server that the client already owns.

A new logical partition (LPAR) is then defined to use the additional capacity added to support the zEnterprise Analytics System 9700 or 9710.

There are a few other considerations when accepting a bolt-on solution. The new capacity is added to the current server with the understanding of the following stipulations:

- The total amount of required capacity is split between processors and zIIPs.
- It is possible that the System Modification Program (SMP) effect caused by adding additional processors or zIIPs to an existing configuration might result in a lower capacity than required for a particular zEnterprise Analytics System 9700 configuration. It might be necessary to add additional processors or zIIPs to compensate for this capacity reduction.
- The total number of zIIPs cannot exceed two times the number of processors on the zEC12 and zBC12. The 2:1 zIIP to processor ratio exists only for the zEC12 and zBC12.

The zEnterprise Analytics System 9700 can help quickly deliver vital business insight to users across an enterprise when and where it is needed. As a result, an organization can anticipate and respond to ever-changing business conditions, and uncover and capture new revenue opportunities. The following list includes examples of the added value provided by the zEnterprise Analytics System 9700:

- Extended operational environment, including analytic processing without affecting transactional systems
- Integrated hybrid technology that optimizes the environment for both high-performance analytics and OLTP
- A single database image to self-manage the diverse range of queries that every business needs to answer
- An analytics platform that can support deep mining and analytical reporting with minimal data movement
- The ability to perform operational analytics with the same quality of service as an OLTP environment on zEnterprise
- A platform of unequaled security (evaluation assurance level (EAL) 5+) with extreme availability and recoverability

The zEnterprise Analytics System 9700 is built on the zEC12 and zBC12 processors. A set of stock configurations developed by the System z Business Analytics Performance Integration (zBAPI) team at the IBM lab in Poughkeepsie, NY, exist only as an initial guidance tool. Detailed sizing, required to order the zEnterprise Analytics System 9700, is based on a client's specific workload profile. When detailed sizing is required, an IBM Lab Advocate or IBM System z data warehousing (z/DW) SWAT Team member will be assigned to assist.

There are two query configurations used to create the workload metrics:

- A basic configuration, defined as *workload*, which is primarily used for query, has a relatively simple database and DB2 Analytics Accelerator refresh processes. The refresh processes should not interfere with a client's real-time processing, sometimes referred to as *fitting within a client's batch window*.

  The basic configuration also assumes only light, if any, INSERT/UPDATE/DELETE activity with minimal extract, load, and transform (ELT), ETL, or other processing against the warehouse data that could affect a warehouse's capacity usage or availability.

- An advanced configuration that is intended for workloads that go beyond query and basic refresh processing. Queries for these workloads tend to be more complex, have a higher likelihood of change activity form INSERT/UPDATE/DELETE activity, ELT, or ETL processing. In most cases, the workload associated with the advanced configuration has a more stringent service level agreement (SLA) requirement.

The testing methodology included the use of workloads consisting of queries of different complexities, running in both the batch and distributed environments, with tables participating in a star schema. These workloads were processed on varying hardware configurations (combinations of processors and zIIPs) accessing tables that ranged in size from 100 gigabytes (GB) to 10 terabytes (TB).

Metrics for larger databases were extrapolated from the tested results. The more rows "touched" by a query, the more capacity is required to meet SLA expectations. The number of processors (both central processor and zIIP), millions of instructions per second (MIPS), and memory requirements all increased to accommodate the larger data capacities.

For each of the testing configurations, the DB2 Analytics Accelerator was defined on an IBM PureData System for Analytics N2002-002 (1/4 rack).

The 9710, built on the zBC12 processor, is designed for customers with a smaller System z footprint. To provide a lower price point, the 9710 is a slightly different offering from the 9700 described in the previous section. In addition to using a zBC12, the default 9710 comes as two alternative configurations. The first configuration offers traditional System z general-purpose processors (CP) along with System z Integrated Information Processors (zIIP). The second configuration offers the traditional System z processors (CP and zIIP) along with an IBM DB2 Analytics Accelerator. With the 9710, both configurations offer disk (DS8870 storage system) as an option.

The following list includes highlights from the current zEnterprise Analytics System 9700 solution:

- DB2 Analytics Accelerator V4.1
- An update to the zEnterprise Analytics System 9710 offering with zBC12 to meet specific customers size requirements
- Enhanced business analytics capabilities with DB2 11 for z/OS
- New software additions and revisions in the Data Analytics Pack:
  - Cognos BI for zEnterprise V10.2.1
  - Addition of IBM Cognos TM1 10.2
  - Replacement of IBM SPSS Modeler 16 with SPSS Modeler Gold 16
- Addition of IBM SPSS Modeler 16 with Scoring Adapter (Linux for System z

- Addition of QMF Enterprise Edition VUE into our Data Analytics Pack to broaden the spectrum of customers with analytics needs
- New additions in the Data Integration Pack include the addition of InfoSphere Data Replication for DB2 for z/OS; InfoSphere Data Replication for Linux, UNIX, and Windows; and InfoSphere Data Replication for Linux for System z to enhance our end-to-end analytics solution offering

Deeply optimized and ready to use, the zEnterprise Analytics Systems 9700 and 9710 can help quickly turn information into insight, and deliver that insight where and when it's needed. As a result, corporations can quickly respond to ever-changing business conditions, and uncover and capture new revenue opportunities for their organization.

Based on an established infrastructure of IBM software, servers, and storage, the IBM zEnterprise Analytics Systems 9700 and 9710 are designed to be simply and flexibly deployed, with the ability to expand to fit evolving corporate business needs.

2

Database design considerations

This chapter provides information about any special database design considerations when using IBM DB2 Analytics Accelerator for IBM z/OS (DB2 Analytics Accelerator).

The DB2 Analytics Accelerator, and the underlying IBM Netezza (now IBM PureData for Analytics) technology, are independent of specific data models. As a result, it generally performs extremely well for existing data models, assuming that effective data distribution and data clustering (organization keys) strategies are taken. Typically, a random distribution and no organization keys provide good access paths.

This chapter describe the data distribution and the data clustering, as they apply to all DB2 Analytics Accelerator systems. Secondarily, it covers logical and physical data model considerations that apply to new data models, but that can also be used when feasible for existing data models.

This chapter contains the following topics:

- DB2 Analytics Accelerator data distribution strategy
- Zone maps (automatic partitioning)

2.1 DB2 Analytics Accelerator data distribution strategy

Good data distribution is one of the most important indicators of query performance in DB2 Analytics Accelerator. It is important to start with an understanding of the fundamentals of data distribution.

To achieve the performance benefits of the massively parallel processing (MPP) architecture, it is necessary that data and processing are distributed as evenly as possible across the processing resources. In DB2 Analytics Accelerator, the parallelized processing unit is called a *data slice*. For example, an N2002-010 has 240 blades. Each blade uses a dedicated set of disks, and DB2 Analytics Accelerator reads a data slice in parallel from these disks.

MPP provides linear scalability, so if data (and therefore query processing) is evenly distributed across all of the available slices, a query on an N2002-010 runs 240 times faster than if we had only a single slice for the same query. Balanced distribution of the data is the key to this performance.

DB2 Analytics Accelerator uses the distribution key of a table to determine how data is allocated to the available slices. The distribution key is composed of one or more columns of the table. As data is loaded into the table, the distribution key is treated as a hash key to allocate the incoming row to a data slice. Repeating values are always assigned the same data slice. Therefore, the distribution key can also be used by the system to find the data slice containing a particular value or set of values.

When data is unevenly distributed across the slices, it is called *data skew*. Skew affects disk usage, but equally (or more) importantly, it affects query performance. When more data for a query is present on a subset of data slices, this causes some data slices (processor + Field Programmable Gate Array (FPGA) + memory + disk) to work harder and longer than others, therefore eliminating the linear performance scalability advantages of the MPP architecture.

Additionally, because the data is not evenly distributed across the available storage in each slice, some slices fill up faster than others, leaving empty space in some slices and, therefore, “filling” the entire DB2 Analytics Accelerator with less data than it can effectively manage.The worst case scenario would be all data for a table loaded into only a single slice, therefore limiting all processing for queries on that data to that single slice on an IBM PureData System for Analytics N2002-010 while leaving 239 slices idle.

The golden rule of the IBM Netezza appliance (now PureData System for Analytics) is simple: Good distribution = good performance.

The key point to always remember about DB2 Analytics Accelerator performance is that it is an MPP system with a shared-nothing design, and all data tables are spread over all nodes (that is, the data slices):

- *All* tables are distributed across *all* active database blades.
- *All* queries run in parallel against *all* active blades.
- *All* loads run in parallel against *all* active blades.

Good distribution is a fundamental element of performance:

- A data slice is an individual element of parallelism.

  For example, PureData System for Analytics N1001-010 = 92 data slices and PureData System for Analytics N2002-010 = 240 data slices.

- If all data slices have the same amount of work to do, a query is 92 times quicker than if one data slice was asked to do the same work.

- Distribution is determined using the DB2 Analytics Accelerator graphical user interface (GUI).
- The same value is always hashed to the same data slice.
- Bad distribution is called data skew.
- Skew to one data slice is the worst-case scenario.
- Skew affects the current query, and others, because the data slice has more work to do.
- Skew means that the machine fills up much quicker, because the system is as full as the largest data slice.

2.1.1 Distribution and performance

MPP is more efficient when table rows are distributed evenly across blades. Each blade has a comparable number of records, and performs approximately the same amount of processing. The optimal goal is to have all blades engaged in all operations (**SELECT**, **JOIN**, **SUM**, **SORT**, **GROUP**, and so on).

For large tables, a hash distribution that evenly distributes table rows across all blades is wanted and optimal. The system distributes rows with the same hash value to the same blade. The distribution key is hashed to a numerical value. Based on the hash value, it is assigned to a specific bucket. Every blade then receives a specific subset of the buckets.

As seen in the example in Figure 2-1, data and processing are relatively evenly spread across the available slices, therefore each blade (or slice) finishes their respective processing at more or less the same time. This is the ideal situation for best performance, where all blades perform equal shares of the work, and therefore start and finish at the same time.

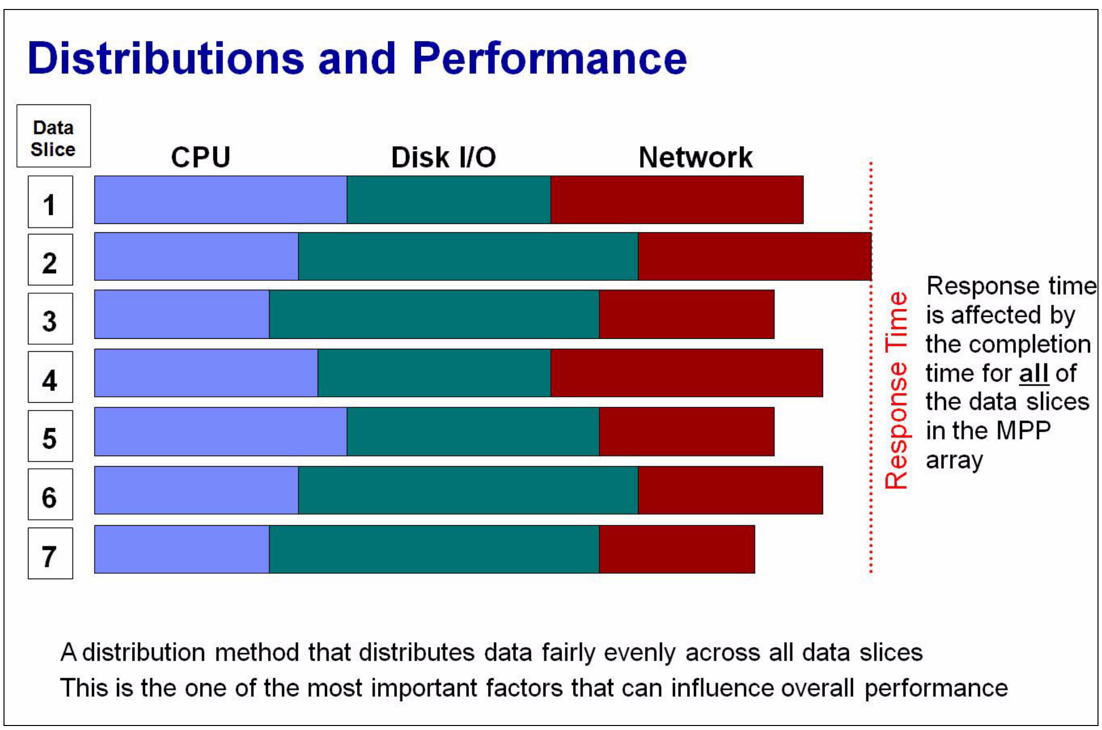


Figure 2-1 Distribution and performance

Data skew

It is important to select a distribution key (column) that provides an even distribution of data across all Snippet Processing Units (SPUs)[1]. Select a distribution key with unique values and high cardinality to distribute the data evenly across all SPUs in the Netezza Performance Server® (NPS®) system.

Figure 2-2 shows a classic example of poor data skew when a binary data type (or even just a column with only 2 or very few values) is chosen as the distribution key. In this example, gender (male or female) is chosen as the distribution key. When the data is loaded, the gender value is hashed, and data rows are written to one slice or the other based on the binary value. Because there are only two values, only two slices are used, leaving all of the other slices empty.

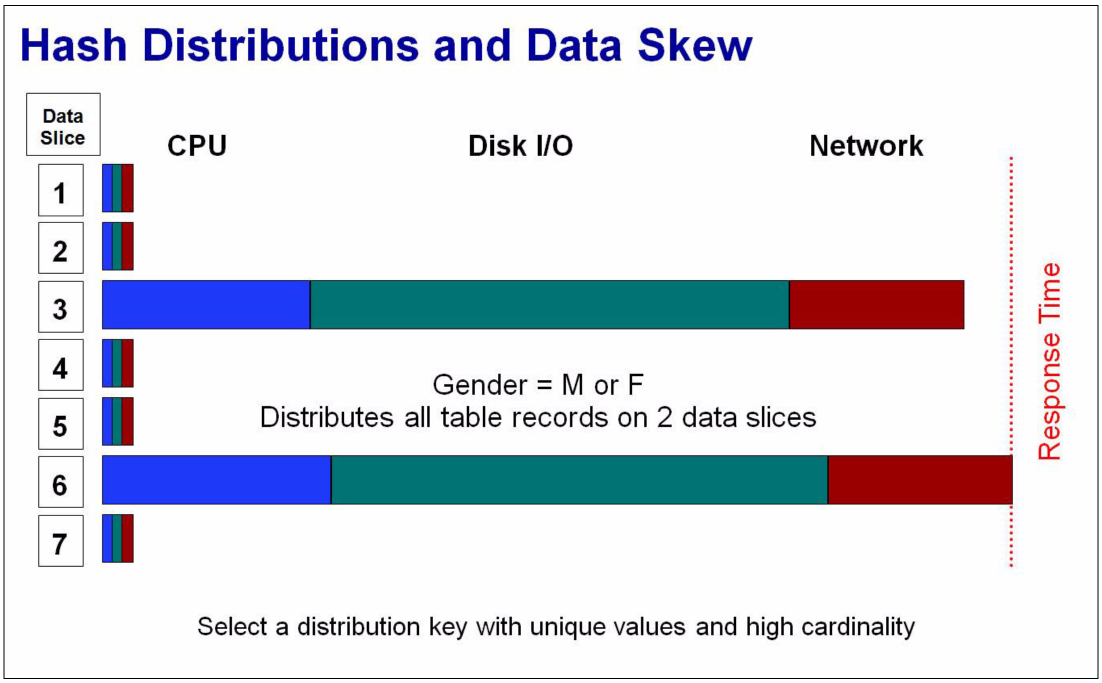


Figure 2-2 Hash distribution and data skew

At query time, only two of the available slices are involved in the query processing, therefore providing the quite unbalanced execution profile shown in the chart in Figure 2-2. This is one of the most inefficient scenarios possible. In practice, binary-value columns should never be used as the distribution key.

More generally, columns with fewer unique values than available slices should also not be used. The following columns are good candidates for distribution keys:

- Columns frequently used as join keys
- Columns frequently aggregated on
- Columns providing even data distribution
- Columns not frequently restricted on
- A few columns combined together (this makes sense only if we join on all of them to another table)

[1] SPUs (or S-Blades) are hardware components that serve as the query processing engines of the PureData for Analytics appliance.

Processing skew

Processing skew happens when a disproportionate amount of the query processing is concentrated in one or more slices, leaving other slices under-used.

In the example shown in Figure 2-3, a DATE column is used as the distribution key. It would appear at first glance that this would make a good key from a data distribution perspective, because over time the data rows for a table would be evenly distributed across available slices. This presumes a relatively consistent number of records for each given date value. A major drawback to using a DATE for the distribution key shows the problem of processing skew.

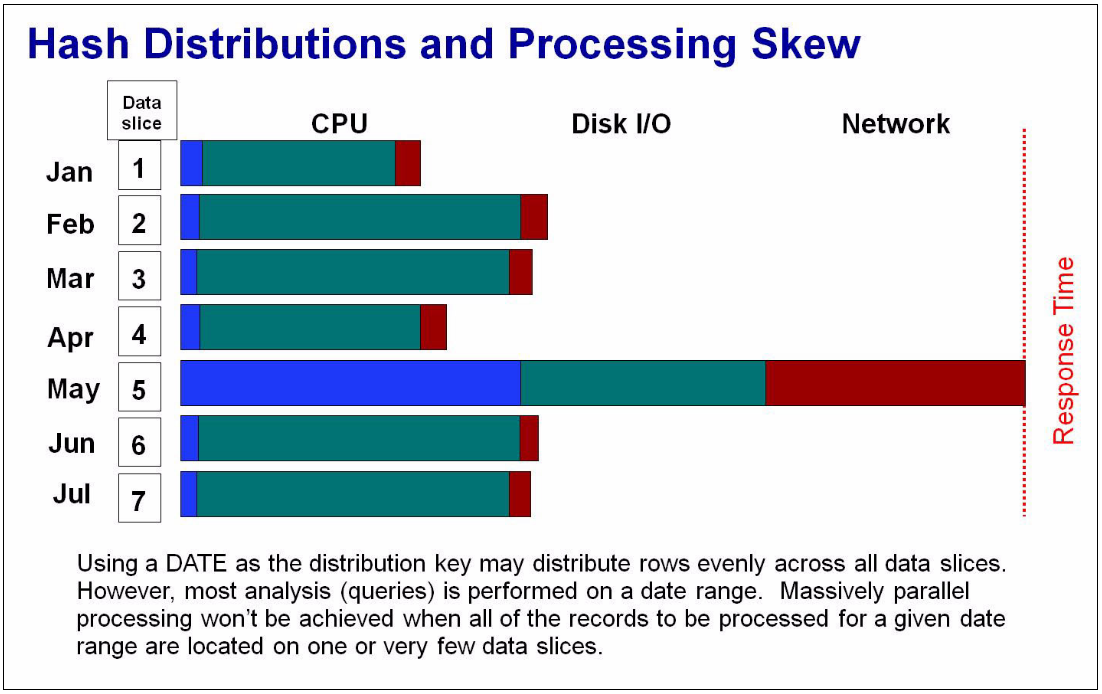


Figure 2-3 Hash distribution and processing skew

This frequently occurs when DATE is used to distribute the data, because data for a given date is concentrated on a single slice. Therefore, any queries that contain DATE predicates selecting data for a single date focus processing on the slice with that day (or few days) worth of data. This leaves the remaining slices idle, except insofar as other tables are joined.

2.1.2 Joins and distributions

Tables are distributed over a potentially large number of data slices on different blades. Therefore, during a join of two tables, there are two possibilities:

- Rows of the two tables that belong together are situated on the same disk, which means that they are co-located and can be joined locally.
- Rows that belong together that are situated on different disks requires that tables must be redistributed at query time. A query using large joins can result in costly data redistribution during join execution when the joined tables are not distributed on the join key.

Joins on randomly distributed tables

Random, of course, basically means round-robin. You have instructed the system to put data all over the place on various data slices. Although a random distribution evenly distributes table rows across all blades, the significant performance gains of co-located joins are lost. To join two tables together, similar records must first be put together (using a broadcast or a redistribute) onto the same data slice, as shown in Figure 2-4.

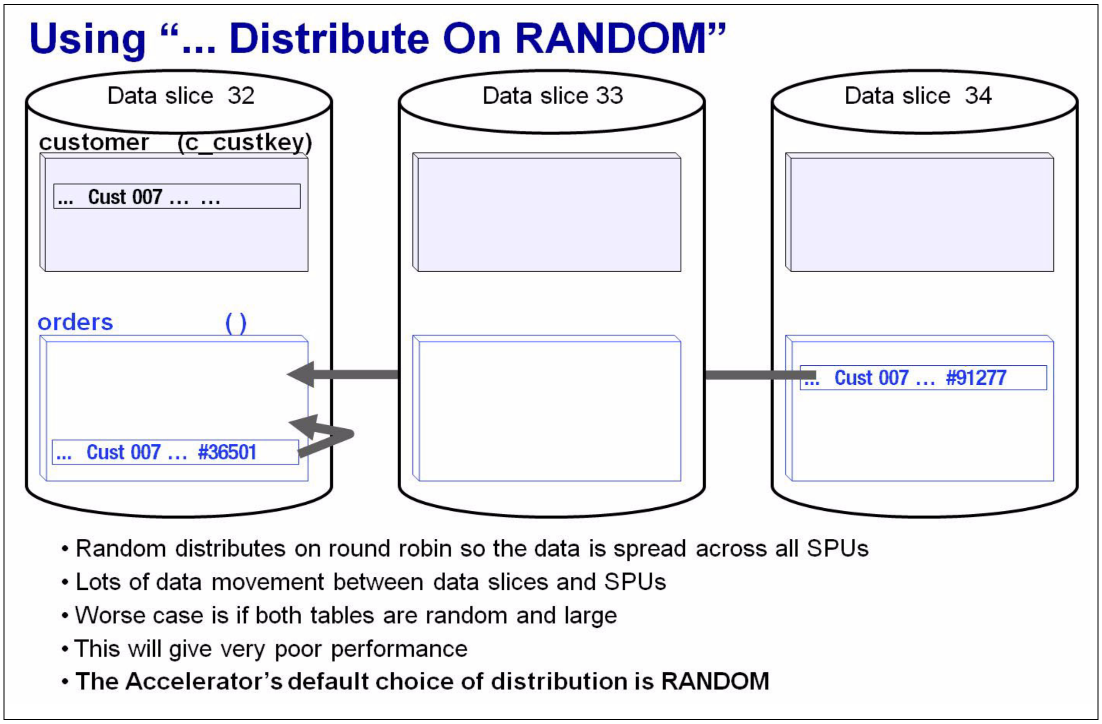


Figure 2-4 Distribute on random

The optimizer decides *which* of the tables are to be moved, and whether to use a **`broadcast`** or a **`distribute`** command. However, data is definitely going to have to be moved around before the join can take place.

If a good distribution key is available to you, use it. Don't choose the DB2 Analytics Accelerator default of `RANDOM` on larger tables just because it is the easy design choice. Query performance can be affected by it.

Co-located tables

When you have chosen a good distribution key, DB2 Analytics Accelerator can then perform a co-located join, which provides the optimal system performance. See Figure 2-5 on page 25.

Each data slice can operate 100% independently of the others, massively parallel and with nothing shared. In this case, the tables are already distributed on the join key.

To join the tables, follow these steps:

1. Read one table and hash it.
2. Read the second table and hash it.
3. Perform the hash join operation.

No communication between the blades is involved. No communication with the host is involved.

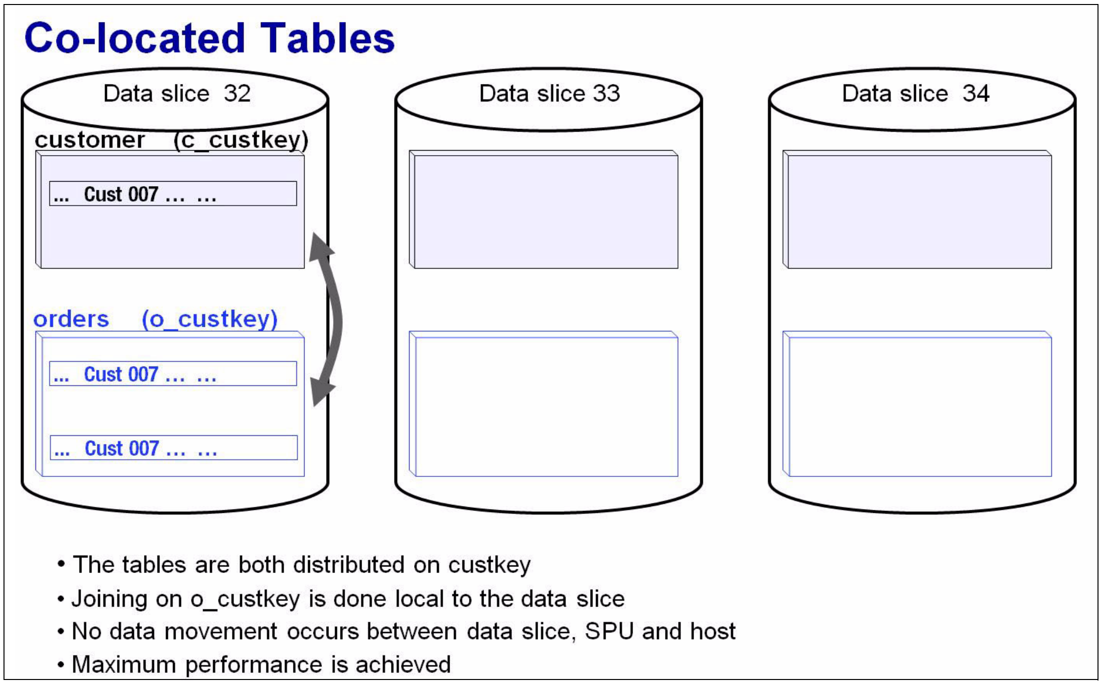


Figure 2-5 Co-located tables

Single table redistribute

To join tables together, matching rows must be located together on the same data slice. Sometimes, your design might preclude certain tables from being distributed on the join key. This is not necessarily bad, but it is not optimal.

In these cases, DB2 Analytics Accelerator automatically redistributes the needed rows and columns from one of the tables on an ad hoc basis. However, there is a performance cost to this that must get paid every time the query is run, because the data must be distributed or redistributed every time, as shown in Figure 2-6 on page 26.

An extra step has been added to the whole process. The system still must perform the basic steps for a co-located join, but now there is an additional step added in, which is the redistribution of the data.

DB2 Analytics Accelerator is able to efficiently handle data redistributions. The SPUs, each operating in parallel, perform the following actions:

1. Scan their portion of the table.
2. Restrict the scan to the records of interest.
3. Project the columns of interest.
4. Determine the destination data slice (which data slice needs this particular record), so that it can be joined to the others.
5. Directly transmit the records across the internal fabric to the other blade and data slice.

Figure 2-6 shows a single table redistribute.

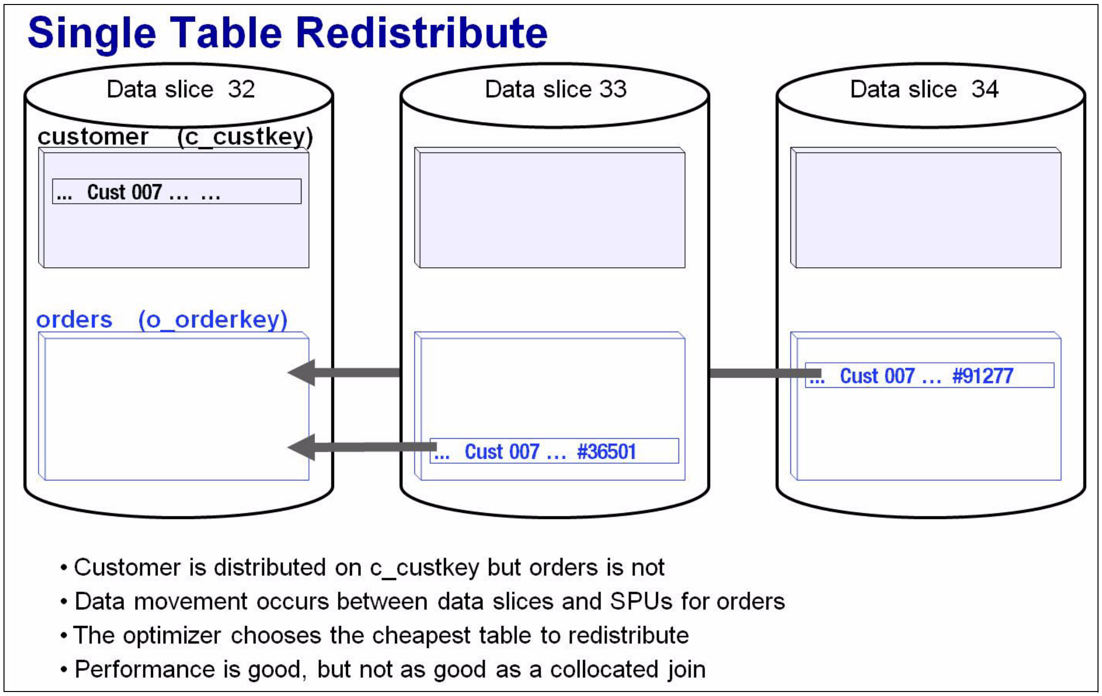


Figure 2-6 Single table redistribute

Joining on different data types

Even though join columns in the two tables might contain the same numeric value and be the distribution keys, if they are different data types, the records are hashed to different data slices because these are mathematically different columns. This must go through the same processing as the "Single table redistribute" on page 25, and incur the same processing usage costs. See Figure 2-7.

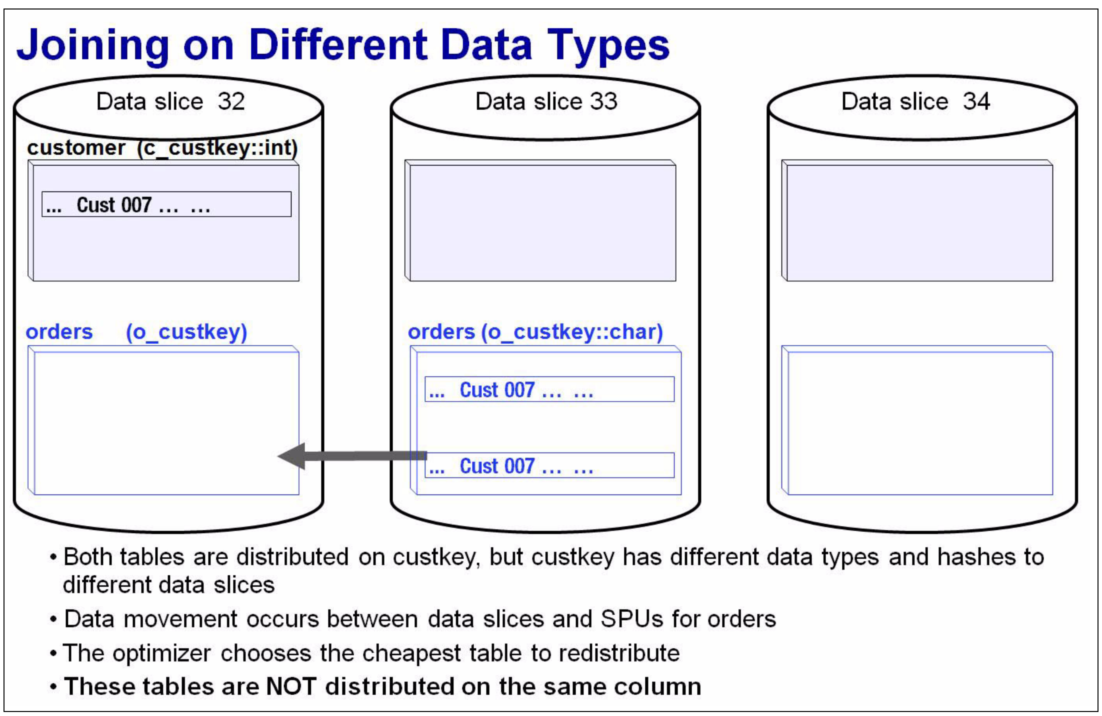


Figure 2-7 Joining on different data types

The database administrator should watch carefully for this situation, because performance effects on large joins can be substantial.

Double table redistribute

In this case, neither table was distributed on the join key that the Structured Query Language (SQL) specified (the **customer key** column). See Figure 2-8.

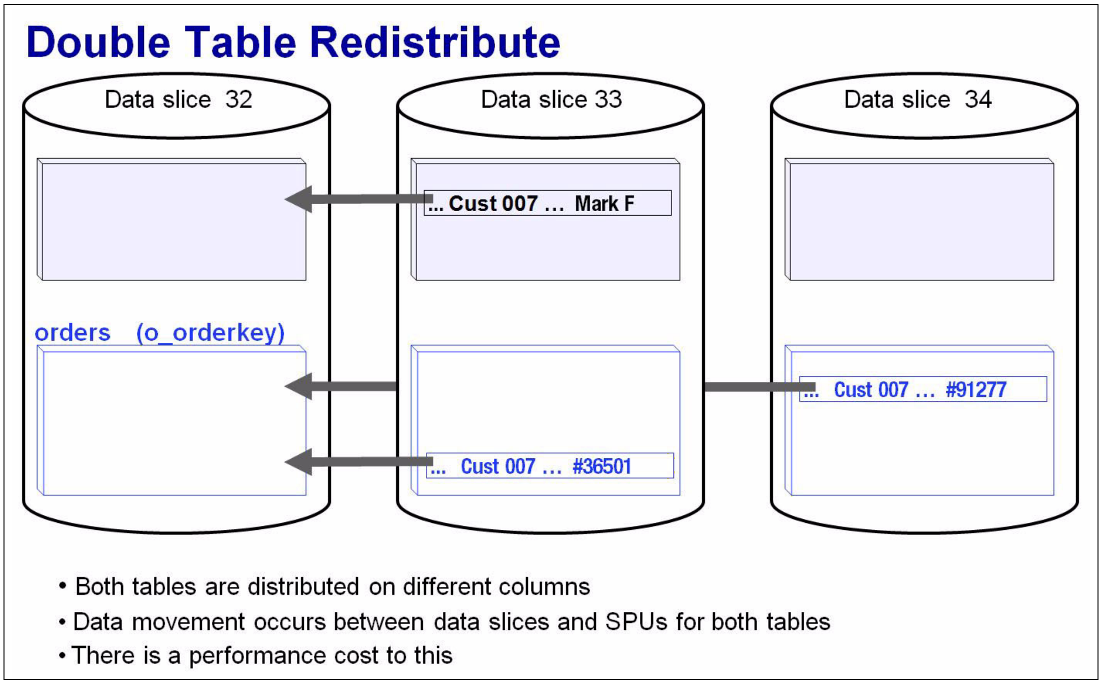


Figure 2-8 Double table redistribute

The system dynamically redistributes only the needed data from both tables before performing the join.

Again, this must happen each and every time this particular join is requested. Therefore, there is a performance cost in having to do this.

Broadcasted table

This join is going to be based on the **nationkey** column. When the nation table was created, it was distributed on that column to begin with.

For the customer table, it was not logical to distribute it on `nationkey`, because it would have resulted in substantial skew (data slices with too much or too little data). The customer table needs to be joined to other tables on other columns. That must be taken into consideration.

Because the customer table is large, rather than redistribute it, NPS instead broadcasts the nation table. See Figure 2-9.

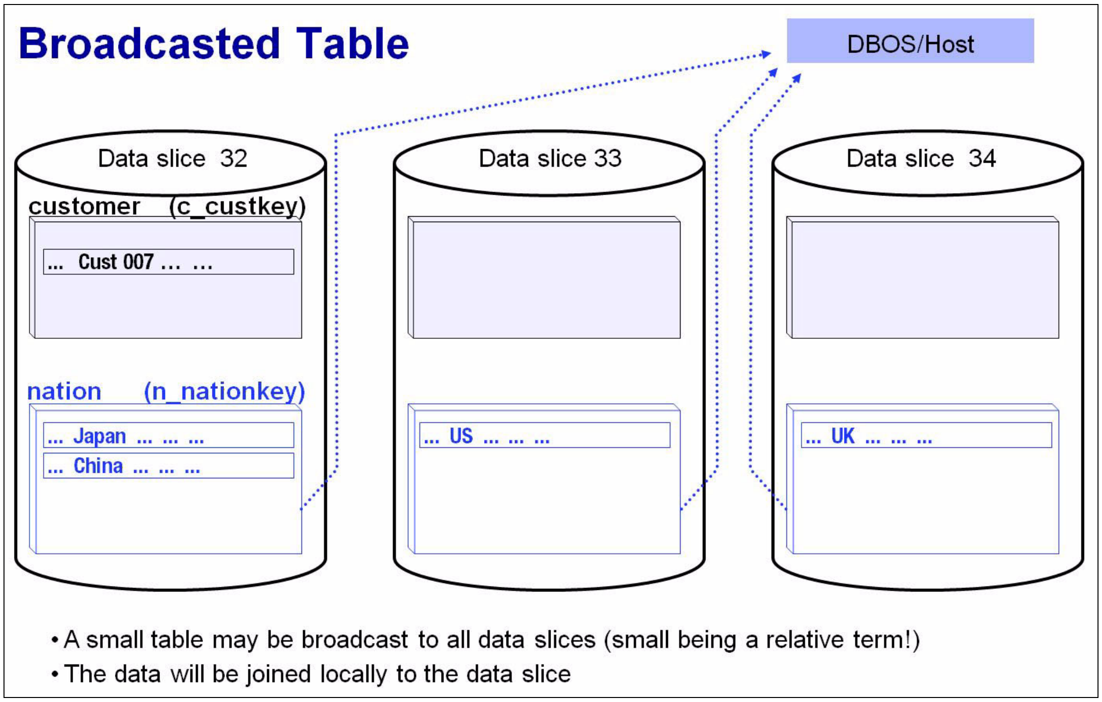


Figure 2-9 Broadcasted table example for a small table

The blades send their individual records from their nation table data slices up to the host, where they are consolidated and returned in full to all of the data slices. Because the data flow must go through the host, this use of the internal network does not scale with the size of the DB2 Analytics Accelerator machine.

2.1.3 General suggestions

The following columns are good candidates for distribution keys:

- Columns frequently used as join keys
- Columns frequently aggregated on
- Columns with high cardinality
- Columns providing even data distribution
- Columns not frequently restricted on (see date example)
- A few columns combined together (makes sense only if we join on all of them)

When choosing a distribution key, consider the following factors:

- The same distribution key value always goes to the same data slice.
- Look at your ON clauses in table joins for candidates for distribution. This enables you to co-locate tables that are commonly joined. Is fact to fact or fact to largest dimension your most painful join? Look to co-locate these.
- INTEGER types provide for optimal performance.
- Never combine columns in a multi-column **`DISTRIBUTE ON (col1, col2, col3)`** in an attempt to generate your own unique key.

- Never add (fabricate) columns to your table and generate unique data values (that is, use a homegrown serial algorithm) in an attempt to generate your own unique distribution key.
- Do not distribute on floating point and Boolean data types.
- Do not **`DISTRIBUTE ON RANDOM`** only because it is the easy or default choice.
- Small reference tables are likely to get broadcasted, so the default of `RANDOM` is usually a good choice for small tables because it forces a broadcast operation.

2.1.4 Example of distribution strategy

Consider a case where the data, logical data model, record counts, and other physical properties are as shown in Figure 2-10 and Table 2-1.

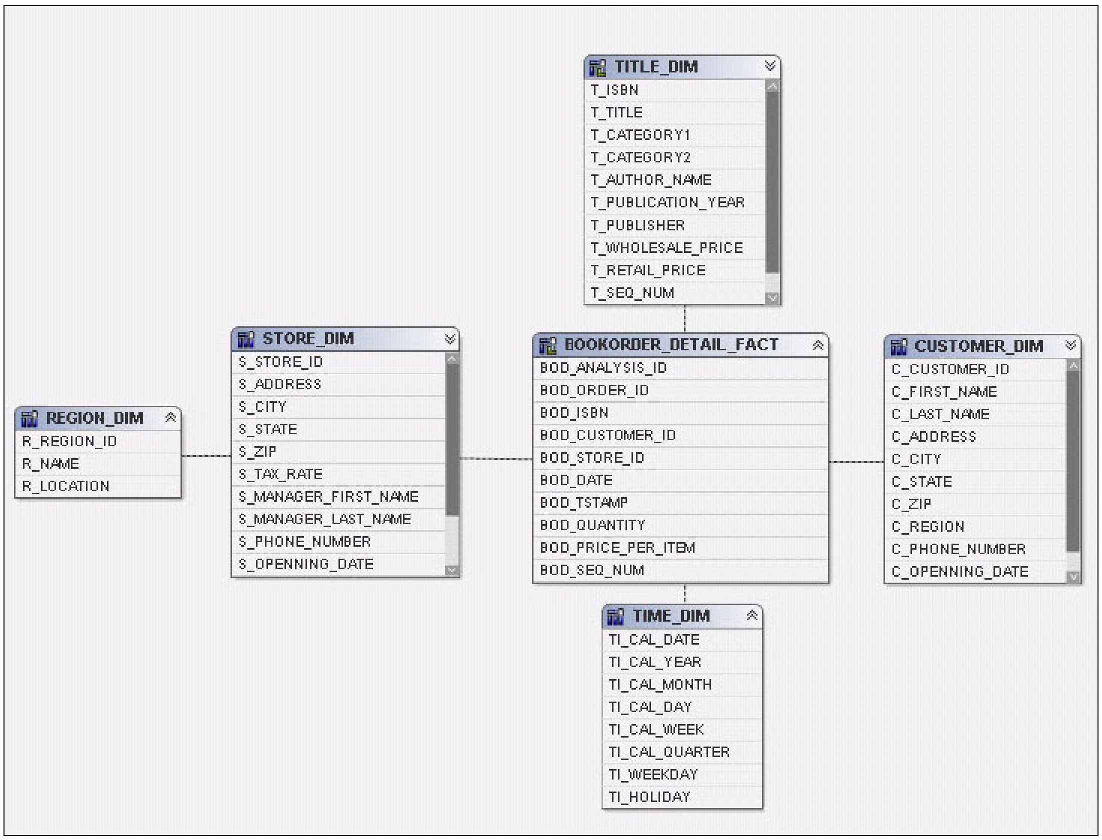


Figure 2-10 Book store data model

Table 2-1 Book store table counts

Table	Record count
BOOKORDER_DETAIL_FACT	9,899,818,991
CUSTOMER_DIM	13,872,410
TITLE_DIM	6,105,440
TIME_DIM	2,557
STORE_DIM	26
REGION_DIM	2

Key physical properties

Also assume the following physical properties of the model are true:

- All ID columns are INTEGER types.
- The first column of each dimension table is the primary key, which joins to a similarly named foreign key on the related tables.
- The primary key for BOOKSTORE_DETAIL_FACT is composed of the first six columns.
- TI_CAL_DATE's foreign key is BOD_DATE, and both are DATE types.
- The largest tables in order are BOOKSTORE_DETAIL_FACT, CUSTOMER_DIM, and TITLE_DIM.
- BOOKSTORE_DETAIL_FACT had approximately the following skew against the foreign key columns:
  - BOD_ANALYSIS_ID: 0%
  - BOD_ORDER_ID: 0%
  - BOD_ISBN: 5%
  - BOD_CUSTOMER_ID: 1%
  - BOD_STORE_ID: 20%
  - BOD_DATE: 1%
- The vast majority of queries are CATEGORY1 and CATEGORY2 grouped analysis against the FACT table, with minimal restriction against TITLE_DIM.
- The CUSTOMER_DIM is frequently restricted on various levels of geographies. When used in joins to the FACT table, surviving tuples are typically far fewer than those of the TITLE_DIM.
- With the exception of the occasional restriction on BOD_ANALYSIS_ID and BOD_DATE, all other restrictions against the FACT table are the result of dimension table joins.

In Table 2-2, we evaluate potential distribution candidates for the FACT table, eliminating unreasonable candidates from further consideration.

Table 2-2 Distribution candidates evaluation

Potential distribution candidate	Should be considered further	Reasoning
RANDOM	No	Although it provides for a perfectly smooth distribution, the FACT table must be fully redistributed for all queries involving joins.
Primary key	No	Although it provides for a perfectly smooth distribution, no single dimension table is distributed on the same 5 columns, making this no better than a random distribution. As a result, the FACT table must be fully redistributed for all queries involving joins.
BOD_ANALYSIS_ID	No	No dimension table joins.
BOD_ORDER_ID	No	Although unique, it has no dimension table joins. All restrictions against BOD_ORDER_ID are either single value or a small number of values. As a result, there are very high processing skews, with only one to a small number of disks responsible for table scan.
BOD_ISBN	Yes	Involved in common joins and has reasonably low skew.

Potential distribution candidate	Should be considered further	Reasoning
BOD_CUSTOMER_ID	Yes	Involved in common joins and has irrelevant skew.
BOD_STORE_ID	No	There are only 150 records in STORE_DIM, and STORE_DIM is typically highly restricted when used in joins with the FACT table.
BOD_DATE	No	All restrictions against BOD_DATE are either single value or a small number of values. As a result, there are very high processing skews, with only one to a small number of disks responsible for table scan.
Various combinations of foreign keys	No	No single dimension table can be distributed on more than one of the FACT table's columns. As a result, the FACT table must be redistributed for all joins.

Of the remaining candidates, BOD_CUSTOMER_ID and BOD_ISBN, we choose the distribution columns that result in a balance of low skew, maximization of join size to take advantage of parallel node-local joins, and likelihood of the join occurring per query. As a result, we choose BOD_ISBN for the following reasons:

- Most queries are anticipated to be merchandising analytics involving the TITLE_DIM and BOOKORDER_DETAIL_FACT tables.
- The vast majority of queries involving TITLE_DIM are CATEGORY1 and CATEGORY2 grouped analysis against the FACT table, with minimal restriction. As such, TITLE_DIM typically has a large number of tuples survive initial scan, creating large joins with the FACT table.
- Skew of BOD_ISBN is rather low (5%).

To benefit from the choice of BOD_ISBN as the distribution key on the FACT table, TITLE_DIM must be distributed on T_ISBN and have the same data type.

Although CUSTOMER_DIM is a larger table, typical restrictions on this table result in relatively small joins with the FACT table. Furthermore, our expectation is that a relatively small subset of queries will involve CUSTOMER_DIM to FACT table joins. So, BOD_CUSTOMER_ID can be eliminated from distribution key consideration.

2.2 Zone maps (automatic partitioning)

Zone maps are built automatically in a DB2 Analytics Accelerator system. Zone maps reduce disk scan operations that are required to retrieve data by eliminating records outside of the start and end range of a `WHERE` clause on restricted scan queries.

As you load, **`UPDATE`**, or **`DELETE`** data in DB2 Analytics Accelerator, the Netezza system is creating 3 megabyte (MB) extents divided into 128 kilobyte (KB) blocks. For each column in each block, DB2 Analytics Accelerator automatically keeps track of the minimum (`MIN`) and maximum (`MAX`) values when you load data, delete data, update data, and so on. Specifically, the values are tracked using statistics, and these statistics are referred to as *zone maps*.

At query time, zone maps are used to scan only relevant data based on column restrictions and dimension-join restrictions. The observed scan performance benefit can be multiple orders of magnitude.

Zone map entries get created and maintained automatically, and that is part of what the FPGA takes into consideration when it is discarding columns and records that are not needed to satisfy your particular query. Again, this is something that happens automatically and in the background, with no negative affect on load/update/delete performance. This removes the need for indexes.

2.2.1 Organization keys (explicit clustering)

Clustered base tables (CBTs) are functionally equivalent to multidimensional clustering (MDC) tables in other databases, such as DB2 for Linux, UNIX, and Windows. If you are familiar with MDC, you have a really good idea about how CBTs function in DB2 Analytics Accelerator.

MDC provides an elegant method for clustering data in tables along multiple dimensions in a flexible, continuous, and automatic way. MDC can significantly improve query performance, in addition to significantly reducing the usage costs of data maintenance operations, such as reorganization, and index maintenance operations during insert, update, and delete operations. MDC is primarily intended for data warehousing and large database environments, and it can also be used in online transaction processing (OLTP) environments.

There are also many applications where it is useful to specify different columns for the MDC key than those on which the table is partitioned. It should be noted that table partitioning is multicolumn, where MDC is multi-dimension.

With CBTs, along with the distribution key for the table, you can also specify an organization key of up to 4 columns. Whatever key you specify as an organization key, DB2 Analytics Accelerator uses them to store, automatically and in the background, your data next to each other on disk in this order. When you load data, the data is placed on disk according to these same organization keys.

In short, organization keys should be used to bias the zone maps on large tables whose queries restrict or join-restrict based on one or more different columns.

The following list includes properties of organization keys:

- On up to four multiple columns (keys), with approximately equal benefit on each irrespective of whether some or all are used in the query predicate.
- No change in storage usage, except perhaps from improved or reduced compression.
- Adapts to the data (no pre-specification of ranges, and so on). Outliers tend to get bunched together.
- Speed up queries against organizing keys.
- Zone map on all data types (including first 8 bytes of character types and numerics up to 18 digits).

Organization keys should be used on large tables. The following list includes example columns:

- Ticker Symbol, Account_ID, Trade_Date
- Purchase Date, Store, Product, Customer_ID
- Latitude and Longitude

How clustering works

DB2 Analytics Accelerator automatically initiates re-clustering of the tables as needed, clustering records in <1 gigabyte (GB) sized chunks using a Hilbert space-filling curve in N-dimensional space. Therefore, predicates on subsets of organized tables have the benefit of multidimensional locality. See Figure 2-11 and Figure 2-12.

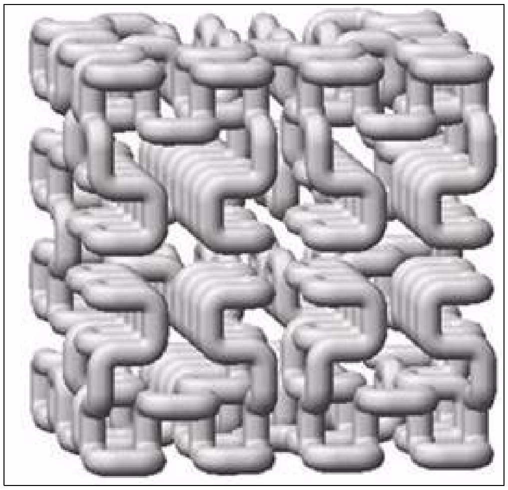


Figure 2-11 Hilbert curve in three dimensions

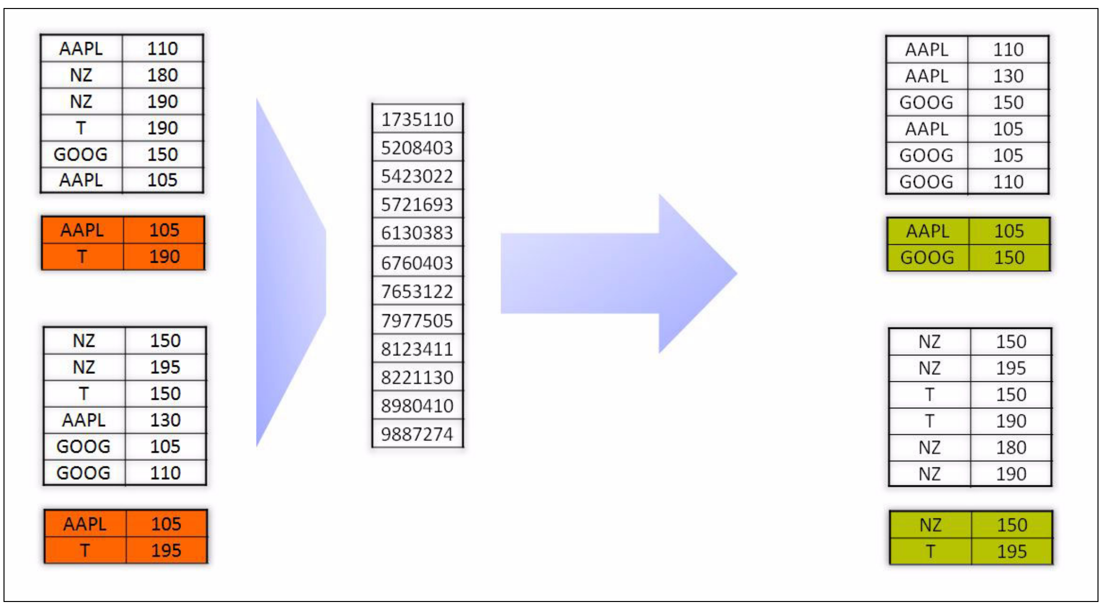


Figure 2-12 Zone maps distribution

Organization key considerations

When selecting an organization key, you should consider the following factors:

- “Organize on (a,b,c)” is exactly the same as “organize on (b,c,a).”
- The scan benefits from restriction on any to all columns in the organization key.
- As more columns are used for organization (up to 4), the disk scan benefit declines.
- Only use organizing keys on tables with millions or more records.
- Organize on columns that are used as common predicates.
- Effective for both low and higher cardinality columns.
- Use on high cardinality columns typically reduces the compression ratio.
- To improve performance of incremental update on target tables, organize on one or more columns of the primary key.

3

Overview of IBM DB2 Analytics Accelerator for z/OS V4.1

This chapter provides information about the new features and updates in IBM DB2 Analytics Accelerator for IBM z/OS (DB2 Analytics Accelerator) V4.1, V4.1 with program temporary fix (PTF)-2, and V4.1 with PTF-3, including the following topics:

- Introduction of DB2 Analytics Accelerator for z/OS V4.1
- SQL enhancements
- Workload management
- Incremental update
- High Performance Storage Saver enhancements
- Monitoring (new counters including incremental update)
- Installation, operations, and maintenance
- DB2 Analytics Accelerator Loader V1.1
- Updates in DB2 Analytics Accelerator V4.1 PTF-2
- Updates in DB2 Analytics Accelerator V4.1 PTF-3
- DB2 Analytics Accelerator tools

3.1 Introduction of DB2 Analytics Accelerator for z/OS V4.1

DB2 Analytics Accelerator for z/OS is a high-performance appliance that integrates IBM Netezza (now PureData System for Analytics) technology with IBM System z hardware and DB2 for z/OS to deliver extremely fast performance to data-intensive and complex DB2 queries for data warehousing, business intelligence (BI), and analytic workloads.

The queries requested under DB2 for z/OS can be run orders of magnitude faster than was previously possible. The performance and cost of DB2 Analytics Accelerator creates unprecedented opportunities for organizations to make use of their data on the IBM zEnterprise platform.

3.1.1 What's new in version 4.1

DB2 Analytics Accelerator V4.1 is enhanced to support a broader set of applications, making acceleration accessible to the vast majority of DB2 for z/OS customers. DB2 Analytics Accelerator V4.1 extends the available uses of the solution to include more data, queries, systems, and applications, to further enhance business benefits and help realize additional cost savings with the solution.

Static SQL support

Enable acceleration of static queries because many DB2 for z/OS customers rely heavily on static Structured Query Language (SQL). This enables customers to route more queries, from a wider base of applications, to DB2 Analytics Accelerator. In addition, the support can eliminate a great deal of the administrative task of identifying tables that have both dynamic and static SQL queries directed at them.

Support for multi-row FETCH operations for local applications

DB2 Analytics Accelerator V4.1 adds support for multi-row fetch (MRF) to enable the acceleration of such queries for local applications, to reduce the processor usage of processing large result sets from DB2 Analytics Accelerator back into DB2.

Support multiple encoding

Customers who use both traditional Extended Binary Coded Decimal Interchange Code (EBCDIC) content and International Unicode content in the same DB2 subsystem can now benefit from DB2 Analytics Accelerator because V4.1 introduces support for both EBCDIC and Unicode in the same DB2 system and DB2 Analytics Accelerator.

Workload balancing and failover support

When multiple accelerators are configured, DB2 Analytics Accelerator V4.1 can perform automated routing, based on workload balancing information, to distribute the workload evenly across the attached accelerators. DB2 Analytics Accelerator V4.1 also delivers failover support. If one accelerator is unavailable for any reason, queries can be automatically rerouted to the available DB2 Analytics Accelerators.

Increased number of replication sources per DB2 Analytics Accelerator

With DB2 Analytics Accelerator V4.1, up to 10 different DB2 subsystems can be replicated into a shared DB2 Analytics Accelerator. DB2 Analytics Accelerator V3.1 can only effectively support two DB2 subsystems.

Better performance for incremental update

Incremental update in DB2 Analytics Accelerator V4.1 supports the DB2 instrumentation facility interface (IFI) filtering provided by DB2 V11, which enables reduced processor use by filtering DB2 log records down to the necessary ones. This support is retrofitted to DB2 V10 as well. In addition, Change Data Capture has been updated in DB2 Analytics Accelerator V4.1 to improve apply throughput.

Support table load while replication is running

In DB2 Analytics Accelerator V3.1, it is necessary to stop replication for all tables of a DB2 subsystem if one replication-enabled table needs to be loaded or re-loaded. With DB2 Analytics Accelerator V4.1, loading of a replication-enabled table can be done while replication of other tables continues.

Enhanced system monitoring

DB2 Analytics Accelerator V4.1 provides expanded information for monitoring applications with the IFI. The separately available IBM Tivoli OMEGAMON® XE for DB2 Performance Expert on z/OS can use this broader information to better inform clients about the usage, execution, and concurrency of queries as they are processed in DB2 Analytics Accelerator.

High Performance Storage System improvements

DB2 Analytics Accelerator V4.1 enables archiving to multiple accelerators, and automates recovering individual partitions that have been archived on an accelerator using stored procedures. This version also enforces a persistent read-only state on partitions that are used by High Performance Storage System (HPSS).

Simplified maintenance

DB2 Analytics Accelerator V4.1 offers an automated NZKit Install function that facilitates the fully automatic installation and activation of Netezza kits, with no firmware or operating system (OS) updates required. It enables the Netezza Performance Server (NPS) kit installation and activation through the same mechanisms as DB2 Analytics Accelerator host software. It also enables clients to perform these maintenance tasks by themselves using the graphical user interface (GUI) or simple stored procedure calls.

3.2 SQL enhancements

In DB2 Analytics Accelerator V4.1, more SQL constructs than before are eligible for routing to DB2 Analytics Accelerator.

3.2.1 Static SQL

Static queries on active or archived data are now routed to DB2 Analytics Accelerator. Acceleration for static queries is determined and fixed at bind package time, and tables must be added and enabled or archived on DB2 Analytics Accelerator before binding to the package. DB2 Analytics Accelerator must be active and started when the query runs, otherwise the execution fails. There are new bind options available, and if the bind options are specified, they set the initial values of the associated special registers.

If you want queries in a static package to be accelerated, you must first add and enable the referenced DB2 tables to DB2 Analytics Accelerator. You **`BIND`** or **`REBIND`** the query's package, specifying the wanted `QUERYACCELERATION` behavior to bind queries in that package for acceleration.

Alternatively, if you want static queries to use data that is archived to DB2 Analytics Accelerator, you must first archive at least one partition of the referenced DB2 tables, and then **BIND** or **REBIND** the package, specifying `GETACCELARCHIVE(YES)`. Also specify the wanted `QUERYACCELERATION` behavior to bind queries in that package for acceleration, and to retrieve archived data. These steps apply any time that you choose queries in a new package to accelerate, or retrieve archived data from DB2 Analytics Accelerator.

This topic is discussed in more detail in 5.6, “Static SQL queries acceleration” on page 101.

3.2.2 ROWSET query offload and MRF for local applications

DB2 Analytics Accelerator V4.1 improves the performance of retrieving the data from a local off-loaded query (static or dynamic) with a very large result table by supporting MRF operations. MRF support enables reduced processor use for retrieving data from a local accelerated query returning a (very) large result set.

For dynamic SQL in local applications, specify `WITH ROWSET POSITIONING` and fetch using a **FETCH NEXT ROWSET** statement with a `FOR N ROWS` clause. To use this support, a local query must use a `WITHOUT RETURN` clause in the **PREPARE** or **DECLARECURSOR** statement, and the query must not be run as part of a SQL Procedural Language (SQL PL) routine.

Additionally, a query must specify `WITH ROWSET POSITIONING` in the **PREPARE** or **DECLARECURSOR** statement, and must use a **FETCH NEXT ROWSET** statement with a `FOR N ROWS` clause to fetch. The row set size must be the same for each **FETCH NEXT ROWSET** statement, and the target host variables must be specified.

This topic is discussed in more detail in 5.5, “Multi-row fetch queries” on page 96.

3.2.3 Additional DB2 functions and data types

With DB2 Analytics Accelerator V4.1, there are more DB2 functions and data types supported. **BITAND()** and **TIMESTAMPDIFF()** are now supported, and there is now functionality that enables comparison between different data types, such as `VARCHAR` and `INTEGER`.

With the updated **BITAND** support, queries using the following functions with `INTEGER`, `SMALLINT`, or `BIGINT` data types are now routed to DB2 Analytics Accelerator: **BITAND**, **BITANDNOT**, **BITOR**, **BITXOR**, and **BITNOT**. However, queries using these functions with `DECIMAL`, `DOUBLE`, `REAL`, or `DECFLOAT` data types are still not routed to DB2 Analytics Accelerator.

For the new **TIMESTAMPDIFF** support, DB2 execution of **TIMESTAMPDIFF** is an estimate, but routing of a query using **TIMESTAMPDIFF** for all intervals returns the correct result. Therefore, different results between DB2 execution and execution on DB2 Analytics Accelerator are expected.

In the case that queries use predicates or functions that compare different data types, DB2 often internally casts the data types to `DECFLOAT`. So far these queries have not been routed to DB2 Analytics Accelerator, but now these queries *are* routed. However, there sometimes might be a loss of precision, because DB2 Analytics Accelerator uses `DOUBLE` rather than `DECFLOAT` for the internal cast.

This topic is discussed in more detail in 5.2, “Query requirements” on page 89.

3.3 Workload management

With DB2 Analytics Accelerator V4.1, workload management updates lead to more efficient use of the DB2 Analytics Accelerator appliance.

This topic is discussed in more detail in Chapter 8, “High availability, disaster recovery, and workload balancing” on page 159.

3.3.1 Workload balancing across multiple accelerators

DB2 Analytics Accelerator V4.1 provides more efficient workload balancing for multiple accelerators. If a query can be run on more than one accelerator, DB2 balances routing between qualifying accelerators, taking into consideration DB2 Analytics Accelerator usage. If multiple accelerators are defined to a DB2 subsystem and are based on the accelerated tables, a query could be routed to multiple accelerators.

In DB2 Analytics Accelerator V3.1, the query was routed to the first accelerator that matched the routing criteria. This method might lead to unbalanced accelerator use. For example, one accelerator might be used heavily, while another is more or less idle. But DB2 now uses accelerator utilization information to route the query to an accelerator.

3.3.2 IBM Worlkload Manager support for local applications

DB2 Analytics Accelerator V4.1 now extends to local queries the already-available IBM Workload Manager for z/OS (WLM) query prioritization support for remote queries. DB2 now detects the service class and importance level for local applications, and sends the WLM importance level on to DB2 Analytics Accelerator with each query.

The service class is detected based on the application’s address space for z/OS Time Sharing Option (TSO), Job Entry Subsystem (JES), and started task control (STC) subsystem types. However, for IBM IMS and IBM CICS® applications, the service class is detected based on past performance block.

3.3.3 Mapping WLM to DB2 Analytics Accelerator

DB2 sends the importance level to DB2 Analytics Accelerator with each query. DB2 Analytics Accelerator maps the importance level to a Netezza priority, and alters the session before query execution using the corresponding priority. These are the same priorities as for remote applications.

However, the changes in prioritization after query start are not reflected. Netezza supports only four different priority levels, therefore multiple WLM importance levels must be mapped against the same Netezza priority. In DB2 Analytics Accelerator V3.1, WLM Importance level 2 mapped to Netezza priority HIGH and levels 3 - 5 mapped to NORMAL.

3.4 Incremental update

Enhancements to incremental update in DB2 Analytics Accelerator include updates to the architecture, the ability to continue replication while loading tables, assured resource allocation, and DB2 IFI filtering.

This topic is discussed in more detail in 6.2, “Incremental update changes” on page 119.

3.4.1 New architecture

Changes to DB2 Analytics Accelerator V4.1 include incremental update support of multiple DB2 subsystems. It is now possible to replicate from up to 10 different subsystems into a shared accelerator. The incremental update feature supports managing multiple Netezza databases by the same replication engine. Using the same replication engine makes it possible to enable up to 10 DB2 subsystems to replicate into one accelerator.

3.4.2 Improved usability (load tables while replication continues)

In DB2 Analytics Accelerator V3.1, it was necessary to stop replication for all tables if one replication enabled table needed to be reloaded, which affected all replicated tables. In DB2 Analytics Accelerator V4.1, reloading of a table can be done while replication of other tables continues. In this case, only the reloaded table is affected. Replication for the table is stopped before the load and started again after the load.

This change supports use cases where DB2 tables are changed by not logged operations, for example, by using the `LOAD` utility. In case of an error during reload, the table is disabled for replication, and needs to be re-enabled by the user.

3.4.3 Improved performance (assured resource allocation for Incremental Update)

The way in which resources are allocated for incremental update (IU) work on DB2 Analytics Accelerator has changed to prevent IU from being under-allocated resources. The replication engine now runs in an independent accelerator resource group. This enables the replication to obtain an assured amount of resources during times of high resource contention.

3.4.4 IFI filtering

Some incremental update users have experienced higher than expected processor usage for the DB2 Analytics Accelerator capture agent running on System z. The apply agent uses DB2 IFI to gain access to change records from the DB2 log.

DB2 V11 and DB2 V10 with authorized program analysis report (APAR) PM90568 provide filtering support, and DB2 capture tasks can qualify a list of database identifier/page set identifier (DBID/PSID) pairs to provide to IFI. Then the DB2 IFI process uses the list to filter the log records, and returns only the log records associated with the qualified objects (in-scope changes).

In-scope changes are then processed and sent to the target apply agent in DB2 Analytics Accelerator. Out-of-scope changes are discarded. With Accelerator V4.1, DB2 IFI does the filtering of out-of-scope changes, thereby reducing processor usage. Further benefit is possible if the source tables are compressed, because DB2 IFI filtering can also avoid the unnecessary decompression process for the log records that are filtered out.

3.5 High Performance Storage Saver enhancements

DB2 Analytics Accelerator V4.1 includes updates to the HPSS, including enhancements to archiving, the restoration of archived partitions, and archiving on multiple accelerators.

This topic is discussed in more detail in 7.1, “HPSS overview” on page 138.

3.5.1 Archiving enhancements

In DB2 Analytics Accelerator V4.1, DB2 partitions are set to persistent read-only (PRO) status (after the partition is moved to DB2 Analytics Accelerator) to prevent future `INSERT` or `UPDATE` actions on the partitions. Partitions to be archived are first backed up, then old partitions are deleted from DB2, and the table is split on DB2 Analytics Accelerator.

Old partitions and image copies are sent to read-only status, and applications have transparent access (no SQL statement changes needed) to the table. Created image copies are protected by the PRO status of the DB2 partitions. This feature improves disaster recovery (DR) scenarios. Additionally, a flexible naming schema using templates for the image copies can be defined.

3.5.2 Restore archived partitions

DB2 Analytics Accelerator V4.1 includes a new stored procedure that automates the process to change one or more partitions from accelerated archived partitions to accelerated partitions, stored in DB2 and on DB2 Analytics Accelerator. This restore is done from image copy data, and is useful to update the data and archive again. It is also possible to restore partitions using IBM Data Studio.

To use the new stored procedure to restore data in DB2 for z/OS, all attached accelerators need to be online.

3.5.3 Archive a table on multiple accelerators

In DB2 Analytics Accelerator V3.1, an archived table can only exist on a single accelerator, which prevents high availability (HA) and DR scenarios. In DB2 Analytics Accelerator V4.1, however, a table can be archived on multiple accelerators. This is accomplished through an archive step per accelerator rather than a single step for multiple accelerators. To archive a table from image copy to an accelerator, at least one accelerator having this table already archived must be active.

3.6 Monitoring (new counters including incremental update)

Enhanced monitoring is available on DB2 Analytics Accelerator V4.1 through a set of new and revised monitoring counters on the system level that are made accessible through IFI. This provides better support of charge-back, capacity planning, monitoring, and problem determination. System-level statistics counters are made available using DB2 statistics, and include the following areas:

- Query execution statistics
- Central processing unit (CPU) costs
- Incremental update (IU) processing
- System state
- System capacity
- System usage

The following list includes examples of these system-level counters:

- Query execution statistics
  - Total number of successful queries sent by all DB2 systems
  - Total number of failed queries sent by all DB2 systems

  - Number of currently running queries
  - Max number of concurrently running queries
- CPU costs
  - Total CPU costs of running queries sent by this or all DB2 systems
  - Total CPU costs of maintenance operations from this or all DB2 systems (for example, **LOAD**, **ARCHIVE**, and **RESTORE**)
- Incremental update processing
  - Total CPU cost associated with the replication apply processes
  - Total numbers of log records read by the replication capture agent
  - Total number of log records processed by the capture agent applicable to tables in DB2 Analytics Accelerator

All “Total” counters in the previous list are available for this DB2 system, or all DB2 systems.

This topic is discussed in more detail in Chapter 9, “Monitoring enhancements” on page 187.

3.7 Installation, operations, and maintenance

Various enhancements to DB2 Analytics Accelerator V4.1 encompass updates to installation, operations, and maintenance, which affect usability and other improvements.

3.7.1 Integrated NPS installation

With DB2 Analytics Accelerator V4.1, the most common Netezza software updates can be done through a stored procedure or IBM Data Studio in the same way as DB2 Analytics Accelerator software updates are already done. In general, a new DB2 Analytics Accelerator release requires a Netezza Performance Server (NPS) kit upgrade. Previously, for NPS kit upgrades, an IBM service engineer was required at the client site to log on to the appliance using a service password.

With DB2 Analytics Accelerator V4.1, the required NPS kit software is delivered as part of the software package that you transfer to the z/OS system to upgrade from a previous version. From there, it can be applied to the DB2 Analytics Accelerator server using the same mechanism as DB2 Analytics Accelerator software. Alternatively, IBM InfoSphere Change Data Capture (InfoSphere CDC) software is applied, by calling a stored procedure or by using IBM Data Studio.

At this time, Netezza Firmware Diagnostics and Tools (FDT) and Netezza Host Platform (HPF) upgrades still require a service engineer, but this circumstance is rare.

This topic is discussed in more detail in Chapter 4, “Installation and maintenance procedures” on page 49.

3.7.2 Multiple code page support in DB2 Analytics Accelerator

Mixed EBCDIC and UNICODE tables are allowed on DB2 Analytics Accelerator V4.1 for the same DB2 subsystem. However, queries that combine both EBCDIC and UNICODE tables still cannot be routed.

This topic is discussed in more detail in 5.2, “Query requirements” on page 89.

3.7.3 Fine-grained access of DB2 Analytics Accelerator control functions

DB2 Analytics Accelerator V4.1 offers fine-grained access control for the **ACCEL_CONTROL_ACCELERATOR** stored procedure. This offers several functions to control DB2 Analytics Accelerator:

- Canceling a task
- Starting replication
- Collecting trace

Access can now be controlled separately for each of these functions, to control who can run each control task. DB2 Analytics Accelerator V4.1 now calls separate SQL-bodied scalar functions for each control task that it provides.

3.7.4 DB2 Analytics Accelerator Studio functional and usability enhancements

DB2 Analytics Accelerator V4.1 Studio includes a new Storage Saver Restore wizard, in addition to an improved, scalable Add Tables wizard that can handle larger DB2 catalogs. The Help Contents menu item now contains all of the DB2 Analytics Accelerator manuals, and the task launcher now contains an Accelerator tab linking to DB2 Analytics Accelerator tutorials and videos. New displays include load progress, which is displayed during load, and a visual explanation that displays the reason for the “not accelerated” message.

3.7.5 Sub-capacity licensing

New sub-capacity licensing for DB2 Analytics Accelerator V4.1 provides for growth on demand. A full rack N200x can be obtained while only paying for 50%, 62.5%, 75%, or 87.5% capacity. The licensee must not use more than the defined capacity, and it is the client’s responsibility to monitor and document the actual usage. The system does provide the interfaces to set and monitor resource limits.

For sub-capacity licensing, capacity is considered to be a combination of data storage used as a percentage of the total available physical storage, and performance settings as a percentage of maximum resource allocation. This can be specified using the **set resource limits for DB2 subsystems** menu item on DB2 Analytics Accelerator.

3.8 Updates in DB2 Analytics Accelerator V4.1 PTF-2

Several enhancements are provided in this Fixpack in the areas of incremental update, DB2 Analytics Accelerator Studio, query history, stored procedures, and installation. The major changes are described in this section. For more detailed information, see the following website:

http://www-01.ibm.com/support/docview.wss?uid=swg27041549

Stored procedure for migration

The **SYSPROC.ACCEL_MIGRATE** stored procedure is introduced to help with migration by adjusting table names in accordance with the new naming schema. This change is required for workload balancing and the acceleration of static SQL queries.

TIME and TIMESTAMP

Support was added for TIME and TIMESTAMP columns containing 24:00:00 values. Conversion is automatically done by DB2.

FOR BIT DATA

Support was added for FOR BIT DATA values in single-byte character EBCDIC tables.

Cast problems

DB2 Analytics Accelerator V4.1 PTF-2 functionality reduces cast problems related to having character data in EBCDIC and Unicode tables located on the same accelerator.

Template support

Template support was added for temporary data sets created by HPSS.

3.9 Updates in DB2 Analytics Accelerator V4.1 PTF-3

Besides corrections, PTF-3 provides enhancements in the load, reload, and incremental update areas.

For details about these enhancements, see the following website:

http://www-01.ibm.com/support/docview.wss?uid=swg27042433

Improve full table reload performance

The time needed to reload an entire table is reduced by dropping the table with the old data. This enhancement applies to full reloads of partitioned tables and non-partitioned tables.

Software version of incremental update components

The version numbers of the DB2 Analytics Accelerator server and replication engine are always displayed in the About section of the DB2 Analytics Accelerator view, even if incremental updates have not been enabled for the connected DB2 subsystem.

Load improvements

Table load performance has been improved starting with NPS V7.1.0.2.

Trace configuration for stored procedures

The enhancement allows the definition of a default global trace behavior for the stored procedures. The default setting is used unless the caller explicitly specifies a different trace configuration in the MESSAGE input parameter.

To support the enhancement, a new optional Data Definition (DD) card, AQTDEFTR, has been introduced for the WLM address space of the stored procedures. The DD card references a data set with and XML definition of the default trace configuration. An example of this setting can be found in the AQTSAMP(AQTDEFTR) member.

3.10 DB2 Analytics Accelerator tools

Starting with the versions mentioned in the following list, these IBM DB2 tools can deliver value to clients using the IBM DB2 Analytics Accelerator for z/OS V4.1:

- DB2 Administration Tool V11.1
- DB2 Analytics Accelerator Loader V1.1
- IBM InfoSphere Optim™ Query Workload Tuner V4.1.0.1
- Tivoli OMEGAMON XE for DB2 Performance Expert V511 and V520
- DB2 Query Monitor for z/OS V3.2

For an introduction to DB2 tools, see the following website:

http://www.ibm.com/support/knowledgecenter/zosbasics/com.ibm.imz.doc.dmtools/tools_db2intro.html?lang=en

The following practices can help you succeed:

- Identify the most expensive threads, plans, packages, and queries over a period of time.
- Estimate savings and compare costs for the workload during various times (accelerated and optimized for greater acceleration).
- Demonstrate management capabilities using IBM Interactive System Productivity Facility (ISPF).
- Load tables in parallel to both DB2 and DB2 Analytics Accelerator.
- Determine the most advantageous set of objects and queries for acceleration.

3.10.1 DB2 Administration Tool V11.1

This tool has been extended to include support for DB2 Analytics Accelerator. You can customize parameters for use with DB2 Analytics Accelerator using DB2 Administration Tool:

- Managing accelerators:
  - You can display information about DB2 Analytics Accelerators that are connected to your DB2 data server.
  - You start an Accelerator using the Start Accelerator panel.
  - You stop an Accelerator using the Stop Accelerator panel.
  - You use the Add Accelerator panel to add an Accelerator.
  - You delete Accelerators using the **`delete`** command.
  - You can reduce DB2 storage space by archiving table partitions to the DB2 Analytics Accelerator.
  - You can create reports that show details for each accelerated table, including change and archive information for the entire table or, if it's a partitioned table, for each part separately.

- Managing accelerated tables:
  - You use DB2 Administration Tool V11.1 to display, add, load, enable, disable, delete, view, and archive accelerated tables for DB2 Analytics Accelerator.
  - You can display information about accelerated tables, which are tables that are associated with the DB2 Analytics Accelerators.
  - You add an accelerated table using the Add Accelerated Table panel.
  - You must load a table with data after its definition has been copied to DB2 Analytics Accelerator.
  - You can enable or disable an accelerated table to enable or disable query offloading for that DB2 table.
  - When you modify data in an accelerated table, you can specify whether to automatically detect and reload the accelerated table. This is useful if you insert, delete, or update records in an accelerated table.
  - You can enable incremental updates to accelerated tables to automatically update tables on a DB2 Analytics Accelerator.
  - You can issue a line command to delete DB2 tables from DB2 Analytics Accelerator (that is, remove it from the accelerated tables), so that query offloading can be disabled for those DB2 tables.
  - You can archive a table partition to DB2 Analytics Accelerator so that DB2 stores only active data, and archive data is moved to DB2 Analytics Accelerator to reduce used DB2 storage space.
  - You can view RUNSTATS and real-time status information for accelerated tables to help you decide whether to reload the accelerated table.

For more information about DB2 Administration Tool V11.1, see the following website:

http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/topic/com.ibm.db2tools.adb111.doc.ug/topics/adbu_idaa_using.htm

3.10.2 DB2 Analytics Accelerator Loader V1.1

DB2 Analytics Accelerator Loader for z/OS, V1.1 (5639-OLA) enables you to load operational DB2 for z/OS data into DB2 Analytics Accelerator. This minimizes the effect on the application when loading related tables into DB2 Analytics Accelerator. It also gives you the flexibility to specify a historical point-in-time to which you want a consistent set of data to be loaded into DB2 Analytics Accelerator to meet your business needs.

You can also load non-DB2 data to DB2 Analytics Accelerator, including non-mainframe data. Loading data into both DB2 Analytics Accelerator and DB2 for z/OS is performed in parallel, which can improve availability and reduce system resource use. It is also possible to load only DB2 Analytics Accelerator, leaving the DB2 table empty.

Note that the DB2 Analytics Accelerator Loader is a separately purchased product. We provide information about this topic in Chapter 10, “DB2 Analytics Accelerator Loader” on page 213.

For more information about DB2 Analytics Accelerator Loader, see the following website:

http://www-03.ibm.com/software/products/en/db2-analytics-accelerator-loader-for-zos

3.10.3 InfoSphere Optim Query Workload Tuner V4.1.0.1

InfoSphere Optim Query Workload Tuner helps database administrators (DBAs) and SQL developers create better-performing SQL statements and query workloads for applications that query DB2 for z/OS subsystems.

The Workload Analytics Acceleration Advisor enables DBAs to decide which tables that are referenced in a workload should be added to accelerators in DB2 Analytics Accelerator.

For more information, see the following website:

http://pic.dhe.ibm.com/infocenter/dstudio/v4r1/index.jsp?topic=%2Fcom.ibm.datatools.qrytune.nav.doc%2Ftopics%2Fhelpindex_qt.html

3.10.4 Tivoli OMEGAMON XE for DB2 Performance Expert V511 and V520

You can use Tivoli OMEGAMON XE for DB2 Performance Expert and DB2 Performance Monitor for z/OS to analyze, monitor, and tune the performance of DB2 and DB2 for z/OS applications. As the primary tool to investigate DB2 accounting and statistics tracing, both batch and online, it has supported DB2 Analytics Accelerator since the product was introduced.

For more information, see the following website:

http://www-01.ibm.com/support/knowledgecenter/OMXEDB2PM520/com.ibm.omegamon.xe.pm_db2.doc_5.2.0/ko2welcome_pm.htm?lang=en

3.10.5 DB2 Query Monitor for z/OS V3.2

DB2 Query Monitor for z/OS is a tool that gives you the ability to efficiently customize and tune your SQL workload and DB2 objects to ensure the effectiveness of your DB2 subsystems, and to improve overall performance.

DB2 Query Monitor indicates whether a query ran in DB2 Analytics Accelerator from within the **View Summaries** → **Optional Summaries** panels (plan, DB2, DBRM/package, and AuthID).

For more information, see the following website:

http://www-01.ibm.com/support/knowledgecenter/SSAURY_3.2.0/com.ibm.db2tools.cqm32.doc.ug/topics/cqmutsk_dispidaainfo.htm

4

Installation and maintenance procedures

This chapter provides a series of considerations regarding installing, maintaining, and debugging the IBM DB2 Analytics Accelerator for IBM z/OS version 4.1 (DB2 Analytics Accelerator).

This chapter contains the following topics:

- Upgrading DB2 Analytics Accelerator to version 4.1
- Optional migration steps
- Handling error situations
- Optimizing DB2 Analytics Accelerator load strategy
- Tuning distribution and organizing keys

4.1 Upgrading DB2 Analytics Accelerator to version 4.1

To benefit from the current features and defect fixes, and to prevent running into known defects, it is suggested to upgrade DB2 Analytics Accelerator software to the most recently available program temporary fix (PTF) level. At the time of writing, the current released version of DB2 Analytics Accelerator is version 4.1.2 (V4.1 PTF2).

The IBM DB2 Analytics Accelerator for z/OS V4.1.0 Release Notes can be found on the following website:

http://www.ibm.com/support/docview.wss?uid=swg27040452

The upgrade process can be divided into the following steps:

1. Checking the prerequisites
2. Installing Netezza Host Platform (HPF) and Netezza Firmware Diagnostics and Tools (FDT)
3. Estimating the upgrade time window
4. Installing DB2 Analytics Accelerator
5. Rolling migration in an HA and disaster recovery environment

4.1.1 Checking the prerequisites

The first step when planning to upgrade to version 4.1 of DB2 Analytics Accelerator software is to check the prerequisites. Because DB2 Analytics Accelerator consists of multiple components, it is important to upgrade all of them to a compatible level, so that the appliance as a whole can function properly.

The document *Prerequisites and Maintenance for IBM DB2 Analytics Accelerator for z/OS Version 4.1*, which contains the up-to-date list of prerequisites and options for running DB2 Analytics Accelerator V4.1, can be found on the following website:

http://www.ibm.com/support/docview.wss?uid=swg27039487

For details about the recent PTF-3, see the following website:

http://www.ibm.com/support/docview.wss?uid=swg27042433

Figure 4-1 lists all of the software components of DB2 Analytics Accelerator.

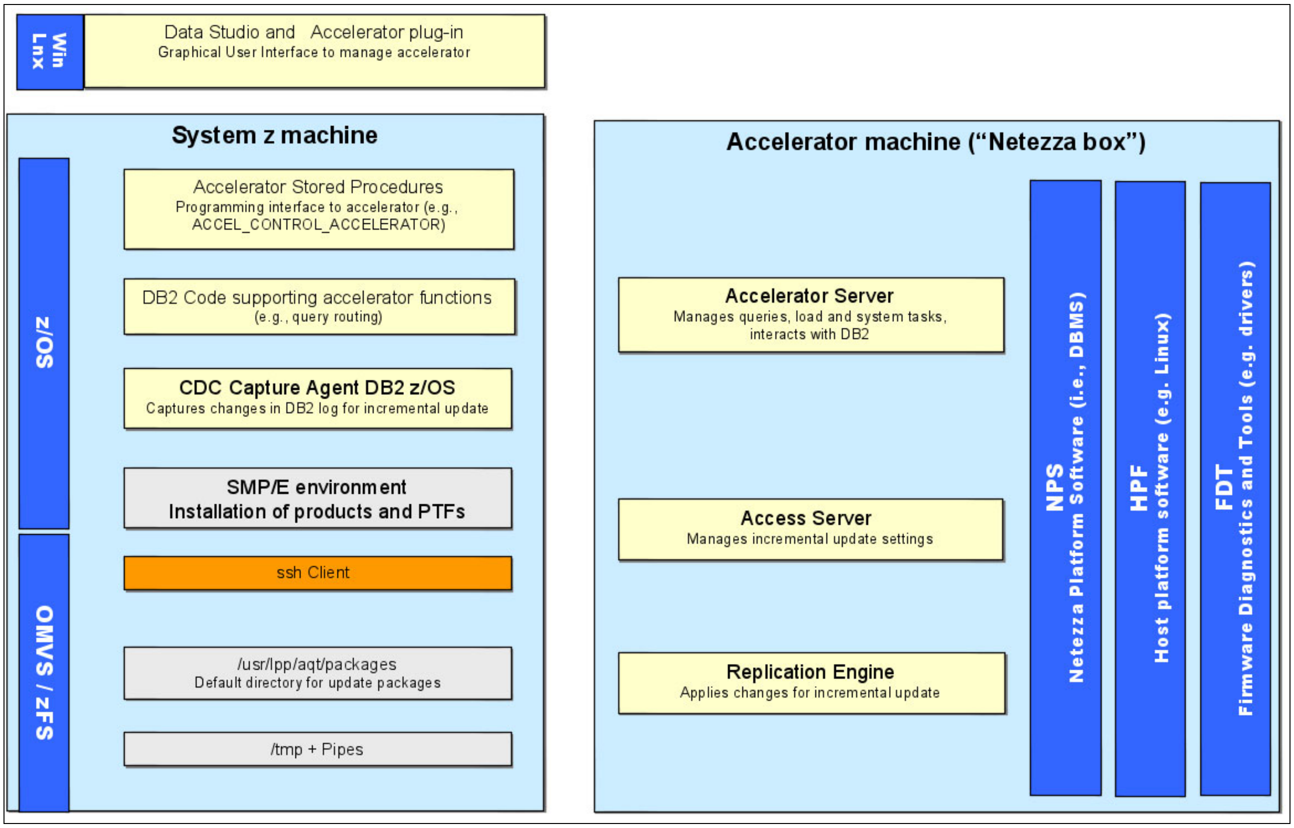


Figure 4-1 Software components of DB2 Analytics Accelerator

Depending on whether the appliance is still running V2.1 of DB2 Analytics Accelerator software, it is necessary to first upgrade to V3.1 PTF1 and then to V4.1. A skip-level migration from DB2 Analytics Accelerator V2.1 to V4.1 is not supported.

Figure 4-2 shows the shortest upgrade paths for a given source version of DB2 Analytics Accelerator.

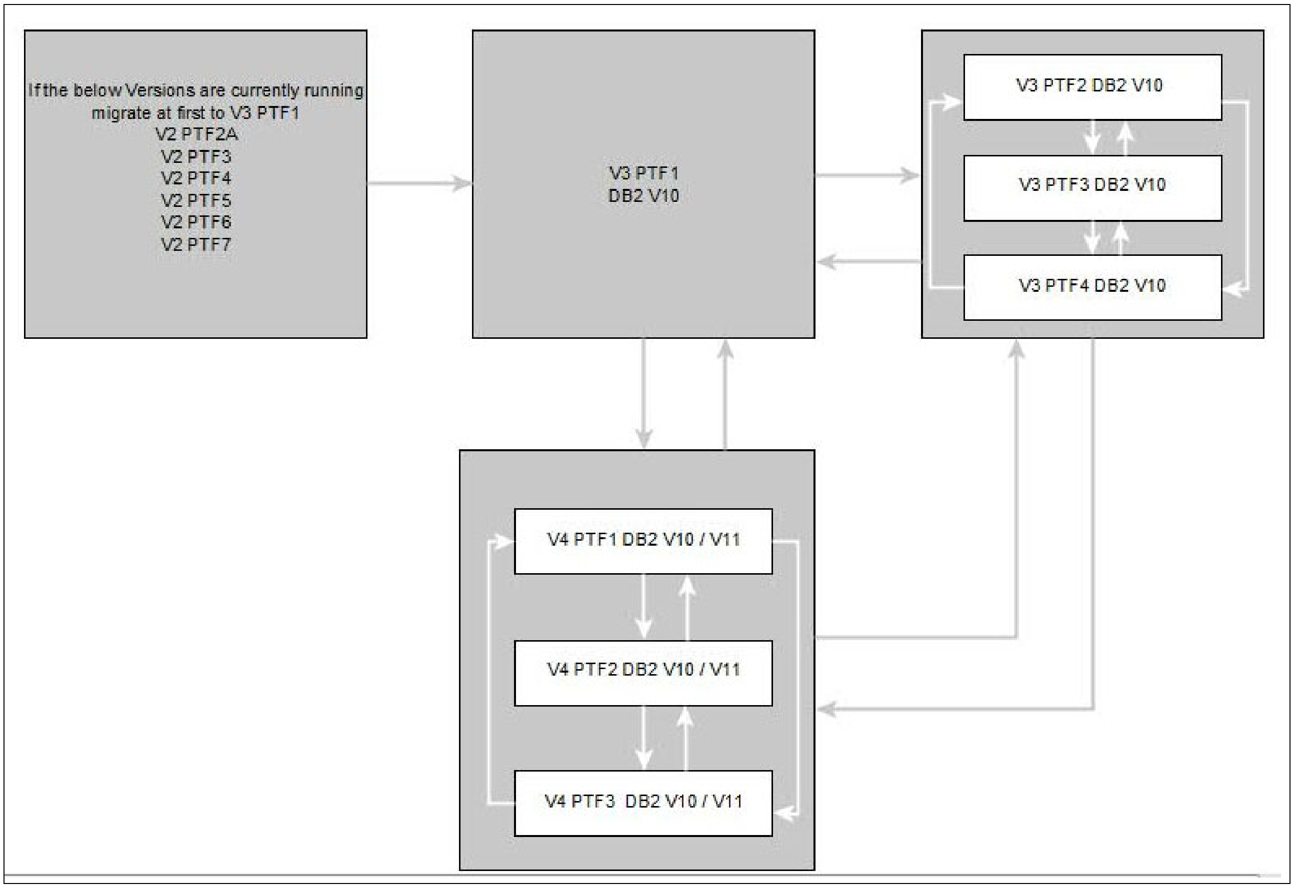


Figure 4-2 Upgrade paths

DB2 Analytics Accelerator V4.1 has the following software prerequisites:

- DB2 for z/OS, including required PTFs and features
- IBM InfoSphere Change Data Capture (InfoSphere CDC) for z/OS, including required PTFs and features
- DB2 Analytics Accelerator V4.1, including required PTFs
- IBM DB2 Analytics Accelerator Studio

4.1.2 Installing Netezza Host Platform and Netezza Firmware Diagnostics and Tools

The Netezza Performance Server (NPS) is part of the DB2 Analytics Accelerator PTFs, and has its own prerequisites to specific minimum versions of the Netezza HPF[1] software and Netezza FDT[2].

The installation of Netezza HPF and Netezza FDT might take several hours, and needs to be conducted by IBM Support due to the appliance nature of the product.

[1] The Netezza HPF software contains kernel updates, firmware, and other special software used by the Netezza host servers. The Netezza HPF upgrade and rollback process includes extensive error-checking, backups, and logging.

[2] Netezza FDT contains the firmware updates for the hardware components of the Netezza system.

If the currently active version of DB2 Analytics Accelerator is version 3.1 or earlier, it is not possible to check the Netezza HPF and Netezza FDT versions as a user, but a problem management record (PMR)[3] must be opened, and IBM Support must verify the currently installed versions before the upgrade.

Starting with DB2 Analytics Accelerator version 4.1, it is possible to determine the Netezza HPF and Netezza FDT versions from DB2 Analytics Accelerator Studio.

Both versions are shown in the **About** section of DB2 Analytics Accelerator, in addition to other software versions, as shown in Figure 4-3[4]. If an upgrade is required, IBM Support schedules a meeting to perform the upgrade. During the course of the meeting, IBM Support upgrades Netezza HPF and Netezza FDT to the most recently available levels.

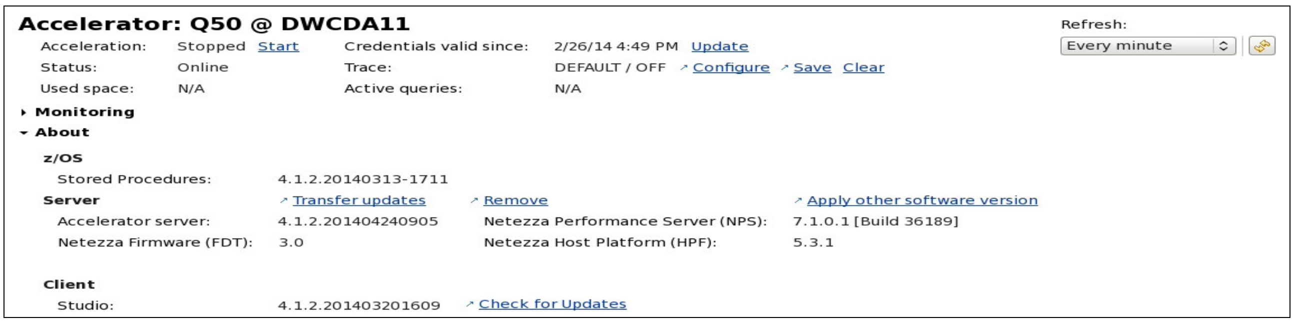


Figure 4-3 Accelerator view showing facilities to check the currently active software versions, and to transfer and apply new software

In case the system in question already fulfills the Netezza HPF and Netezza FDT requirements, NPS can be upgraded using the “Automated transfer and installation (apply action) of updates for the Netezza Performance Server (NPS)” DB2 Analytics Accelerator V4.1 feature during the installation phase, as described in 4.1.4, “Installing DB2 Analytics Accelerator” on page 55.

4.1.3 Estimating the upgrade time window

After getting a clear picture of the missing prerequisites, it is possible to estimate the update time window, (that is, the time that it takes to upgrade DB2 Analytics Accelerator to V4.1). This information can be helpful to plan an outage. Table 4-1 shows conservative estimated times per component.

Table 4-1 Time windows for update to V4.1

Component	**Time**	**Upgrade needed**
	Prerequisites	
DB2 prerequisite PTFs	1 hour	X
DB2 Analytics Accelerator Studio	1 hour	X
Function modification identifier (FMID) HAQT410 and HCHCA21	1 hour	X

[3] https://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/index.jsp?topic=%2Fcom.ibm.datatools.aqt.doc%2Finstallmanual%2Ftask%2Ft_idaa_inst_db2_ptf_sps.html

[4] Notice that the Access Server and Replication Engine versions are only shown for DB2 subsystems with the incremental updates feature enabled.

Component	Time	Upgrade needed
IBM DB2 Analytics Accelerator V4.1 PTF2	1 hour	X
Transfer Packages to DB2 Analytics Accelerator	0.5 hour	X
Netezza HPF	1 hour	
Netezza FDT	6-10 hours	
Installation		
Incremental Update Access Server	0.5 hour	X
Incremental Update Replication Engine	0.5 hour	X
DB2 Analytics Accelerator Server	1 hour	X
Netezza Performance Server	1 hour	
DP2 Analytics Accelerator DB2 stored procedures	0.5 hour	X
IBM InfoSphere Change Data Capture for z/OS	1 hour	X
Verification		
Loading a sample table and offloading a query	0.5 hour	
Testing incremental update operations	0.5 hour	
Total time		

Note that the previous estimates are conservative, and under normal circumstances the individual tasks should complete in less time. One notable exception is the installation of Netezza FDT. Depending on your DB2 Analytics Accelerator appliance hardware configuration, and on the number of hardware components in the system that need to be upgraded to the most recent firmware level, Netezza FDT installation might take between 6 and 10 hours.

Due to the number of components involved in the upgrade, and due to the number of steps necessary to complete the upgrade, it is suggested to split the upgrade process into separate blocks of tasks that can be completed individually, without affecting the mode of operation of DB2 Analytics Accelerator. Individual blocks can be identified, in the following order:

1. Install the IBM System z and client-side prerequisites, and transfer DB2 Analytics Accelerator-side components to DB2 Analytics Accelerator.
2. Install Netezza HPF and Netezza FDT, if applicable.
3. Install, upgrade, and configure the remaining components.Verify DB2 Analytics Accelerator functionality.

The first individual block is composed of the first set of tasks, grouped under Prerequisites in the previous list of upgrade components, as shown in Table 4-1 on page 53. This block excludes the Netezza HPF and Netezza FDT upgrades, which are concentrated in a separate block because they involve an outage of DB2 Analytics Accelerator.

The last block spans the remaining components (steps) in Table 4-1 on page 53, grouped under Installation and Verification. generally comprises the upgrade itself, where the previous steps include the installation of prerequisites. This step requires an outage of DB2 Analytics Accelerator as well. All of these steps are explained in more detail in the following section. When creating a PMR for an IBM Netezza upgrade, see *IBM PureData System for Analytics Upgrades Frequently Asked Questions*, which is available on the following website:

http://www.ibm.com/support/docview.wss?uid=swg21626217

4.1.4 Installing DB2 Analytics Accelerator

After determining which prerequisites are currently missing or outdated, the next step is to install the prerequisites and DB2 Analytics Accelerator[5]. It is important to install all of the prerequisites, and then DB2 Analytics Accelerator, in a specific order. If this order is not observed, there will be situations where the help of IBM Support is needed to complete the installation. Such a scenario might, for instance, happen when first installing DB2 Analytics Accelerator stored procedures, and then attempting to upgrade DB2 Analytics Accelerator.

Adhere to the following order when installing the prerequisites:

1. IBM DB2 prerequisite PTFs
2. IBM DB2 Analytics Accelerator Studio
3. Function modification identifier (FMID) HAQT410 and HCHCA21
4. DB2 Analytics Accelerator PTFs

It is suggested to keep the System Modification Program Extended (SMP/E) target libraries separate from the runtime libraries, and not to change the runtime environment to the most recent level yet. Furthermore, it should be kept in mind that HAQT410 and DB2 Analytics Accelerator PTFs take up a considerable amount of space in the program directory.

After upgrading the z/OS components of IBM DB2 Analytics Accelerator and IBM DB2 Analytics Accelerator Studio, the software components on the DB2 Analytics Accelerator side must be upgraded. All components on the DB2 Analytics Accelerator side (with the exception of Netezza HPF and Netezza FDT) can be installed using DB2 Analytics Accelerator Studio client application.

Use the transfer and apply functionality in DB2 Analytics Accelerator Studio to transfer software to DB2 Analytics Accelerator and subsequently install it. This functionality can be found in the **About** section of the DB2 Analytics Accelerator view, as illustrated in Figure 4-3 on page 53. The transfer and apply sequence shown in Table 4-2 must be followed when installing DB2 Analytics Accelerator components.

Table 4-2 Transfer and apply sequence

Step	Component name	Component name pattern
1	Incremental Update Access Server	`dwa-iu-as-<yyyymmdd-hhmm>.tar`
2	Incremental Update Replication Engine	`dwa-iu-re-<yyyymmdd-hhmm>-base.tar` `dwa-iu-re-<yyyymmdd-hhmm>-update.tar`
3	Accelerator Server	`dwa-<yyyymmdd-hhmm>.tar`
4	NPS	`dwa-nps-<yyyymmdd-hhmm>.tar`

[5] The complete IBM DB2 Analytics Accelerator Installation Guide can be found on the following website: http://publibfp.dhe.ibm.com/epubs/pdf/h1270381.pdf

When you intend to split the upgrade into separate blocks of tasks to minimize outage, all components must be transferred first, and then applied in a separate step. During the transfer phase, DB2 Analytics Accelerator is available to accelerate queries, but DB2 Analytics Accelerator is unavailable during the apply phase.

Note that it is not possible to transfer the NPS package unless DB2 Analytics Accelerator server is already at the V4.1 or later level. When synchronously running the transfer and apply operations, this limitation does not apply, because DB2 Analytics Accelerator server is upgraded before NPS.

During the transfer of the NPS package, the Netezza HPF and Netezza FDT prerequisites are checked. If they are not satisfied, an error message is displayed (Figure 4-4). Because those prerequisites should already have been validated in previous steps, this message should not appear. However, if it does, for instance because this verification was skipped, a PMR must be opened to have Netezza HPF and Netezza FDT updated by IBM Support.

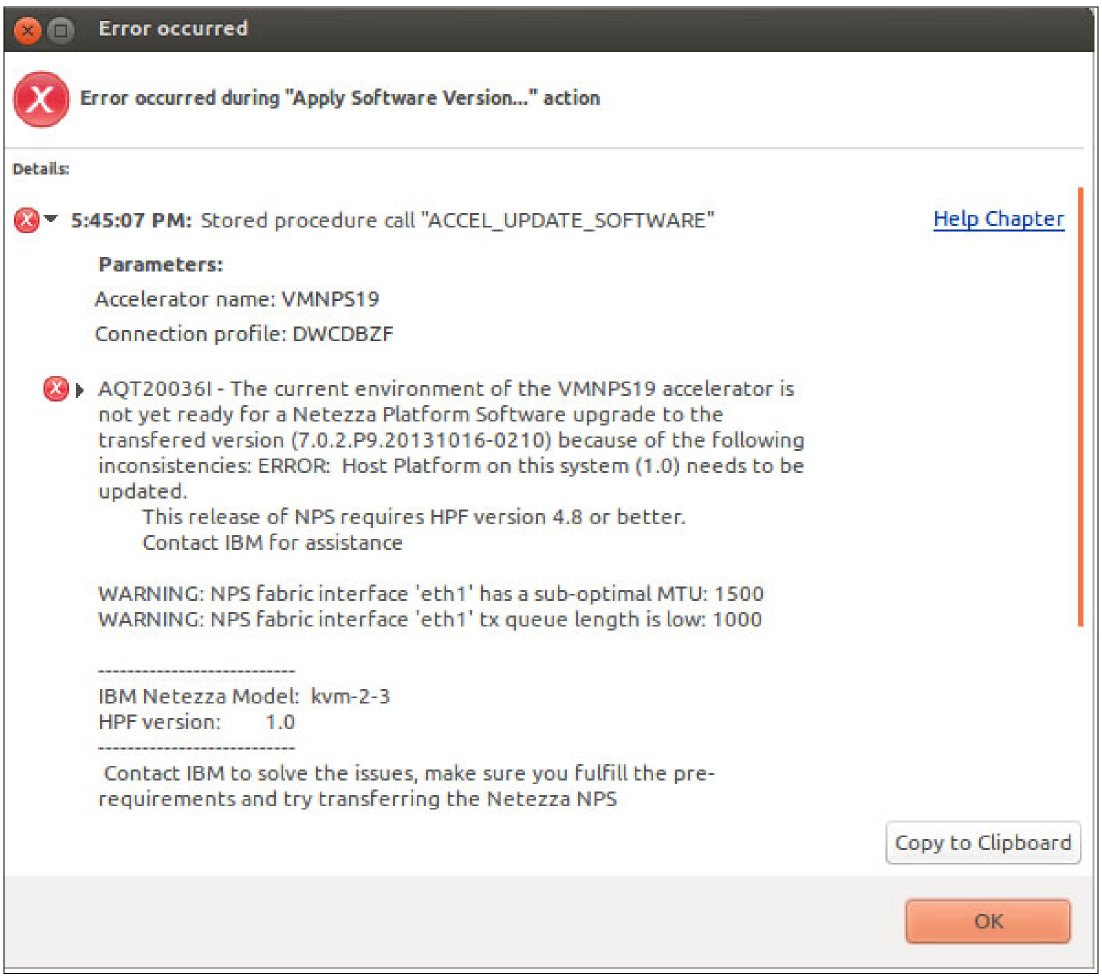


Figure 4-4 NPS pre-check

Ensure that no Configuration Console session is open before starting the apply phase of step 3, otherwise the installation fails with error message AQT10050I. More details about this problem can be found in the “Software update blocked by tasks of type Configuration Console Job” technote, available on the following website:

http://www.ibm.com/support/docview.wss?uid=swg27039231

If Netezza HPF and Netezza FDT were found to be earlier than the required level, and it was agreed with IBM Support that NPS should be upgraded to the required level in addition to Netezza HPF and Netezza FDT, step 4 of the DB2 Analytics Accelerator upgrade process can be skipped.

To complete the installation, DB2 Analytics Accelerator stored procedures must also be upgraded to version 4.1, by running the `AQTTIJSP` batch job and refreshing the IBM Workload Manager for z/OS (WLM) environment. Because DB2 Analytics Accelerator V4.1 is the first version to include IBM InfoSphere Change Data Capture for z/OS V10.2.1, the associated started task has to be upgraded, configured, and restarted.

DB2 Analytics Accelerator's incremental update feature was used in V3.1, and more than one DB2 subsystem was configured to replicate changes to DB2 Analytics Accelerator. Because of this, the number of DB2 subsystems replicating changes to DB2 Analytics Accelerator has to be temporarily decreased to one by disabling incremental update for the other DB2 subsystems, in order for the new limit of 10 DB2 subsystems replicating changes to take effect.

The reason for this is that, internally, one process is running for each DB2 subsystem replicating changes. However, in Version 4.1 the behavior changed, and only one process is now serving up to 10 DB2 subsystems. This change only takes effect after decreasing the number of processes to one. In case that only one DB2 subsystem was replicating changes before the upgrade, no data changes are lost during the migration.

After running all of these steps successfully, DB2 Analytics Accelerator has been upgraded to V4.1. It is suggested to test operations by loading a sample table and attempting to offload a query to DB2 Analytics Accelerator, and to test the incremental update feature before moving DB2 Analytics Accelerator back to production.

In case the newly activated version of DB2 Analytics Accelerator does not work as expected, it is possible to revert to the old version under certain preconditions. It is possible to complete the following actions:

- Go back and forth between PTF levels of IBM DB2 Analytics Accelerator V4.1.
- Fall back from IBM DB2 Analytics Accelerator V4.1 to V3.1.
- Open a PMR and ask IBM Support to roll back the system in all other situations.

During the significant part of the upgrade process, DB2 Analytics Accelerator is not available to accelerate queries. However, it is possible to maintain operations in a high-availability (HA) DB2 Analytics Accelerator environment by running a rolling migration. See the following section for details.

4.1.5 Rolling migration in an HA and disaster recovery environment

In an HA and disaster recovery (DR) environment, it is possible to minimize the downtime during an upgrade, and therefore the window for a planned outage, by conducting a rolling migration. The following list includes the minimum requirements for such a scenario:

- A DB2 data sharing group consisting of at least two members
- Separate WLM DB2 Analytics Accelerator stored procedure environments[6]
- Two DB2 Analytics Accelerators being connected to the data sharing group
- For the incremental update feature to be used, a dynamic virtual Internet Protocol address (VIPA) HA setup for InfoSphere Change Data Capture for z/OS, as outlined in http://www.ibm.com/support/docview.wss?uid=swg27037912

[6] Chapter 11, "Running different versions of IBM DB2 Analytics Accelerator for z/OS at the same time" in the IBM DB2 Analytics Accelerator V4.1 Installation Guide, SH12-7038, describes how to run multiple DB2 Analytics Accelerator versions in parallel in a coexisting environment.

Before the upgrade, it is necessary to synchronize both DB2 Analytics Accelerators in terms of loaded tables and configuration (tables enabled for acceleration, and tables enabled for incremental updates). Both DB2 Analytics Accelerators should be started, so that both of them are capable of serving the same acceleration requests.

The outlined process for conducting a rolling upgrade assumes that all of the components (DB2, DB2 Analytics Accelerator, InfoSphere Change Data Capture for z/OS) are redundantly available. Therefore, all of the components can be upgraded step-by-step by first upgrading the inactive components, and then the active ones. This makes it possible to operate in a mixed DB2 Analytics Accelerator V3.1 and V4.1 environment for the time of the upgrade, and lays the foundation for rolling upgrades.

Figure 4-5 illustrates the individual steps of a rolling migration, and shows the DB2 Analytics Accelerator software version for the DB2 and DB2 Analytics Accelerator side.

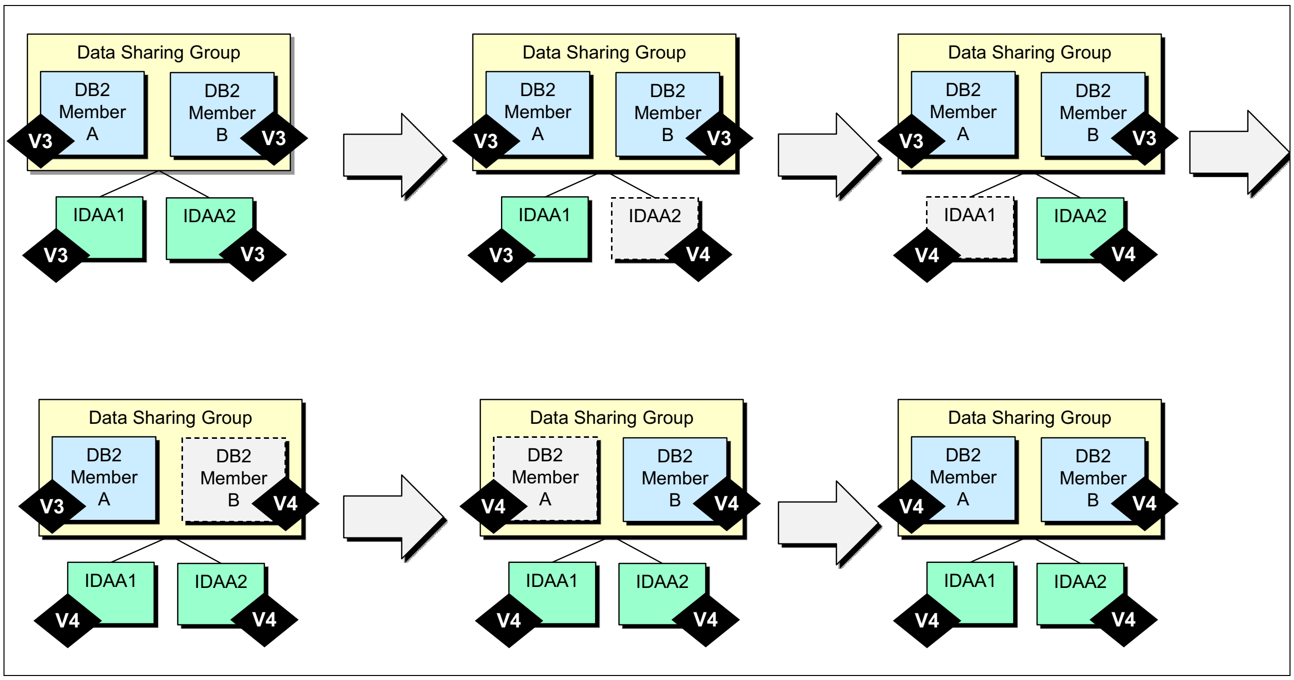


Figure 4-5 Rolling migration from DB2 Analytics Accelerator V3.1 to V4.1

To conduct a rolling upgrade, follow these steps:

1. Stop the second DB2 Analytics Accelerator (IDAA2), so that all requests are routed to the first DB2 Analytics Accelerator (IDAA1).
2. Next, IDAA2 needs to be upgraded to IBM DB2 Analytics Accelerator V4.1, as outlined in the previous sections.

   Note that for the moment DB2 Analytics Accelerator stored procedures must not yet be activated, and the InfoSphere Change Data Capture for z/OS must not yet be reconfigured and restarted. Restarting InfoSphere Change Data Capture is done as a last step after upgrading both DB2 Analytics Accelerators and the System z-side components.
3. After upgrading IDAA2, it needs to be started again.
4. The same procedure must now be repeated for IDAA1:

   a. Stop IDAA1.
   b. Upgrade IDAA1 to V4.1.
   c. Start IDAA1.

Both Accelerators are now running with IBM DB2 Accelerator V4.1.

5. Upgrade the DB2 data sharing group to V4.1:
   a. Stop the DB2 Analytics Accelerator stored procedure WLM environment on DB2 Member B.
   b. Upgrade the stored procedures using the `AQTTIJSP` batch job.
   c. Restart the WLM environments.
6. Upgrade InfoSphere Change Data Capture for z/OS:
   a. Stop the instance running on data sharing group member B using the z/OS command **`MODIFY CDCSZA1,SHUTDOWN,IMMED,HANDOVER`**. This command shuts down the instance, and automatically triggers a failover to one of the hot-standby instances running on other data sharing group members.
   b. After that, upgrade the InfoSphere Change Data Capture instance to version 10.2.1.
   c. Reconfigure InfoSphere Change Data Capture.
   d. Restart InfoSphere Change Data Capture.
7. After successfully upgrading member B, the same procedure must be applied for member A, and for all other members of the data sharing group.
8. The final step before officially moving the system back into production is to run a verification test:
   a. Load and query a sample table.
   b. Testing incremental update operations.

   With this step, the rolling migration is completed successfully.

The procedure described here can also be used to apply PTF-level upgrades.

4.2 Optional migration steps

There have been changes in DB2 Analytics Accelerator V4.1 due to the addition of functionalities. This section provides migration considerations if you have a current DB2 Analytics Accelerator V3.1 environment and want to migrate priority definitions for queries, maintenance tasks, and incremental updates. It also describes the optional step needed to use preexisting table definitions in the pseudo-catalog for static Structured Query Language (SQL) and workload balancing.

This section provides information about the following topics:

- The workload management definitions for query prioritization
- The priority setting of accelerator maintenance tasks
- New resource management for incremental update
- Migrating SYSACCELERATEDTABLES table

4.2.1 The workload management definitions for query prioritization

DB2 Analytics Accelerator V3.1 provided controls for prioritizing work on a DB2 Analytics Accelerator. These controls can be divided into two categories:

- DB2 Analytics Accelerator prioritization for a given DB2 subsystem
- DB2 Analytics Accelerator prioritization across multiple DB2 subsystems

For a given DB2 subsystem, you have the ability to prioritize accelerated distributed data facility (DDF) queries and Accelerator maintenance tasks.

System z extends some of its resource management capabilities to DB2 Analytics Accelerator by using the already known WLM *importance* that is associated with each query. In particular, the WLM importance is passed to DB2 Analytics Accelerator with the query. This importance is mapped to internal DB2 Analytics Accelerator priorities, resulting in accelerated queries that are being managed in a manner that is consistent with System z.

Table 4-3 shows the four DB2 Analytics Accelerator V3.1 priority levels that are used to differentiate work on DB2 Analytics Accelerator. Notice how work that is running at importance 3, 4, or 5 all runs at the same priority on DB2 Analytics Accelerator.

Table 4-3 Mapping of WLM importance to DB2 Analytics Accelerator V3.1 priorities

WLM importance	DB2 Analytics Accelerator priorities
System	Critical
Importance 1	Critical
Importance 2	High
Importance 3, 4, 5	Normal
Discretionary	Low

DB2 Analytics Accelerator Version 4.1 extends the already-available WLM query prioritization support for remote queries to local queries. DB2 now detects the service class and importance level for local applications, and sends the importance level to DB2 Analytics Accelerator by a special register before **PREPARE**.

Service class is detected based on an application's address space for the following types:

- Time Sharing Option (TSO) subsystem type, for example, SQL Processor Using File Input (SPUFI).
- Job Entry Subsystem (JES) subsystem type (for example, host language applications that are run using TSOBATCH).
- Started task control (STC) subsystem type for any applications opened using **START**.
- For IBM IMS and IBM CICS applications, the service class is detected based on a passed performance block.

DB2 Analytics Accelerator now maps the importance level to the Netezza priority before query execution (same priorities as for remote applications).

DB2 sends the importance level to DB2 Analytics Accelerator with each query.

The changes in prioritization after query start are not reflected.

Netezza supports only 4 different priority levels, so multiple WLM importance levels have to be mapped against the same Netezza priority.

Table 4-4 shows the mapping of importance to priority in DB2 Analytics Accelerator V4.1. Compare it with Table 4-3 on page 60.

Table 4-4 Mapping of WLM Importance to DB2 Analytics Accelerator 4.1 priorities

WLM Importance	DB2 Analytics Accelerator priorities
System	Critical
Importance 1	Critical
Importance 2	Critical
Importance 3	High
Importance 4	Normal
Importance 5	Low
Discretionary	Low

Note that there are changes compared to DB2 Analytics Accelerator V3.1, in which WLM importance level 2 mapped to Netezza priority High (in version 4.1, it maps to Critical) and levels 3, 4, and 5 mapped to Normal (in version 4.1, they map to High, Normal, and Low).

Important: Performance administrators should revisit their query WLM service class definitions, and make appropriate updates to ensure that the associated WLM classification rules are in line with business objectives.

Setting these priorities typically necessitates communication between the database administrator (DBA) and the System z performance administrator. Because WLM importance level 1 is generally not recommended for transactions or queries, probably many users were not benefitting from the highest level of accelerator priority, CRITICAL, which is now available with DB2 Analytics Accelerator V4.1.

4.2.2 The priority setting of accelerator maintenance tasks

The priority of the DB2 Analytics Accelerator side of accelerator load, and the other accelerator self-driven maintenance tasks (reorgs and runstats) are controlled from the DB2 Analytics Accelerator configuration console.

For a given attached DB2 subsystem (or data sharing group), queries and maintenance tasks run in the same DB2 Analytics Accelerator resource sharing group. If you anticipate concurrent execution of queries and loads, it is important to properly set the relative priority between them.

To make it easier for the performance administrator to set the relative DB2 Analytics Accelerator priorities between queries and load (of the same DB2 subsystem), the DB2 Analytics Accelerator V4.1 configuration console panel for specifying the priority of maintenance tasks has changed to use “Importance” terminology consistent with WLM, versus the original version 3.1 options.

Figure 4-6 shows the DB2 Analytics Accelerator V3.1 configuration console maintenance panel.

```
Queries are run with the priority that is set in the corresponding
System z Workload Manager (WLM) environment. You can set the priority
of maintenance operations, such as loading data, analogously for
the 'DZA1DDF' subsystem:
 * 1: DEFAULT
   2: SYSTEM
   3: HIGHEST
   4: HIGH
   5: NORMAL
   6: LOW
   7: LOWEST
   8: DISCRETIONARY

Enter the appropriate number to set the priority level.
To return to the main menu of the Configuration Console, just press <enter>.
 (Default 0) >
```

Figure 4-6 DB2 Analytics Accelerator V3.1 configuration console: Priority of maintenance tasks

Figure 4-7 shows the new DB2 Analytics Accelerator V4.1 configuration console maintenance panel.

```
Queries are run with the priority that is set in the corresponding
System z Workload Manager (WLM) environment. You can set the priority
of maintenance operations, such as loading data, analogously for
the 'BZA1DDF' subsystem:
   1: IMPORTANCE 1
   2: IMPORTANCE 2
   3: IMPORTANCE 3
   4: IMPORTANCE 4
   5: IMPORTANCE 5
   6: DISCRETIONARY
 * 7: DEFAULT

Enter the appropriate number to set the priority level.
```

Figure 4-7 V4.1 Accelerator configuration console: Priority of maintenance tasks

Table 4-5 on page 63 exhibits the mapping of these options to the DB2 Analytics Accelerator V3.1 internal priorities.

Table 4-5 DB2 Analytics Accelerator V3.1 maintenance mapping

Version 3.1 configuration console	WLM comparable importance	DB2 Analytics Accelerator V4.1 priority	Notes
SYSTEM	IMPORTANCE 1	CRITICAL	The version 3.1 maintenance priority options introduced new terminology not familiar to system (WLM) or database administrators.
HIGHEST	IMPORTANCE 1	CRITICAL	
HIGH	IMPORTANCE 2	HIGH	
NORMAL	IMPORTANCE 3,4,5	NORMAL	
LOW	IMPORTANCE 3,4,5	NORMAL	
LOWEST	IMPORTANCE 3,4,5	NORMAL	
DISCRETIONARY	DISCRETIONARY	LOW	
DEFAULT	DEFAULT (IMPORTANCE 3)	NORMAL	Load tasks run as NORMAL, although the other maintenance tasks run as LOW.

Table 4-6 exhibits the mapping of these options to the version 4.1 internal priorities.

Table 4-6 DB2 Analytics Accelerator V4.1 maintenance mapping

Version 4.1 configuration console	DB2 Analytics Accelerator priority	Notes
IMPORTANCE 1	CRITICAL	Administrators only use IMPORTANCE 2 to 5 to make use of DB2 Analytics Accelerator priority levels. Using IMPORTANCE terminology makes it easier to set relative priorities between queries and loads.
IMPORTANCE 2	CRITICAL	
IMPORTANCE 3	HIGH	
IMPORTANCE 4	NORMAL	
IMPORTANCE 5	LOW	
DISCRETIONARY	LOW	
DEFAULT (IMPORTANCE 4)	NORMAL	Load tasks run as NORMAL, although the other maintenance tasks run as LOW.

Important: When upgrading to version 4.1, administrators should re-examine their accelerator maintenance priority settings to ensure that they are set consistently with their business and workload objectives.

4.2.3 New resource management for incremental update

Before version 4.1, the incremental update work on DB2 Analytics Accelerator was fixed at a NORMAL priority, and was susceptible to getting minimal resources if competing with queries or other tasks running at higher priorities. In version 4.1, incremental update work is managed in its own accelerator resource sharing group, and has a fixed MIN and MAX resource allocation of 100, assuring it a portion of DB2 Analytics Accelerator resources. This is described in detail in Chapter 6, “Load and incremental update” on page 113.

4.2.4 Migrating SYSACCELERATEDTABLES table

To support static SQL and workload balancing, the naming scheme for tables had to be changed. A table in DB2 for z/OS needs to have the same name on all accelerators to use workload balancing capabilities and static SQL in DB2 Analytics Accelerator V4.1. You must migrate the names of the tables on each accelerator participating in the workload balancing setup in HA and DR configurations.

The reason is that the SYSACCEL.SYSACCELERATEDTABLES table needs a new name schema in the REMOTENAME column to recognize different instances of the same tables that are available on multiple accelerators. This is needed to support workload balancing across different accelerators, in addition to being able to **BIND** static SQL to the correct instance.

If you upgrade to IBM DB2 Analytics Accelerator V4.1, to benefit from workload balancing or static SQL, you need to remove tables that have been added to DB2 Analytics Accelerators with previous IBM DB2 Analytics Accelerator versions, and then add them again.

REMOTENAME is the mapped name of the table in DB2 Analytics Accelerator. In version 3.1, the mapped name was different for the table on the Netezza side for all Accelerators.

To help with the redefinition of the tables, a new stored procedure has been made available to run the migration. Version 4 PTF-2 introduces the **SYSPROC.ACCEL_MIGRATE** stored procedure, which converts existing table names in accordance with the new naming scheme. Running the stored procedure is less effort than removing, redefining, reloading, and re-enabling the tables, so consider upgrading DB2 Analytics Accelerator to version 4 PTF-2 or later to use the stored procedure.

While running the stored procedure, the target accelerator is in maintenance mode, which means that you cannot run queries or complete administrative tasks during this time.

If the same tables exist on multiple accelerators, to use workload balancing capabilities, you must migrate the tables on each accelerator participating in the workload balancing setup.

The stored procedure always migrates all of the tables on an accelerator. The stored procedure might be unable to migrate tables that were defined with DB2 Analytics Accelerator V2.1. For these tables, you need to remove the tables, and then redefine them by using the **SYSPROC.ACCEL_ADD_TABLES** stored procedure (or the corresponding function in DB2 Analytics Accelerator Studio).

The stored procedure reports a warning for each table that fails the migration. If an error has occurred, some of the tables might be left unchanged. In such a case, you must run the stored procedure again until it runs to completion, or manually remove and re-define the tables.

Tables to be migrated to the new naming scheme must be in the IntialLoadPending or Loaded state. Disable incremental updates before you start the following stored procedure:

```
CALL PROCEDURE SYSPROC.ACCEL_MIGRATE (<accelerator_name>, <message>);
```

In this procedure, the following values exist:

accelerator_name The unique name of DB2 Analytics Accelerator. This accelerator must have been defined by the **SYSPROC.ACCEL_ADD_ACCELERATOR** stored procedure.

message The message parameter is used by all DB2 Analytics Accelerator for z/OS stored procedures. It serves as an input parameter and as an output parameter.

The stored procedure renames the REMOTENAME (mapped table names) in SYSACCEL.SYSACCELERATEDTABLES that have DB2 Analytics Accelerator-dependent values and in the corresponding accelerators, and sets the value in the SUPPORTLEVEL column to 3.

Example 4-1 shows the contents of relevant columns before and after calling the **SYSPROC.ACCEL_MIGRATE** stored procedure for Accelerator VMNPS509, and OTHERACC with tables defined with DB2 Analytics Accelerator V3.1.

Example 4-1 Calling ACCEL_MIGRATE to change table names on two Accelerators

```
NAME               CREATOR ACCELERATORNAME REMOTENAME                      REMOTECREATOR SUPPORTLEVEL
 ------------------ ------- --------------- ------------------------------- ------------- ------------
 EFODIA_REQ_WRITTEN BCKE    VMNPS509        EFODIA_REQ_WRITTEN-UID_01300009 BCKE          4
 MCS_SWITCH_HOLD    BCKE    VMNPS509        MCS_SWITCH_HOLD-UID_-ID_17      BCKE          2
 MCS_SWITCH_HOLD    BCKE    OTHERACC        MCS_SWITCH_HOLD-UID_-ID_80      BCKE          2
 MCS_SWITCH_HOLD1   BCKE    VMNPS509        MCS_SWITCH_HOLD1-UID_01300025   BCKE          4

CALL SYSPROC.ACCEL_MIGRATE('VMNPS509', NULL, NULL) -- the XML parameter #2, not used, must be NULL!

NAME               CREATOR ACCELERATORNAME REMOTENAME                      REMOTECREATOR SUPPORTLEVEL
 ------------------ ------- --------------- ------------------------------- ------------- ------------
 EFODIA_REQ_WRITTEN BCKE    VMNPS509        EFODIA_REQ_WRITTEN-UID_01300009 BCKE          4
 MCS_SWITCH_HOLD    BCKE    VMNPS509        MCS_SWITCH_HOLD-UID_0130000D    BCKE          3
 MCS_SWITCH_HOLD    BCKE    OTHERACC        MCS_SWITCH_HOLD-UID-ID_80       BCKE          2
 MCS_SWITCH_HOLD1   BCKE    VMNPS509        MCS_SWITCH_HOLD1-UID_01300025   BCKE          4

CALL SYSPROC.ACCEL_MIGRATE('OTHERACC', NULL, NULL)

NAME               CREATOR ACCELERATORNAME REMOTENAME                      REMOTECREATOR SUPPORTLEVEL
 ------------------ ------- --------------- ------------------------------- ------------- ------------
 EFODIA_REQ_WRITTEN BCKE    VMNPS509        EFODIA_REQ_WRITTEN-UID_01300009 BCKE          4
 MCS_SWITCH_HOLD    BCKE    VMNPS509        MCS_SWITCH_HOLD-UID_0130000D    BCKE          3
 MCS_SWITCH_HOLD    BCKE    OTHERACC        MCS_SWITCH_HOLD-UID_0130000D    BCKE          3
 MCS_SWITCH_HOLD1   BCKE    VMNPS509        MCS_SWITCH_HOLD1-UID_01300025   BCKE          4
```

To find the tables that do not yet use the new naming scheme, submit the following query:

```
SELECT ACCELERATORNAME, CREATOR, NAME, SUPPORTLEVEL
FROM SYSACCEL.SYSACCELERATEDTABLES
WHERE SUPPORTLEVEL < 3;
```

4.3 Handling error situations

In problem situations, it is important to act in a timely manner, and to handle the occurring errors correctly and effectively to prevent outages. There are three general problem categories:

- Problems that can be fixed by consulting DB2 Analytics Accelerator resources
- Problems that can be detected by using monitoring software available on System z, but that might require interaction with IBM Support to be fixed
- Problems that require the intervention of IBM Support to be fixed

Problems detected using system automation, and problems that require IBM Support to be resolved, need to be handled using a PMR. They might also necessitate a remote support session. Both of these topics are discussed in the following sections.

There are error situations, however, that can be solved by consulting technical resources and documentation about DB2 Analytics Accelerator. Those resources might help you better understand the encountered problem, and the actions required to solve it. The resources in Table 4-7 are helpful in such situations.

Table 4-7 Resources and their web addresses

Resource name	Resource address
General	
Overview	http://www-947.ibm.com/support/entry/portal/product/information_management/db2_analytics_accelerator_for_z/os?productContext=1003251075
Guides and manuals	http://www-01.ibm.com/support/docview.wss?uid=swg27040146
IBM Knowledge Center	http://www-01.ibm.com/support/knowledgecenter/search/IBM%20DB2%20Analytics%20Accelerator?lang=en
Technotes/techdocs	
List all available DB2 Analytics Accelerator technotes	http://ibm.co/1lskXSn
IBM DB2 Analytics Accelerator for z/OS V4.1.0: Release Notes	http://www-01.ibm.com/support/docview.wss?uid=swg27040452
Prerequisites and Maintenance for IBM DB2 Analytics Accelerator for z/OS Version 4.1	http://www-01.ibm.com/support/docview.wss?uid=swg27039487
Software update blocked by tasks of type “Configuration Console Job”	http://www-01.ibm.com/support/docview.wss?uid=swg27039231
High Availability Setup for the Incremental Update Capture Engine	http://www-01.ibm.com/support/docview.wss?uid=swg27037912
Structure of DSNX881I Messages	http://www-01.ibm.com/support/docview.wss?uid=swg27037905
Synchronizing data in IBM DB2 Analytics Accelerator for z/OS	http://www-01.ibm.com/support/docview.wss?uid=swg27038501
How to store the query history in a DB2 for z/OS table	http://www-01.ibm.com/support/docview.wss?uid=swg27039739
Replacing a disk	http://www-01.ibm.com/support/docview.wss?uid=swg27040456
Initiating a Snippet Processing Unit (SPU) Failover	http://www-01.ibm.com/support/docview.wss?uid=swg27040455
How IBM DB2 Analytics Accelerator for z/OS handles correlated sub-queries	http://www-01.ibm.com/support/docview.wss?uid=swg27037928
Network requirements for System z	http://www-01.ibm.com/support/docview.wss?uid=swg27024236
Network connections for IBM DB2 Analytics Accelerator	http://www-01.ibm.com/support/docview.wss?uid=swg27023654
Network Configurations for IBM DB2 Analytics Accelerator for z/OS	http://www-01.ibm.com/support/docview.wss?uid=swg27028171
IBM Redbooks publication	
Optimizing DB2 Queries with IBM DB2 Analytics Accelerator for z/OS	http://www.redbooks.ibm.com/abstracts/sg248005.html
Hybrid Analytics Solution using IBM DB2 Analytics Accelerator for z/OS V3.1	http://www.redbooks.ibm.com/abstracts/sg248151.html
IBM DB2 Analytics Accelerator: High Availability and Disaster Recovery	http://www.redbooks.ibm.com/Redbooks.nsf/RedpieceAbstracts/redp5104.html

4.3.1 Error messages

Error messages might occur during maintenance operations using the IBM Accelerator Studio, during batch jobs when calling Accelerator stored procedures, or during query processing. Generally, query errors are reported as regular SQLCODEs, and SQLSTATEs to the client. Errors resulting from maintenance operations and from batch jobs are reported as `AQT<x>yyyyI` messages. Advanced Query Tool (AQT) messages can be classified depending on the number denoted by <*x*>:

0 Message originates from DB2 Analytics Accelerator Studio.
1 Message originates from DB2 Analytics Accelerator stored procedures.
2 Message originates from DB2 Analytics Accelerator Server.

A list of AQT error messages, including a message explanation, problem determination, and possibly a user action section, can be found in the *IBM DB2 Analytics Accelerator Version 4.1 Stored Procedures Reference*, SH12-7039:

http://publibfp.dhe.ibm.com/epubs/pdf/h1270391.pdf

The messages can also be found online on the following website:

http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/topic/com.ibm.datatools.aqt.doc/sp_msg/messages/idaa_msgs.html

Figure 4-8 on page 68 shows message AQT10050I, which was generated during an unsuccessful call to **ACCEL_LOAD_TABLES**. As stated by the error message, the Netezza back end is currently not online. Therefore, the **ACCEL_LOAD_TABLES** call failed, because it is not possible to store data on DB2 Analytics Accelerator when the back end is offline.

In many cases the error message details and the associated explanation and user action help you determine what the problem is, and then resolve it. If it is not possible to fix the error using the available documentation, it is necessary to open a PMR to resolve the situation.

Figure 4-8 shows message AQT10050I.

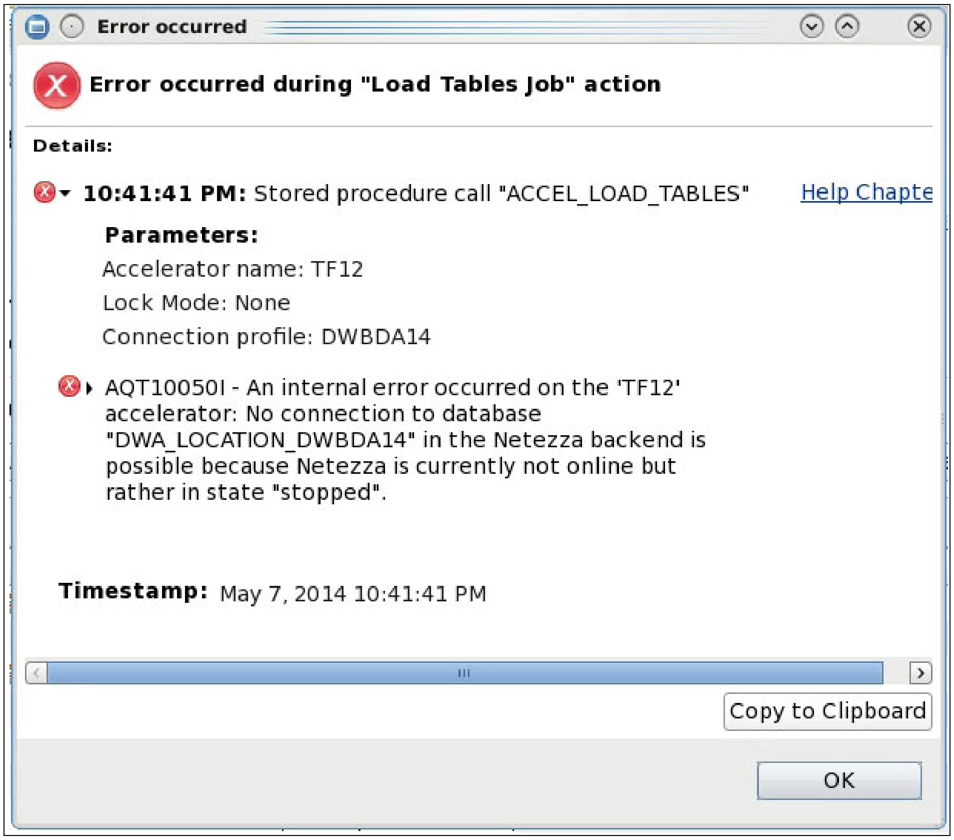


Figure 4-8 AQT10050I error message

4.3.2 DSNX881I alert messages

Hardware and software alerts are reported using default subsystem name (DSN), specifically DSNX881I, messages[7]. These messages are generated for certain hardware and software problems originating from DB2 Analytics Accelerator, and are transferred to the z/OS system log (SYSLOG) of every DB2 subsystem paired to DB2 Analytics Accelerator. The precondition for receiving DSNX881I messages is that DB2 Analytics Accelerator has to be started. For started accelerators, this information is transferred every 20 seconds.

Examples of events triggering DSNX881I messages include broken hard disks, host state changes, and incremental update-related alerts. DSNX881I messages can either be informational messages, warnings, or errors. Depending on the severity and the type of error, different actions might apply. The *Structure of DSNX881I Messages* techdoc lists the general structure of the messages, in addition to a complete list of possible messages, including their severity and suggested actions:

http://www-01.ibm.com/support/docview.wss?uid=swg27037905

Use system automation tools to scan the system log for DSNX881I messages at regular intervals, and notify support personnel about important messages so that appropriate actions can be taken.

[7] See the following website for more information about DSNX881I messages:
http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/index.jsp?topic=%2Fcom.ibm.db2z10.doc.msgs%2Fsrc%2Ftpc%2Fdsnx881i.htm

4.3.3 Opening a problem management record

There are problems that cannot be solved by consulting the available DB2 Analytics Accelerator documentation, and cannot be prevented from becoming severe by preventive actions, such as monitoring DSNX881I messages. For those problems, it is necessary to open a PMR, so that IBM Support can investigate the problem and deliver an appropriate solution. The following list includes possible reasons for opening a PMR:

- Error messages that were not solvable using the available resources
- DSXN881I-related error messages for which the suggested action is to contact IBM Support
- Hardware and software problems causing outages

A PMR can be opened using the IBM Service requests home page (http://www.ibm.com/support/servicerequest/). Make sure to select **IBM DB2 Analytics Accelerator** (Component ID 5697AQT00) as the product, and not **PureData System for Analytics N1001/N2001**.

All incoming PMRs for DB2 Analytics Accelerator are screened and dispatched internally to the appropriate technical support group, to make sure that problems are solved by the correct technical support specialists. Due to the integrative nature of DB2 Analytics Accelerator, PMRs are internally routed to the group best suited to solve the problem, and are re-routed if the problem involves more than one of the product components.

To expedite the PMR processing, it is suggested to include the following information as a preferred practice:

- Problem description, including details about the nature of the problem, such as error messages and the effect that it is having
- Details of steps that were taken to lead to the current state
- Screen captures that can help the investigation
- IBM DB2 Analytics Accelerator Trace
- Relevant problem-specific diagnostic information, including but not limited to the following items:
  - DB2 SYSLOG
  - InfoSphere CDC JES log
  - Dumps or core dumps

Some of the diagnostic information might be time-sensitive, therefore, it is suggested to collect all information at the time of failure.

During the course of the problem management record, IBM support determines the root cause of the problem and delivers an appropriate solution, such as a hot fix or corrective actions on DB2 Analytics Accelerator.

For select problems, additional information might be required, necessitating remote access to the IBM DB2 Analytics Accelerator environment for remote diagnosis. In such instances, IBM Support collects information from, or fixes the problem directly on, the impeded system. The detailed procedure is explained in the following sections.

4.3.4 Collecting a trace archive

The IBM DB2 Analytics Accelerator trace is a single archive file containing support information, such as log and trace files, facilitating the problem determination and analysis for IBM Support. The content of the file is selectable, offering flexibility in the amount of data sent to IBM support and the nature of data to be sent (for example, the inclusion of potentially sensitive data, such as core files).

DB2 Analytics Accelerator trace archive content can be divided into two categories:

- Data being collected unconditionally
- Data being collected depending on the checked boxes in the DB2 Analytics Accelerator Save Trace dialog

The table in Figure 4-9 contains a high-level overview of DB2 Analytics Accelerator information collected during a save trace operation.

Unconditional data	Conditional data

• *Accelerator Server*
  ◦ Version information
  ◦ Process information
  ◦ Configuration information
• *Netezza*
  ◦ Version information
  ◦ Process information
  ◦ Configuration information
  ◦ Hardware information
  ◦ System health information
• *Incremental Update*
  ◦ Version information
  ◦ Configuration information
	• *Catalog dumps*
	  ◦ Database and table schema dumps
	• *Traces*
	  ◦ Accelerator server trace files
	  ◦ Netezza log files
	  ◦ Incremental update log and trace files
	• *Plans*
	  ◦ Query execution specific information
	    ▪ Query text
	    ▪ Diagnostic information
	    ▪ Execution plans
	• *Core dumps*
	  ◦ Accelerator core dumps
	  ◦ Load log files (may include table data)
	  ◦ Incremental update load log files (may include table data)
	• *Netezza core dumps*
	  ◦ Host core dumps
	  ◦ SPU core dumps
	• *Log history*
	  ◦ Historical log and trace data
	  ◦ Historical query execution plans
	• *Detailed Netezza diagnostics*
	  ◦ Additional Netezza log and trace data
	• *Stored Procedure Trace Information*
	  ◦ Only collected if stored procedure traces are activated
	• *Eclipse error log*
	  ◦ Accelerator Studio related log files

Figure 4-9 High-level overview of conditional and unconditional trace archive content

The most convenient way to collect a trace is using DB2 Analytics Accelerator Studio. Trace-related operations can be found in the upper right corner of the DB2 Analytics Accelerator view shown in Figure 4-10 on page 71. They provide for configuring, saving, and clearing DB2 Analytics Accelerator traces.

By default, the DB2 Analytics Accelerator server trace level is set to `DEFAULT`, and the DB2 Analytics Accelerator stored procedure trace level is set to `OFF`. It is advised not to change the trace level defaults unless explicitly prompted to do so.

Using a more fine-grained trace level might cause performance degradations due to the higher frequency of generated trace messages, and the added resource usage to generate those messages and to derive diagnostic information. This leads to shorter periods of time being captured by the trace files, because a maximum file size limit is imposed on trace files to prevent filling up disk space indefinitely.

As soon as this limit is reached, the oldest trace messages are discarded to make room for new messages. The option to clear the trace should be used with caution, and it is suggested to run it only if instructed to do so by IBM Support.

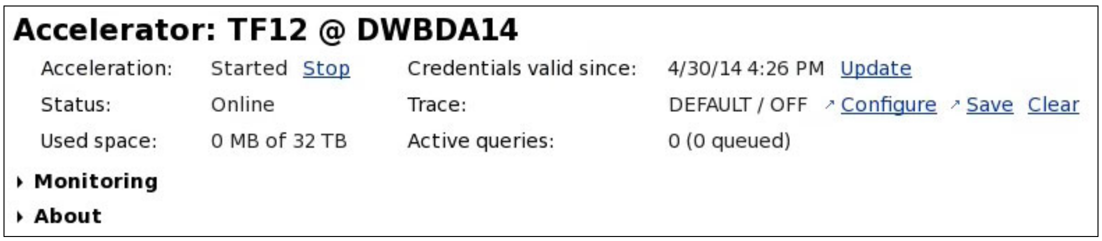


Figure 4-10 DB2 Analytics Accelerator trace operations

Because all trace files are deleted during the clear trace operation, all historical data is lost, making it impossible to diagnose and solve problems that occurred before the clear trace operation. DB2 Analytics Accelerator manages trace files on its own, by wrapping the trace files when reaching the maximum trace file size limit. Similar limits apply for other diagnostic files being generated by DB2 Analytics Accelerator server and other DB2 Analytics Accelerator components.

You can create a trace archive that can then be attached to a PMR by using the **Save** menu option in the upper right corner of DB2 Analytics Accelerator Studio. A new dialog then opens, offering choices about the contents of the trace file.

For most problems, it is sufficient to accept the default selection, as shown in Figure 4-11.

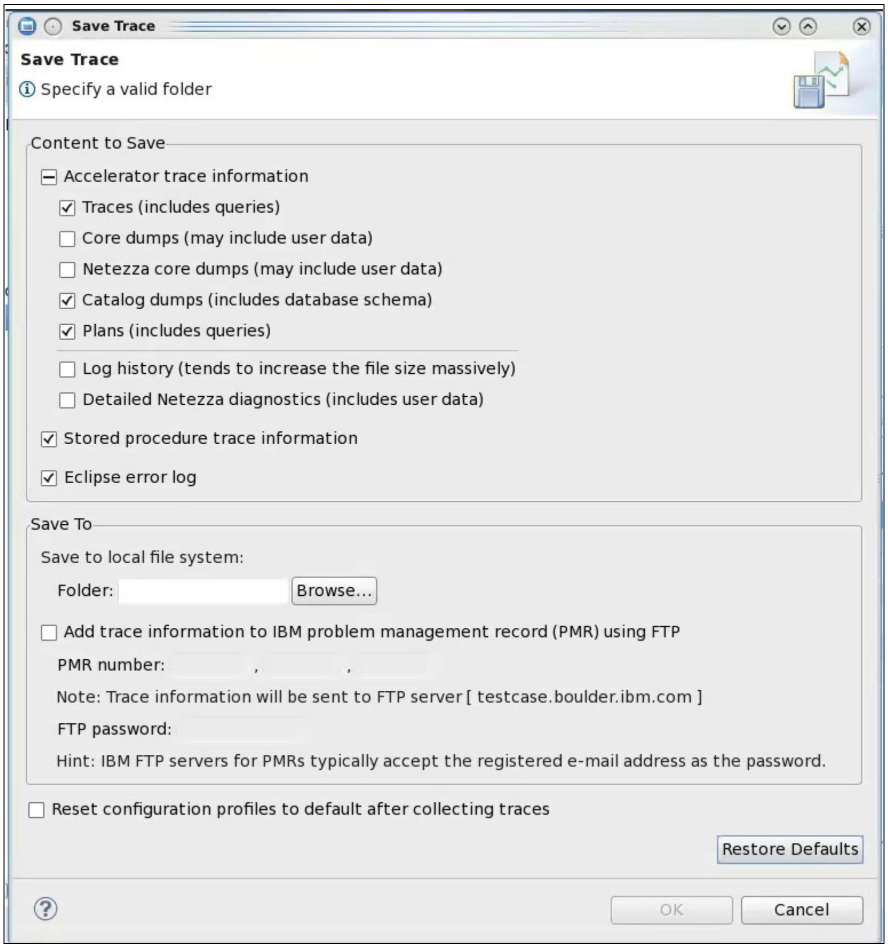


Figure 4-11 Save Trace dialog

Only if the problem is related to a specific feature that can be linked to one of the available choices, or if prompted by IBM Support, should additional options be checked. Depending on the selected options and the appliance model, the time to collect a trace archive can vary from one minute to several minutes[8]. The obtained trace archive should then be attached to the PMR. This saves time during PMR processing, because DB2 Analytics Accelerator trace archive is the central source of problem determination in about 80% of the cases.

4.3.5 Remote support

There are problems where the information contained in the trace archive and diagnostic information is not sufficient to solve a problem, or where problem solving requires remote access to DB2 Analytics Accelerator. In those situations, remote support sessions help to close the gap and pose an effective and time-efficient alternative to onsite support. These sessions are carried out using the Cisco WebEx[9] platform.

[8] In rare cases trace collection times out with an error message. In such situations, it is necessary to contact IBM Support.

[9] http://www.webex.com/

WebEx enables you to share the screen content with IBM Support, and provides for remote support by virtually sharing your mouse and keyboard. Control over the remote session can be regained at any time during the web meeting by clicking with the mouse, because local input always has precedence over remote interaction.

Before an upcoming WebEx session, the agenda of the remote support session is discussed, and a date and time are scheduled (either using the PMR or a telephone conference). Next, IBM support invites you to the WebEx meeting by email. The email notification (Figure 4-12) contains a hyperlink to join the meeting, in addition to a calendar entry.

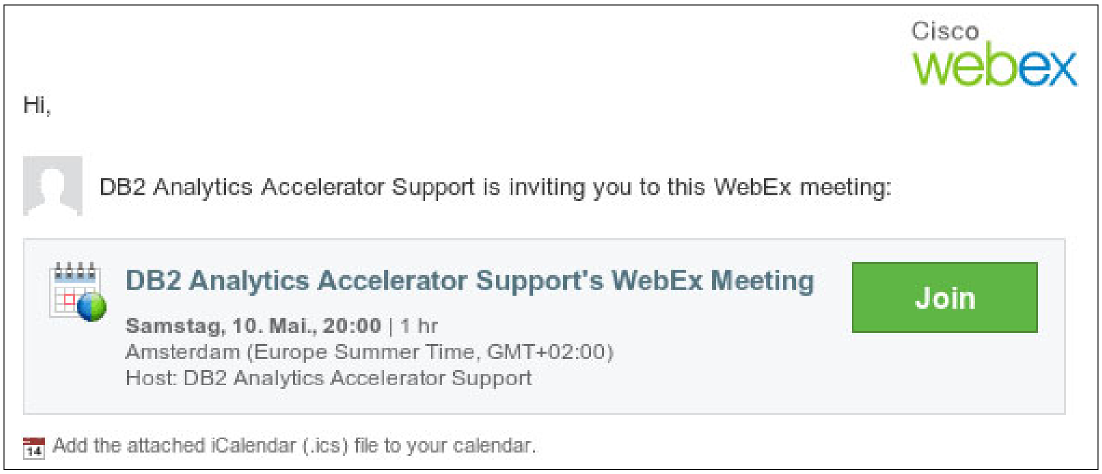


Figure 4-12 WebEx notification email

Following the link opens the meeting application, as shown in Figure 4-13 on page 74. From there, it is possible to initiate the screen sharing session by clicking **Share My Desktop**.

In the lower right corner of the application, there is a chat application that can be used to exchange messages with the support personnel. For instance, you can provide the DB2 Analytics Accelerator Service Password[10] at the beginning of the meeting, if it was agreed that the client will open a session to DB2 Analytics Accelerator and enter the service password in preparation for the meeting.

IBM Support will take over control of the session from the point where a connection to DB2 Analytics Accelerator is established. During the session, the service technician will explain what is being done to the system, so that all actions become apparent. The remote session will be closed in mutual agreement that the set goals have been achieved, or on which follow-up actions are required to conclude the PMR.

[10] A temporary, appliance serial number-specific password, allowing administrative access to the system, only valid for the day it was generated.

Figure 4-13 shows the open WebEx meeting.

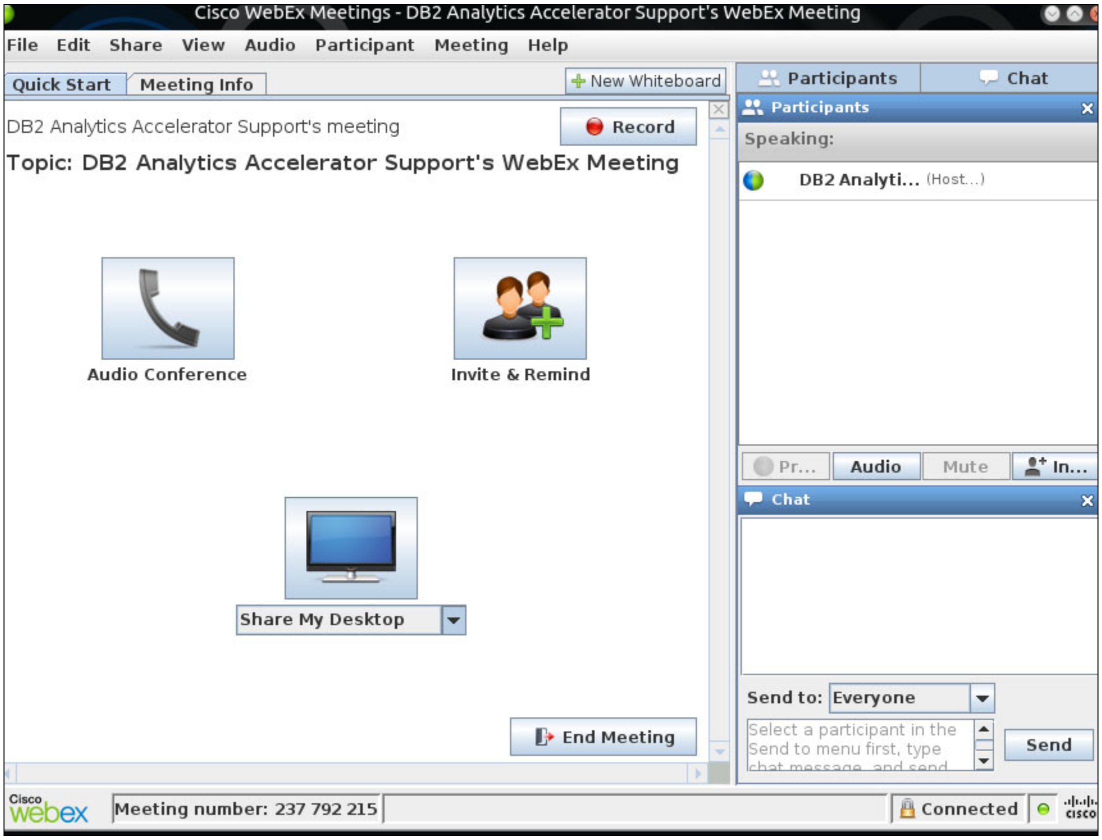


Figure 4-13 WebEx meeting application

As a final step, IBM Support will update the PMR to document what has been done during the web meeting. IBM Support also either lists the success of the remote session, or lists the action items that have been defined during the meeting.

4.4 Optimizing DB2 Analytics Accelerator load strategy

DB2 Analytics Accelerator offers three different data refresh options:

- Full-table load and reload
- Partition-based reload
- Incremental update

The first two options are snapshot-based reload options, but the third option constantly replicates the latest changes from DB2 to DB2 Analytics Accelerator. Depending on the following factors, latency goals for tables can differ, and so do the suggested refresh options per table:

- The amount of data changed per table
- The number of changed partitions
- The requirements of the different business units using DB2 Analytics Accelerator technology in their day-to-day business

Before implementing a data refresh strategy, it is advisable to iterate over the list of tables currently loaded on DB2 Analytics Accelerator, and tables that you plan to load, and cluster them into different latency groups based on an assessment of real business needs. One of those groups, for instance, might comprise all tables that should be five minutes old at most, another group might contain tables with a maximum age of one hour, and so on.

Because DB2 Analytics Accelerator is optimized for analytics workloads, it is at a disadvantage when dealing with online transaction processing (OLTP)-type workloads, such as trickle feed updates initiated by the incremental update feature. Keeping that in mind, it is suggested to favor mass load operations over incremental update, and only use incremental update for a subset of tables when business use cases require near real-time data currency.

Using the newly created list containing groups of tables belonging to the same data currency cluster, a load strategy can be implemented. The implementation of such a strategy is described in detail in the chapter about data latency management in *Hybrid Analytics Solution using IBM DB2 Analytics Accelerator for z/OS V3.1*, SG24-8151, and in the article "Synchronizing data in IBM DB2 Analytics Accelerator for z/OS" available on the following website:

http://www-01.ibm.com/support/docview.wss?uid=swg27038501

These sources include information about how to parallelize load table stored procedure calls to improve throughput for segmented and partitioned tables.

Managing data currency on DB2 Analytics Accelerator using DB2 Analytics Accelerator Studio is not possible for large amounts of tables, especially for refresh intervals of less than a week. Therefore, it is suggested to use automation for this task.

In its most basic form, an automation solution might only trigger regular batch jobs calling the sample C job provided in **SAQTSAMP(AQTSCALL)** to run reloads according to the defined latency goals for *all* tables. A more sophisticated data refresh solution would dynamically determine when a table needs to be refreshed using the available DB2 real-time statistics (RTS) information.

Using such a dynamic approach only refreshes the tables and partitions that have changed since the last refresh interval, avoiding unnecessary reload and therefore saving time and capacity on System z. Tables which, due to their latency goals, should be refreshed using the incremental update feature, have to be defined once as replicated tables in DB2 Analytics Accelerator Studio.

They can also be defined using the **ACCEL_SET_TABLES_REPLICATION** stored procedure, which enables or disables incremental updates for one or more tables on an accelerator. No regular maintenance operations are necessary in that case, as opposed to reload-based strategies in which reloads have to be scheduled regularly.

The latency decisions should be revisited at regular intervals to make sure that they are still up-to-date, and that the system is capable of achieving the set goals. Especially after initially setting up an automated load strategy, it is necessary to verify that the system is capable of achieving the set goal. If this is not the case, either latency goals have to be adjusted, or a more potent appliance configuration has to be used.

When adjusting latency goals, there are two possibilities:

- The first possibility is to adjust the incremental update latency goal using the DB2 Analytics Accelerator Configuration Console, as shown in Example 4-2.

Example 4-2 Setting incremental update latency using the Configuration Console

```
 **********************************************************************
 * Welcome to the IBM DB2 Analytics Accelerator Configuration Console
 **********************************************************************

 You have the following options:
 (1) - Change the Configuration Console Password
 (2) - (Menu) Run Netezza Commands
 (3) - (Menu) Run Accelerator Functions
 (4) - (Menu) Manage Hardware
 (5) - (Menu) Manage Incremental Updates
 ----------------------------------------------------------------------
 (x) - Exit the Configuration Console
5

 main -> 5
 ----------------------------------------------------------------------
 You have the following options:

 (0) - Go back one level
 (1) - Enable incremental updates
 (2) - Disable incremental updates
 (3) - Update DB2 subsystem credentials
 (4) - Change the latency goal for incremental updates

 (Default 0) > 4


This will change the current latency goal for replication
A higher value will reduce the frequency of changes being applied on the target
side causing a better throughput. This should be used when lots of changes need
to be applied through replication.
A lower value means that changes are applied more frequently consuming more
resources on the target side.
The goal may not necessarily be achieved and lower values can cause bad
performance for to many changes.
Do you want to continue? (y/n): y

Existing replication instances:
  Netezza

Current latency goal is:  1 min. ( min. in config file)

Please specify the new latency goal in minutes (0 to abort):
```

- The second possibility is to decide to refresh a table using either the full-table reload or partition-based reload method, rather than incrementally updating it.

To minimize the batch window necessary to refresh DB2 Analytics Accelerator table data, and to process all refresh requests in the batch window, load table stored procedure calls should be parallelized. This process is described in the chapter about data latency management in *Hybrid Analytics Solution using IBM DB2 Analytics Accelerator for z/OS V3.1*, SG24-8151. Moreover, RTS information should be used to reload only tables that changed during the last interval to further decrease the batch window size.

DB2 Analytics Accelerator introduced support for Netezza (now IBM PureData for Analytics) N2001 and N2002 appliance models, and support for overriding **SAQTSAMP(AQTENV)** environment settings on a per-stored-procedure-call level. These features bring new possibilities to enhance the existing DB2 Analytics Accelerator load automation strategy.

With the PureData for Analytics N2001 host model refresh comes an upgrade to the processor and main memory on the Netezza hosts, making more resources available and enabling a higher degree of parallelism when loading tables concurrently. During experiments, it was empirically proven that a PureData for Analytics N2001 system provides for a higher degree of load parallelism compared to PureData for Analytics N1001 systems.

Figure 4-14 shows load throughput results for a varying number of concurrent streams. Although the best throughput rate lay at 8 parallel load streams for PureData for Analytics N1001 models, this number increased to 20 for PureData for Analytics N2001 models[11]. This either provides for a higher number of parallel load streams as compared to N1001, or to a similar throughout as before with concurrent incremental update and query workload.

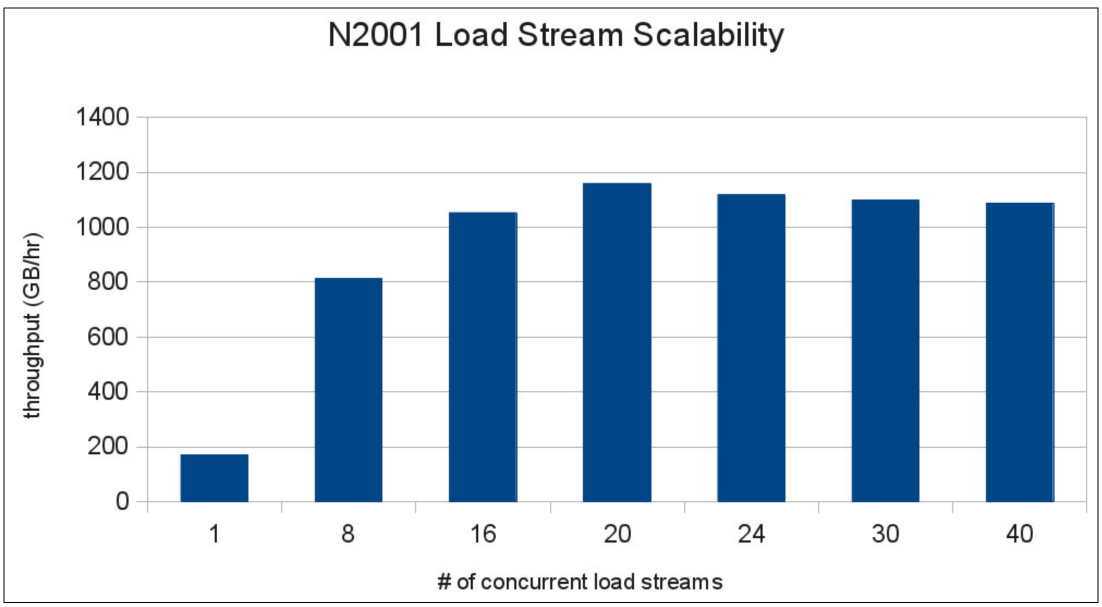


Figure 4-14 N2001 Load stream scalability

Starting with DB2 Analytics Accelerator V4.1 PTF2, it is possible to set environment variables in the **MESSAGE** parameter of the stored procedure call. This feature can be used to override the **SAQTSAMP(AQTENV)** defaults on a per-table basis, to prioritize some tables over others by expediting the load process using a higher-than-normal degree of parallelism.

[11] Both, N1001 and N2001 were conducted using full-rack appliance configurations (N1001-010, N2001-010)

It is also possible to use this feature to ensure constant resource use on the System z and Accelerator side throughout the complete load process for the table. This is accomplished by making sure that the number of partitions of a table is divisible without remainder by the value of AQT_MAX_UNLOAD_IN_PARALLEL. For example, loading a table with 10 partitions using the default AQT_MAX_UNLOAD_IN_PARALLEL value of 4 would require loading 2x4 partitions in parallel and 1x2 partitions. Using a value of 5 would only require loading 2x5 partitions in parallel.

To do so, the Extensible Markup Language (XML) code listed in Example 4-3 has to be passed to the **ACCEL_LOAD_TABLES** stored procedure as **MESSAGE**.

Example 4-3 Setting environment variables

```
<?xml version="1.0" encoding="UTF-8" ?>
<aqttables:messageControl xmlns:aqttables="http://www.ibm.com/xmlns/prod/dwa/2011"
version="1.2">
<environment>
   <environmentVariable name="AQT_MAX_UNLOAD_IN_PARALLEL" value="8" />
</environment>
</aqttables:messageControl>
```

These two new features offer an extended flexibility over the previous DB2 Analytics Accelerator behavior, enhancing the potential and possibilities when devising and optimizing a DB2 Analytics Accelerator load strategy.

4.5 Tuning distribution and organizing keys

Chapter 2, “Database design considerations” on page 19, described how setting distribution and organizing keys can, in some instances, improve performance for queries. Although DB2 Analytics Accelerator delivers unparalleled query performance immediately upon implementation, business requirements might necessitate the definition of distribution and organizing keys. Defining these keys can improve performance for a set of critical queries, or decrease the latency for incremental update operations.

It is important to remember that setting both distribution *and* organizing keys is a tuning measure, and although it can drastically improve performance for a set of queries, there might be other queries that suffer a performance degradation due to the selected keys.

A prominent example of such opposing goals is defining distribution and organizing keys *to tune query performance,* versus defining distribution and organizing keys *to reduce incremental update latency*. There might be situations where *different key combinations* have to be chosen for query performance and for incremental update performance.

4.5.1 Choosing distribution keys

A distribution key defines how table data is distributed between the worker nodes. To distribute the work across all worker nodes evenly, it is important to avoid a situation in which one worker node governs a larger amount of table data than another node. This goal can be achieved by choosing between two available distribution schemes:

- Random distribution, where records of a table are distributed among worker nodes in a round-robin fashion
- Hash distribution, where records of a table are distributed among worker nodes based on a hash function that takes into account up to four key columns, so that records with the same hash values are distributed to the same worker node

The key to high query performance is a uniform distribution of data across worker nodes, and optimal access to data. Defining subpar distribution keys might result in data skew[12], which is *a non-uniform distribution of data across worker nodes*. An example of such a suboptimal distribution key would be selecting a Boolean column (True/False, Input/Output), because data would only be distributed between two worker nodes.

The nodes managing more data than others have to work harder and longer, and they need more resources and time to complete their jobs. They become the performance bottleneck in query processing. An optimal table distribution schema has no skew. By definition, randomly distributed tables do not have data skew, because each worker node receives the same fraction of data.

Several factors support achieving the goal of uniform data distribution and optimal data access. The following list of preferred practices for choosing an optimal distribution key are a compilation of the best practices listed in the Netezza System Administration Guide and the DB2 Analytics Accelerator Distribution keys documentation in IBM Knowledge Center[13]:

- The more distinct the distribution key values, the better they distribute across nodes.
- Rows with the same distribution key value are distributed to the same worker node.
- Parallel processing is more efficient when table rows are evenly distributed across nodes.
- Tables used together should use the same columns for their distribution key.
- Keys largely used in `equi join` clauses make good candidates for the distribution key.
- Keys consisting of more than one column restrict the generality of the selection.
- Columns referenced by frequently run queries should be preferred when selecting distribution keys.

The Netezza database system is capable of performing join optimizations if the join columns are used as the distribution key for all tables being joined. In this case, it is possible to run a collocated join, because all rows to be matched during the join operation are located on the same worker node.

Selecting different distribution keys for two tables used in a join operation, alternatively, results in table data being temporarily redistributed to another worker node, or broadcasted to all worker nodes during the join operation. Such redistribute and broadcast operations might take more time than choosing random distribution.

Selecting more than one column as the distribution key for a table results in a restricted generality of the key, meaning that queries joining tables now have to reference, rather than one, *all* of the distribution key columns to profit from collocated joins. In addition to that, in queries joining a table J with multiple tables K1, K2, …, Kn on different columns, only one of the pair can benefit from a collocated join. In such a situation, it is suggested to collocate the biggest tables, rather than defining a multi-column distribution key.

Although distribution keys using date, time, or time stamp columns would distribute data evenly across nodes, they should not be used if queries were to use the same column in range and equality predicates, because this would result in an uneven distribution of work. For example, if the data was to be distributed on the *month* column, a query retrieving sales revenue for a given month would only be processed by one worker node.

Altering the distribution key for an existing table requires a background reorganization job to be scheduled in order for the changes to take effect, which might negatively affect query performance while running.

---

[12] IBM DB2 Analytics Accelerator Studio lists the skew value for each table. A skew value of 0 implies that the table is not skewed, where a value greater than 0 denotes a skewed table. The higher the value, the greater the data skew.

[13] http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/index.jsp?topic=%2Fcom.ibm.datatools.aqt.doc%2Fgui%2Fconcepts%2Fc_idaa_dist_keys.html

4.5.2 Choosing organizing keys

Organizing keys define how table data is organized in a worker node. By setting organizing keys, it is possible to physically cluster records of a table with the same key column values. Up to four organizing keys are allowed per table. If no organizing key is set, no organization is imposed on the records in the table, which results in the data being physically arranged by the date and time of insertion. Organized tables are often called clustered base tables (CBT). Figure 4-15 shows the physical data layout of an unorganized and organized table.

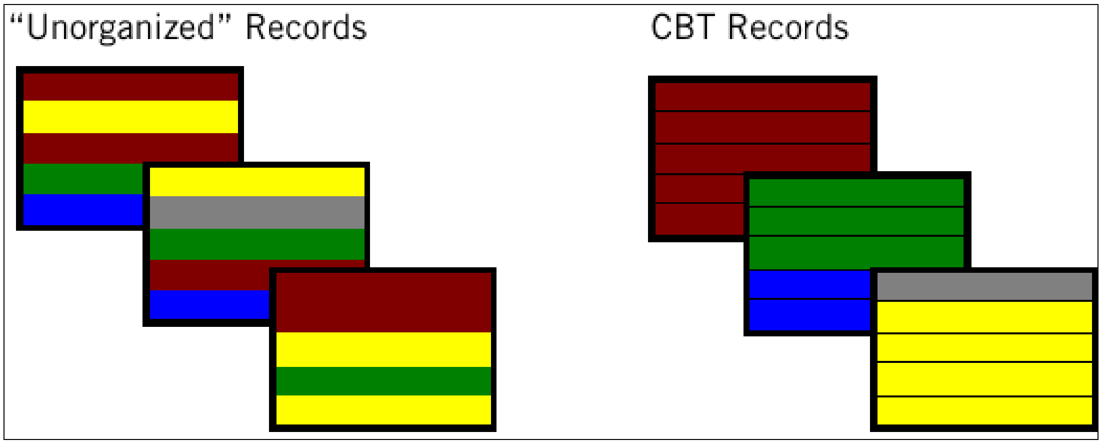


Figure 4-15 Structure of an unorganized and organized table

If range or equality predicates of incoming queries reference one or more organizing key columns, clustered base tables offer a performance advantage over unorganized tables. This enables the Netezza database system to skip entire blocks of data that are irrelevant for processing the current query, eliminating unnecessary disk scans. Unlike distribution keys, the generality of the keys is not negatively effected by defining multiple organization keys.

Optimizing a set of queries by setting organizing keys will not negatively affect the performance of residual queries, compared to not setting organizing keys at all. Organizing keys and zone maps work together to reduce the amount of data that needs to be scanned to process the data. Zone maps summarize the minimum and maximum range of data inside columns saved in specific database pages, and organizing keys help to narrow down the list of pages to be scanned by physically grouping records by organizing key columns.

Several factors support fully using the potential of organizing keys. This list contains combined suggestions for choosing well-suited distribution keys and organizing keys:

- Small tables will not benefit from the definition of organizing keys, due to the small amount of data to be scanned.
- Large tables containing millions of records and more benefit most from the definition of organizing keys, assuming that queries restrict on column values that are physically scattered across the table.
- The more the columns used as organizing keys are used in query predicates, the higher the benefit of organizing keys.
- The order of the columns defined as organizing keys does not matter.
- No preference exists for any of the organizing key columns.
- Not all organizing keys need to be referenced in a query in order for organizing keys to improve query performance. Even if only a subset of the columns is referenced, the query will benefit from faster processing.

Setting organizing keys during an initial load operation, or altering the organizing keys for an existing table, requires a background reorganization job to be scheduled in order for the changes to take effect, which might negatively affect query performance while running.

4.5.3 Optimizing for high query performance

Despite the high query performance that DB2 Analytics Accelerator delivers immediately upon implementation, special business requirements might necessitate tuning a defined set of queries for higher-than-normal query performance. In such situations, to boost query performance, it is necessary to identify and define distribution and organizing keys for tables referenced by those queries.

However, because this is a tuning measure, it is only possible to improve query performance for a subset of queries, at the expense of a possibly lower-than-normal query performance for the remaining queries. Therefore, it is suggested not to apply tuning measures, unless it is absolutely necessary. DB2 Analytics Accelerator is already delivering an unparalleled query performance with default settings, which should make tuning as it is known from regular database management systems unnecessary.

Optimizing queries for short elapsed times is not possible without an intimate knowledge of the workload. Therefore, defining the correct distribution and organizing keys is a time-consuming process. The first step in this process is to identify the set of queries that should be tuned. This can either be done by starting with a given set of queries derived from actual use cases and business requirements, or by identifying the longest running queries on DB2 Analytics Accelerator and using those.

Determining the longest-running queries on DB2 Analytics Accelerator can be achieved by importing DB2 Analytics Accelerator query history into a DB2 for z/OS table, as described in the following online document:

http://www.ibm.com/support/docview.wss?uid=swg27039739

After the set of queries that should be tuned has been identified, all queries should be ranked from 1 to n, where rank 1 is the query that has the highest priority to run fast, and rank n has the lowest priority to be tuned. Starting with the first query in the weighted list, an in-depth analysis has to be conducted to identify the tables referenced in join conditions, and the range and equality predicates used to restrict the query result set.

The former columns are used to define distribution keys for the referenced tables, so that collocated joins can be used. The latter columns are used to define organizing keys on the tables, so that large parts of table data can be skipped during the table scan, due to the use of zone maps. When choosing key candidates, the preferred practices outlined in the previous sections should be observed, to make sure that the best suitable keys are chosen per table.

In most cases, the join columns correspond to DB2 unique indexes, which generally are good candidates for distribution keys. The DB2 indexes are good candidates because they have a high cardinality, reducing the risk of data skew. When choosing a distribution key, it is suggested to start with the fact table, the table with the largest size, and identify the most suitable distribution key for that table.

After choosing a distribution key for the fact table, distribution keys should be chosen for the dimension tables, too, using the same criteria as applied to the fact table. Preferably, the same distribution key used for the fact table should be used for the dimension tables as well, if both columns are used together in a join condition, to instigate a collocated join.

However, small tables with a size of less than 100 megabytes (MB), and correspondingly less than 100,000 tuples, should always be distributed on random. Generally, it is suggested to refrain from using a multi-column distribution key, because all columns defined as distribution keys have to be used in a query, so that the query can profit from higher performance.

Improving query performance by means of defining organizing keys relates to identifying the columns used in equality and range predicates during query processing, and defining organizing keys for those columns. It is possible to define up to four organizing keys. Unlike distribution keys, the generality of the selection is not affected by defining more than one distribution key. Organizing keys will have a beneficial effect even if not all of the defined keys are used in a query.

If a query contains more than four range and equality predicates on a single table, it is suggested to use those predicates as organizing keys, which are more selective. Subsequently, all remaining queries have to be processed in the same way, starting with the higher-ranked queries. Naturally, improving query performance becomes harder with the number of queries to be tuned.

If most of the queries reference the same subset of tables, and if they use the same columns in join conditions, it is possible to tune all of those queries for a higher query performance. However, as soon as queries using different join columns are added to the mix, it becomes more complicated to optimize query performance. The database administrator has to prioritize the queries and optimization goals against each other.

Carefully chosen distribution and organizing keys can significantly improve elapsed times for queries. Figure 4-16 shows the effect of distribution and organizing keys on the Transaction Processing Performance Council Ad-Hoc (TPC BenchmarkH, or TPC-H) workload with a scale factor of 100 on an N1001-010 IBM DB2 Analytics Accelerator appliance. In this particular case, enhancements of up to 68% were achieved by using distribution and organizing keys.

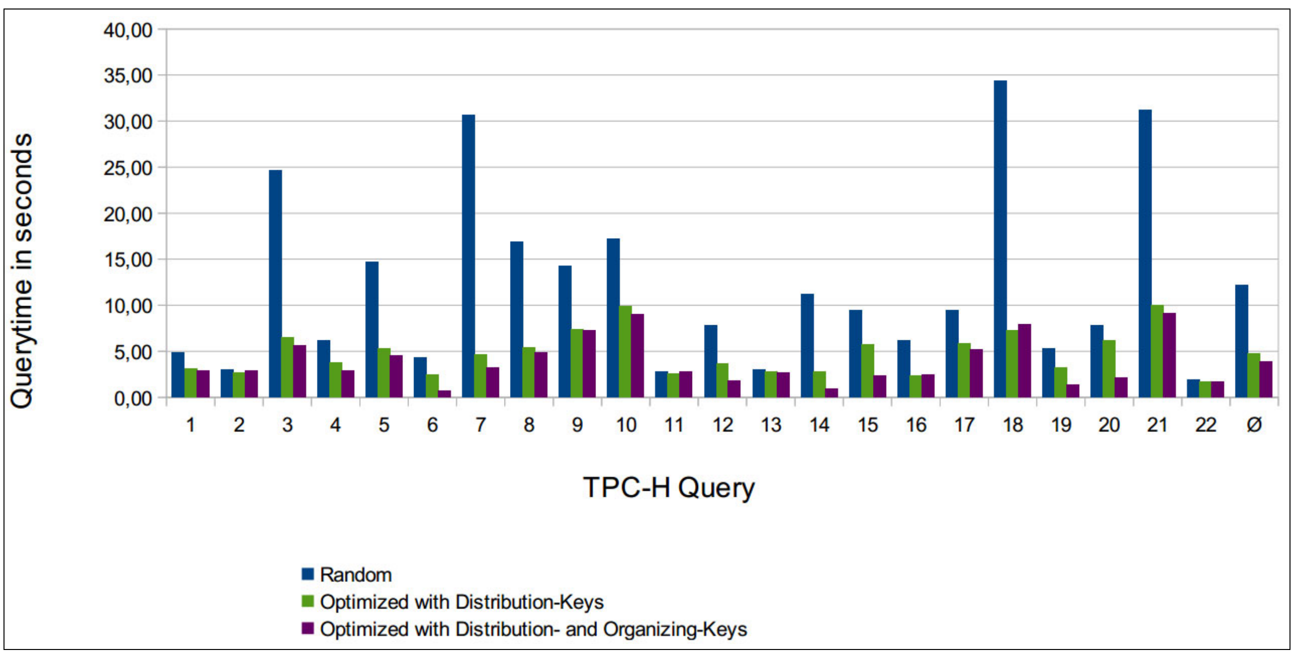


Figure 4-16 Effects of distribution and organizing keys on query performance

4.5.4 Optimizing for low incremental update latency

In most cases, query performance of DB2 Analytics Accelerator is already satisfactory. However, if the incremental update feature is used and a high volume of **UPDATE** and **DELETE** statements are to be expected, it makes sense to tune the system for a low incremental update latency.

Due to the fact that Netezza uses multi-version concurrency control to ensure that queries always see a consistent snapshot of the data, and that the system uses no indexes to access data, but always runs a full table scan, running **DELETE** statements is an expensive operation on Netezza. For each **DELETE** statement, all table data has to be scanned, and the tuple to be deleted has to be marked as deleted in the database system.

Internally, **UPDATE** statements are broken down into separate **DELETE** and **INSERT** statements, resulting in long processing times compared to **INSERT** operations, where a record is simply appended to the end of a page. DB2 Analytics Accelerator applies **INSERT** and **DELETE** statements in batches to reduce the number of individual table scans. However, by optimizing the inter-worker node and intra-worker node table layout, it is possible to speed up the **DELETE** processing during incremental update operations.

Experiments have shown that it is beneficial to set a distribution and organizing key on unique key columns of accelerated tables to expedite the processing of **DELETE** operations. Table 4-8 shows results for applying **INSERT** and **DELETE** operations with and without distribution and organizing keys defined.

The first two scenarios apply **INSERT** and **DELETE** operations as two single, large transactions, the latter two scenarios apply **INSERT** and **DELETE** operations in several, smaller transactions. In the two experiments, performance improvements of 27% and 36% were observed by setting distribution and organizing keys on the DB2 unique key columns.

Table 4-8 Comparison of total elapsed time for INSERT and DELETE with and without distribution and organizing keys

	Rows		Elapsed time (seconds)			
Test scenario	**Inserted**	**Deleted**	**INSERT**	**DELETE**	**Total**	**Improvement**
Separate **INSERT** and **DELETE** with random distribution	43,916,377	22,007,406	7093	5997	13,091	
Separate **INSERT** and **DELETE** with **DISTRIBUTE** and **ORGANIZE** on unique key	43,916,377	22,007,406	5200	5088	10,288	27%
Mixed **INSERT** and **DELETE** with random distribution	43,916,377	22,007,406	9232	5733	14,966	
Mixed **INSERT** and **DELETE** with **DISTRIBUTE** and **ORGANIZE** on unique key	43,916,377	22,007,406	6544	4448	10,992	36%

The following list includes DB2-side unique constraints that qualify as distribution and organizing key columns to DB2 Analytics Accelerator **UPDATE** and **DELETE** performance:

- DB2-side primary index (**PRIMARY KEY**)
- DB2-side unique constraint on non-nullable columns (**CREATE UNIQUE INDEX**)
- DB2-side unique constraint on nullable columns (**CREATE UNIQUE INDEX**)
- DB2 informal constraint (UNIQUE keyword in **CREATE TABLE** statement)

DB2-side unique constraints on nullable columns with WHERE NOT clause (**CREATE UNIQUE WHERE NOT NULL INDEX**) do not qualify for DB2 Analytics Accelerator incremental update operations.

If some tables are already optimized for high query performance by means of distribution keys, optimizing for low incremental update latency might be a contrary goal if the candidate keys for both goals were different, therefore restricting the generality of the selection. This restriction does not apply for organizing keys, because the generality of the selection is not restricted by the number of keys chosen.

The goals will only conflict if four organizing keys are already defined, and another one is necessary to optimize for incremental update latency. In this case, the trade-offs between query performance and incremental update latency have to be considered.

5

Query enhancements

IBM DB2 Analytics Accelerator for z/OS (DB2 Analytics Accelerator) and IBM DB2 for z/OS provide a single unique system for mixed dynamic query workloads.

This chapter mainly covers the new query acceleration features that were introduced in DB2 Analytics Accelerator V4.1. Also, some exiting query acceleration functionalities are described in this chapter, as background for the new features.

This chapter contains information about the following topics:

- New query acceleration features
- Query requirements
- Query acceleration settings and enablement
- Increase enablement of query acceleration
- Multi-row fetch queries
- Static Structured Query Language (SQL) queries acceleration
- Large result set and the cost of returned rows

5.1 New query acceleration features

DB2 Analytics Accelerator V4.1 introduced several enhancements in the query acceleration area:

- Static query acceleration (see 5.6, "Static SQL queries acceleration" on page 101)
- Local query acceleration when using multiple row fetch (see 5.5, "Multi-row fetch queries" on page 96)
- Workload balancing (see Chapter 8, "High availability, disaster recovery, and workload balancing" on page 159)

This section describes how query acceleration is expanded by new support for the following five functions:

- Support for more predicates
- FOR BIT DATA subtype support
- Support for a datetime 24-hour value
- Unmatched data type support
- S2upport for more online analytical processing (OLAP) specification expressions

Several of these functions have been introduced by DB2 Analytics Accelerator V4.1 program temporary fix 2 (PTF2), as described in the following release notes:

http://www-01.ibm.com/support/docview.wss?uid=swg27041549

5.1.1 Support for more predicates

Enhancements in DB2 Analytics Accelerator V4.1 and PTF2 enable more predicates. Queries using the following predicates are eligible for routing to DB2 Analytics Accelerator:

- NOT IN (<*subquery*>)
- <ALL (<*subquery*>)
- <=ALL (<*subquery*>)
- >ALL (<*subquery*>)
- >=ALL (<*subquery*>)
- =ALL (<*subquery*>)
- <>ALL (<*subquery*>)
- NOT IN (<*subquery*>)

For instance, the queries shown in Example 5-1 and Example 5-2 on page 87 are now both eligible for query acceleration.

Example 5-1 The NOT IN query is eligible for DB2 Analytics Accelerator offload

```
SELECT A.C1, A.C1, A.C2
FROM TABLEA A
WHERE A.C1 NOT IN (SELECT B.C1
      FROM TABLEB B
      WHERE B.C2 = 3);
```

Example 5-2 The <ALL query is eligible for DB2 Analytics Accelerator offload

```
SELECT A.C1, A.C1, A.C2
FROM TABLEA A
WHERE A.C1 <ALL (SELECT B.C1
      FROM TABLEB B
      WHERE B.C2=3);
```

5.1.2 FOR BIT DATA subtype support

DB2 Analytics Accelerator V4.1 with PTF 2 supports query acceleration for the queries referencing the Extended Binary Coded Decimal Interchange Code (EBCDIC) FOR BIT DATA data type. When the statement references a FOR BIT DATA data type expression, the statement must only reference EBCDIC single-byte character set (SBCS) and EBCDIC FOR BIT DATA expression to be eligible for query acceleration.

For example, the sample query in Example 5-3 is eligible for query acceleration, and the sample query in Example 5-4 is not eligible for query acceleration.

Example 5-3 The query FOR BIT DATA is eligible for DB2 Analytics Accelerator offload

```
CREATE TABLE TABLEA (
   C1 CHAR(10),
   C2_FORBIT CHAR(10) FOR BIT DATA,
   C3_MIXED VARCHAR(10) FOR MIXED DATA)
CCSID EBCDIC;

SELECT C1, C2_FORBIT
FROM TABLEA A
   WHERE C1_FORBIT < 0x'8080');
```

Example 5-4 The query with MIXED data is not eligible for DB2 Analytics Accelerator offload

```
SELECT C2_FORBIT, C3_MIXED
FROM TABLEA A
   WHERE C1_FORBIT < 0x'8080');
```

5.1.3 Support for a datetime 24-hour value

A query containing a datetime value with 24 hours (TIME value 24:00:00 and TIMESTAMP values with the time part 24:00:00) can now be successfully run in DB2 Analytics Accelerator V4.1 with PTF 2.

Example 5-5 shows a sample datetime query.

Example 5-5 The datetime query is eligible for DB2 Analytics Accelerator offload

```
SELECT TRANSACTION_ID
FROM TRANSACTION
    WHERE TRANSACTION_TS > '2013-01-23 24:00:00';
```

5.1.4 Unmatched data type support

DB2 10 for z/OS has enabled the compatibility of string and numeric data types, so that a string expression can compare with a numeric expression or a string expression can have arithmetic operation with a numeric expression. DB2 Analytics Accelerator V4.1 enabled the query acceleration for this DB2 10 functionality.

Example 5-6 shows an example. In this case TRANSACTION_ID is a CHAR(10) column.

Example 5-6 The query with unmatched data type is also eligible for DB2 Analytics Accelerator offload

```
SELECT TRANSACTION_ID
FROM TRANSACTION
   WHERE TRANSACTION_ID > 123;
```

5.1.5 Query acceleration support for more OLAP specification expressions

The following three OLAP functions are supported for query acceleration:

- CORR
- COVAR
- COVAR_SAMP

Together with DB2 Analytics Accelerator V3.1 support, here is a table with the currently supported OLAP specification expression list.

OLAP specification expressions			
AVG	CORRELATION	COUNT	COUNT_BIG
COVARIANCE	COVARIANCE_SAMP	MAX	MIN
STDDEV	SUM	VARIANCE	VARIANCE_SAMP
RANK	DENSE_RANK	ROW_NUMBER	

The query in Example 5-7 is eligible for query acceleration.

Example 5-7 The query with CORR function is eligible for DB2 Analytics Accelerator offload

```
SELECT SALES_DATE,
      CORR(SALES_AMOUNT) OVER (ORDER BY SALES_DATE RANGE 00000306. PRECEDING )
      AS CUM_SALES
FROM SALES;
```

5.2 Query requirements

There are types of queries that are not supported, and some query functionality limitations that preclude the query routing.

5.2.1 Query types requirements

A query needs to satisfy the following mandatory criteria to be considered for query acceleration:

- The cursor is not defined as a scrollable or row set cursor that is remote.
- The query is defined as read-only.
- The query is from a package, and not a plan with database request modules (DBRMs).
- The query is a SELECT statement, or an INSERT FROM SELECT statement.

  If the query is a SELECT statement, DB2 only accelerates the whole query, not a query block (part of a query), which means that the whole query will either run in DB2, or run in DB2 Analytics Accelerator, but not both.

  If a query is an INSERT FROM SELECT statement, and subsystem parameter QUERY_ACCEL_OPTIONS specifies number 2, DB2 will accelerate the whole SELECT statement. The INSERT operation is processed by DB2. Query acceleration criteria of the INSERT FROM SUBSELECT statement are the same as for the SELECT statement.

5.2.2 Query functionality limitations

DB2 will accelerate a query only when the SQL functionality required to run the query is supported by DB2 Analytics Accelerator. The following list describes the query functionalities that are not supported by DB2 Analytics Accelerator:

- Encoding scheme of the statement is multiple encodings. This can be either because tables are from different encoding schemes, or because the query contains a coded character set identifier (CCSID)-specific expression, for example, a cast specification expression with a CCSID option.

  If the statement references any FOR BIT DATA, the whole statement must only reference EBCDIC FOR BIT DATA or EBCDIC SBCS expressions.

- The query FROM clause specifies data-change-table-reference; that is, the query is select from FINAL TABLE or select from OLD TABLE.

- The query contains a *correlated table expression*. A correlated table expression is a table expression that contains one or more correlated references to other tables in the same FROM clause. Regular correlated queries might qualify for acceleration.

- The query contains a recursive common table expression reference.

- The query contains a string expression (including a column) with an unsupported subtype.

  The following list includes the supported subtypes:

  - EBCDIC SBCS
  - EBCDIC FOR BIT DATA
  - ASCII SBCS
  - UNICODE SBCS
  - UNICODE MIXED
  - UNICODE double-byte character set (DBCS) (graphic)
  - EBCDIC MIXED
  - EBCDIC DBCS (graphic)

> **Note:** EBCDIC MIXED and EBCDIC DBCS(GRAPHIC) are supported only when the QUERY_ACCEL_OPTIONS includes number 1 and SYSACCEL.SYSACCELERATEDTABLES.SUPPORTLEVEL value is 2 or greater.
>
> EBCDIC FOR BIT DATA is supported only when SYSACCEL.SYSACCELERATEDTABLES.SUPPORTLEVEL value is 4 or greater.

- The query contains an expression with an unsupported result data type.

  The following list includes supported result data types[1]:

  - CHAR
  - VARCHAR
  - GRAPHIC (non-ASCII)
  - VARGRAPHIC (non-ASCII)
  - SMALLINT
  - INT
  - BIGINT
  - DECIMAL
  - FLOAT
  - REAL
  - DOUBLE
  - DATE
  - TIME
  - TIMESTAMP with precision 6
- The query refers to a column that uses a field procedure (FIELDPROC).
- The query (SQL statement) uses a special register *other than* the following values:
  - CURRENT DATE
  - CURRENT TIME
  - CURRENT TIMESTAMP
- The query contains a date or time expression in which LOCAL is used as the output format, or a CHAR function in which LOCAL is specified as the second argument.

  Exception: If QUERY_ACCEL_OPTIONS 4 is specified, this restriction does not apply. See the subsystem parameter QUERY_ACCEL_OPTIONS option 4 for more details.
- The query contains a sequence expression (NEXTVAL or PREVVAL).
- The query contains a user-defined function (UDF).
- The query contains a ROW CHANGE expression.
- A date, time, or timestamp duration is specified in the query. Only labeled durations are supported.
- The query contains a string constant that is longer than 16,000 characters.
- A new column name is referenced in a sort-key expression, as shown in the following example:

  ```
  SELECT C1+1 AS X, C1+2 AS Y FROM T WHERE ... ORDER BY X+Y;
  ```
- The query contains a correlated scalar-fullselect. Here *correlated* means that the scalar-fullselect references a column of a table or view that is named in an outer subselect.

[1] The listed types refer to the built-in types. User-defined types (UDTs) are not allowed.

- The query contains one of the scalar functions in the function list, and also satisfies these three conditions:
  - CODEUNIT16, OCTETS, or NO CODEUNITS is specified.
  - QUERY_ACCEL_OPTION does not specify 3.
  - Non-SBCS string argument is contained.

  The following list includes the scalar functions:
  - SUBSTRING
  - CHARACTER_LENGTH
  - CAST specification
  - CHAR
  - VARCHAR
  - LOCATE
  - LOCATE_IN_STRING
  - LEFT
  - RIGHT
  - POSITION
- The query contains a cast specification with a result data type of GRAPHIC or VARGRAPHIC.
- The query contains one of the following scalar functions or cast specifications with a string argument that is encoded in Universal Transformation Format (UTF)-8 or UTF-16, and QUERY_ACCEL_OPTIONS does not specify option 3:
  - CAST(arg AS VARCHAR(*n*)) where *n* is less than the length of the argument
  - VARCHAR(arg, *n*) where *n* is less than the length of the argument
  - LOWER(arg, *n*) where *n* is not equal to the length of the argument
  - UPPER(arg, *n*) where *n* is not equal to the length of the argument
  - CAST(arg as CHAR(*n*))
  - CHAR
  - LEFT
  - LPAD
  - LOCATE
  - LOCATE_IN_STRING
  - POSSTR
  - REPLACE
  - RIGHT
  - RPAD
  - SUBSTR
  - TRANSLATE if more than one argument is specified
- The query uses a LENGTH function, but the argument of this function is not a string or is encoded in UTF-8 or UTF-16.
- The query uses a DAY function where the argument of the function specifies a duration.
- The query uses a MIN or MAX scalar function and the result data type is NOT INTEGER or FLOAT, or the scalar function contains more than 4 arguments.
- The query uses an EXTRACT function, which specifies that the SECOND portion of a TIME or TIMESTAMP value must be returned.
- The query uses one of the following aggregate functions with the DISTINCT option:
  - STDDEV
  - STDDEV_SAMP
  - VARIANCE
  - VAR_SAMP
- The query uses table functions (ADMIN_TASK_LIST, ADMIN_TASK_STATUS, and so on).

- The query references a scalar function that is *not* included in Table 5-1.

Table 5-1 List of DB2 scalar functions supported in DB2 Analytics Accelerator

Scalar functions			
ABS	FLOAT	MIDNIGHT_SECONDS	SIGN
ADD_MONTHS	FLOOR	MIN	SMALLINT
BIGINT	HOUR	MINUTE	SPACE
CEILING	IFFULL	MOD	SORT
CHAR	INTEGER	MONTH	STRIP
COALESCE	JULIAN_DAY	MONTHS_BETWEEN	SUBSTR
CONCAT	LAST_DAY	NEXT_DAY	TIME
DATE	LCASE	NULLIF	TIMESTAMP
DAY	LEFT	POSSTR	TIMESTAMP_FORMAT
DAYOFMONTH	LENGTH	POWER	TRANSLATE
DAYOFWEEK	LN	QUARTER	TRUNCATE
DAYOFWEEK_ISO	LOCATE	RADIANS	UCASE
DAYOFYEAR	LOCATE_IN_STRING	REAL	UPPER
DAYS	LOG10	REPEAT	VALUE
DECIMAL	LOG	REPLACE	VARCHAR
DEGREE	LOWER	RIGHT	VARCHAR_FORMAT
DIGIT	LPAD	ROUND	WEEK_ISO
DOUBLE	LTRIM	RPAD	YEAR
EXP	MAX	RTRIM	
EXTRACT	MICROSECOND	SECOND	
SUBSTRING	CHARACTER_LENGTH	POSITION	BITAND
BITANDNOT	BITOR	BITNOT	TIMESTAMPDIFF

5.3 Query acceleration settings and enablement

Dynamic query acceleration is controlled by the special registers CURRENT QUERY ACCELERATION and CURRENT GET_ACCEL_ARCHIVE.

- CURRENT QUERY ACCELERATION is a special register variable that identifies when DB2 sends queries to an Accelerator server, and what DB2 does if DB2 Analytics Accelerator server fails. It takes five possible values:
  - NONE
  - ENABLE
  - ENABLE WITH FAILBACK
  - ELIGIBLE
  - ALL

To enable query acceleration, CURRENT QUERY ACCELERATION must be set as a value other than NONE.

- The special register CURRENT GET_ACCEL_ARCHIVE specifies whether a query that references an archived table uses the archived data. The possible values are YES and NO.

These special registers' values can be set at different scope levels:

- System scope

  Subsystem parameters QUERY_ACCELERATION and GET_ACCEL_ARCHIVE set the default value of the special register at system level.

- Statement scope

  Special registers CURRENT QUERY ACCELERATION and CURRENT GET_ACCEL_ARCHIVE can be set to control an individual SQL statement.

- Application scope

  The bind option QUERYACCELERATION and GETACCELARCHIVE can be specified to initialize the special register CURRENT QUERY ACCELERATION and CURRENT GET_ACCEL_ARCHIVE values that are used for dynamic queries if an explicit SET statement has not already been done before the package or program is started.

  If the bind option is set, it will override the subsystem parameter default value. The bind option can be specified for Java Database Connectivity (JDBC) and Open Database Connectivity (ODBC) applications, and BIND PACKAGE can be qualified by any identifier that is supported by DB2.

- Through profile table facility

  In DB2 11, for the remote application, the profile monitoring SPECIAL_REGISTER keyword can be set to override a default special register value. If a profile filter matches a connection attribute, DB2 automatically applies the special register value to the DB2 process when the connection is initially established, and when a connection is reused.

  DB2 sets any profile table-derived values for the connection after processing all special register settings provided by the client before the first SQL statement run on the connection.

The following list is a step-by-step sequence showing how to set the CURRENT QUERY ACCELERATION through the profile table.

1. Create the profile monitoring tables. A complete set of profile tables includes the following objects:

   – SYSIBM.DSN_PROFILE_TABLE
   – SYSIBM.DSN_PROFILE_HISTORY
   – SYSIBM.DSN_PROFILE_ATTRIBUTES
   – SYSIBM.DSN_PROFILE_ATTRIBUTES_HISTORY

   The SQL statements for creating the profile tables and the related indexes can be found in member `DSNTIJOS` of the `SDSNSAMP` library.

2. Insert rows into SYSIBM.DSN_PROFILE_TABLE to create a profile. The value that you specify in the PROFILEID column identifies the profile, and DB2 uses that value to match rows in the SYSIBM.DSN_PROFILE_TABLE and DSN_PROFILE_ATTRIBUTES tables.

   Specifying different columns in DSN_PROFILE_TABLE as filtering criteria can define different scopes for SQL statements. For example, you can create a global profile with PROFILE ID 1:

   ```
   INSERT INTO SYSIBM.DSN_PROFILE_TABLE (PROFILEID) VALUES (1);
   ```

You can also create a profile for SQL statements from a specific authorization ID and IP address with PROFILE ID 2:

```
INSERT INTO SYSIBM.DSN_PROFILE_TABLE
(PROFILEID,AUTHID,IPADDR,PLANNAME,COLLID,PKGNAME,PROFILE_TIMESTAMP) VALUES
(2,'TOM','9.10.128.3',NULL,NULL,NULL,CURRENT TIMESTAMP);
```

In the DSN_PROFILE_TABLE, you now see the rows listed in Table 5-2.

Table 5-2 Sample data in the SYSIBM.DSN_PROFILE_TABLE table

AUTHID	PLAN NAME	COLL ID	PKG NAME	IPADDR / LOCATION	PROFILEID	PROFILE_ ENABLED
null	null	null	null	null	1	Y
TOM	null	null	null	9.10.128.3	2	Y

3. Insert rows into the SYSIBM.DSN_PROFILE_ATTRIBUTES table to define the type of monitoring:

   PROFILEID column: Specify the profile that defines the statements that you want to monitor. Use a value from the PROFILEID column in SYSIBM.DSN_PROFILE_TABLE.

   KEYWORDS column: Specify one of the following monitoring keywords: ACCEL_TABLE_THRESHOLD, ACCEL_CHECK_RESULTSIZE, ACCEL_RESULTSIZE_THRESHOLD, SPECIAL_REGISTER, and so on.

   ATTRIBUTEn column: Specify the appropriate attribute values depending on the keyword that you specify in the KEYWORDS column.

   For example, suppose that you have inserted the row listed in Table 5-3 into the SYSIBM.DSN_PROFILE_ATTRIBUTES table for PROFILEID 1.

Table 5-3 Sample data in SYSIBM.DSN_PROFILE_ATTRIBUTES table

PROFILEID	KEYWORDS	ATTRIBUTE1	ATTRIBUTE TIMESTAMP
1	SPECIAL_REGISTER	SET CURRENT QUERY ACCELERATION= ENABLE	2014-04-24...

   This row specifies that DB2 sets the CURRENT QUERY ACCELERATION special register to ENABLE when referenced by dynamic SQL in the application that is included in profile 1.

4. Load or reload the profile tables into memory by issuing the **`START PROFILE`** DB2 command.

   Rows with a `Y` in the PROFILE_ENABLED column in SYSIBM.DSN_PROFILE_TABLE are now in effect. DB2 monitors any statements that meet the specified criteria.

5. When you finish monitoring these statements, stop the monitoring process by performing one of the following tasks:

   – To disable the monitoring function for a specific profile, delete that row from DSN_PROFILE_TABLE, or change the PROFILE_ENABLED column value to `N`.

     Then, reload the profile table by issuing the **`START PROFILE`** DB2 command.

   – To disable *all* monitoring and subsystem parameters that have been specified in the profile tables, issue the **`STOP PROFILE`** DB2 command.

Static query acceleration is controlled by the new bind options QUERYACCELERATION and GETACCELARCHIVE. The default values for both QUERYACCELERATION and GETACCELARCHIVE are not specified. To enable static query acceleration, QUERYACCELERATION must be set to a value other than NONE.

5.4 Increase enablement of query acceleration

Some queries are not routed to DB2 Analytics Accelerator due to the DB2 optimizer verification, based on current restrictions or incomplete information. This section describes ways to improve eligibility in the following cases:

- Ambiguous queries
- Lack of selective predicates
- Correlated subqueries

5.4.1 Ambiguous queries

An *ambiguous* query is a query that does not have either a FOR READ ONLY, FOR FETCH ONLY, FOR UPDATE OF, ORDER BY, or GROUP BY clause, nor does the query have join, or aggregation, or DISTINCT. In these cases, DB2 is not sure whether the query or cursor is used for update, therefore the query cannot be offloaded to DB2 Analytics Accelerator for acceleration.

For example, consider a simple query that selects a column from a table with a predicate:

```
SELECT COL2 FROM TABLEX WHERE COL1=‘y’;
```

This is an ambiguous query. This query cannot be offloaded to DB2 Analytics Accelerator in spite of the fact that the query speedup on DB2 Analytics Accelerator can be significant.

Using WITH UR in the query or BIND with ISOLATION(UR) will make the query read only, and enables the query to be considered for offload.

Another workaround to enable query acceleration for an ambiguous query is to add one of the following clauses:

- `FOR READ ONLY`
- `FOR FETCH ONLY`

A query that has a FOR READ ONLY or FOR FETCH ONLY clause is no longer ambiguous, and it can be offloaded to DB2 Analytics Accelerator for acceleration.

5.4.2 Lack of selective predicates

If a query does not have a WHERE predicate, and it does not have GROUP BY aggregate functions specified (all rows are to be returned), and you set QUERY_ACCELERATION to ENABLE, the query will not currently be offloaded to DB2 Analytics Accelerator. One example of this is a simple query:

```
SELECT KEY  FROM FACT_TABLE X ORDER BY X.KEY FETCH FIRST 1000000 ROWS ONLY;
```

With four IBM zEnterprise EC12 (zEC12) processors and query parallelism degree 8, the query elapsed time on System z is 999 seconds and query processor time on System z is 3721 seconds. Because the sort cost for a large table is relatively high, this query should be offloaded to DB2 Analytics Accelerator to improve query performance.

One workaround is to add a redundant predicate, such as `1=1`, to satisfy the heuristics checking by IBM DB2 Optimizer on the existence of a predicate. Query performance improves significantly, with only 59 seconds of query elapsed time when the query is offloaded to the IBM PureData System for Analytics N2001-010 accelerator (Striper full rack) powered by IBM Netezza technology. Another workaround would be to set the QUERY_ACCELERATION register to ELIGIBLE or ALL.

5.4.3 Correlated subqueries

Certain correlated subqueries *cannot* currently be offloaded to DB2 Analytics Accelerator. For example, correlated subquery in the following cases:

- SET operations (UNION, INTERSECT, EXCEPT, and so on)
- Aggregates with GROUP BY and HAVING clauses
- CASE or WHEN expressions
- Clauses that are connected by the OR operator

DB2 Analytics Accelerator transforms the correlated subqueries in a `WHERE` clause to equivalent `JOIN` expressions.

Depending on the form and level of nesting, DB2 Analytics Accelerator might be unable to run the correlated subquery and get an error. If there is an error during `PREPARE` or first `OPEN`, the correlated subquery can be run on DB2 if the `CURRENT QUERY ACCELERATION` special register is set to `ENABLE WITH FALLBACK`.

A workaround to enable acceleration for query failback is for the user to rewrite the subquery from correlated to non-correlated. For reference, see the following website:

http://www.ibm.com/support/docview.wss?uid=swg27037928

5.5 Multi-row fetch queries

The objective of this feature is to use multi-row fetch (MRF) to improve performance for a local accelerated query with a large result set. MRF can also improve query elapsed time if the result set output processing from DB2 to the application is the bottleneck. When the application fetches data efficiently using MRF, and when query execution time on DB2 Analytics Accelerator is very small, we measured a reduction of up to 94.6% of the query elapsed time for queries with large result sets.

To benefit from this feature, application change is needed if ROWSET or MRF is not yet enabled.

- Specify WITH ROWSET POSITIONING in DECLARE:

```
EXEC SQL DECLARE DYN1 CURSOR WITH ROWSET POSITIONING FOR STMT1;
```

- Alternatively, specify WITH ROWSET POSITIONING in PREPARE:

```
STMTSTR =  'SELECT COL2, COL4, COL8'
                   || ' FROM  FACT_TABLE'
                   || ' WHERE  COL2 > 0'
                   || ' ORDER BY  COL4'
                   || ' FOR FETCH ONLY';

ATTRSTR = 'WITH ROWSET POSITIONING';
EXEC SQL PREPARE STMT1 ATTRIBUTES :ATTRSTR FROM :STMTSTR;
```

- Use FETCH NEXT ROWSET with a FOR N ROWS clause to fetch:

```
DCL INT2B(100) BIN FIXED(15);
DCL INT4B(100) BIN FIXED(31);
DCL INT8B(100) BIN FIXED(63);
NROWS = 100;
EXEC SQL FETCH NEXT ROWSET  FROM  DYN1 FOR :NROWS ROWS INTO
  :INT2B,:INT4B, :INT8B;
```

There are some restrictions on MRF enhancement in DB2 Analytics Accelerator V4.1:

- The query acceleration of ROWSET queries or MRF does not apply to remote queries.
- The query acceleration of ROWSET queries does not apply if the query is part of an SQL Procedural Language (SQL PL) routine.
- The query acceleration of ROWSET queries or MRF does not apply if the query using cursor declared WITH RETURN clause.

Before DB2 Analytics Accelerator V4.1, queries cannot be offloaded to DB2 Analytics Accelerator if rowset cursor or MRF is used. DB2 Analytics Accelerator V4.1 supports offloading both dynamic and static local queries with rowset cursor and MRF, except for SQL PL. Both DB2 10 and DB2 11 for z/OS support this feature. DB2 10 support is added by authorized program analysis reports (APARs) PM93786, PM93787, PM93788, and PM93789.

The following code is an example of rowset cursor:

```
DECLARE CUR1 CURSOR WITH ROWSET POSITIONING FOR STATEMENT1;
```

The following code is an example of MRF:

```
FETCH NEXT ROWSET FROM  CUR1 FOR 100 ROWS INTO :hv
```

MRF reduces System z processor usage significantly for queries with large result sets.

Fetch 100 rows per fetch as a starting point. The optimal number of rows per fetch is query-dependent.

The following sections describe aspects of fetch operations:

- Measurements of single versus multi-row fetch
- Number of rows per fetch
- Measurements of local versus remote fetch

5.5.1 Measurements of single versus multi-row fetch

Using MRF can save up to 95.5% of z/OS-side processor time for a simple query that fetches millions of rows, because most of the time spent is returning data from DB2 to the application. When the application fetches data efficiently using MRF, we see that queries with large result sets saved up to 94.6% of the end-to-end elapsed time when using MRF. Query performance improvement from MRF is highly dependent on the query, the workload, and the test environment. How much improvement you get in your situation can vary.

Queries select up to 16 integer/decimal columns and fetch up to 40 million rows:

```
SELECT COLC FROM FACT_TABLE WHERE COLC > 0 FETCH FIRST N ROWS ONLY;
```

Processor use reduction varies depending on the number of columns, the number of rows, and the data types in the result set. A simple query that selects one column with a large answer set has a big improvement using MRF.

Measurements are done using a set of queries executed in DB2 Analytics Accelerator returning from a few million rows to up to 40 million rows. DB2 11 running on a z/OS logical partition (LPAR) with four dedicated zEC12 central processor units (CPUs), and DB2 Analytics Accelerator V4.1 on PureData for Analytics N2001-010 are used for the measurements. The measurements use a static application returning millions of rows through a Time Sharing Option (TSO) batch. A default of 100 rows per fetch is used.

To study the improvement from MRF, a simple query that selects a column and fetches millions of rows from a large table (`SELECT COLC FROM FACTTAB WHERE COLC>0...`) is measured. Figure 5-1 on page 99 shows DB2 query CPU time to fetch 5, 10, 20, and 40 million rows by single-row fetch (SRF) and MRF. There are four measurements for each scenario:

- The first measurement data shows query CPU time on System z using SRF.
- The second measurement data shows query CPU time on System z using MRF.
- The third measurement data shows query CPU time on System z using SRF, when the query is executed on DB2 Analytics Accelerator.
- The fourth measurement data shows query CPU time on System z using MRF, when the query is executed on DB2 Analytics Accelerator.

In the first two scenarios, query acceleration is disabled. Query acceleration is enabled for the third and the fourth scenarios.

The test result shows that when a query is run on System z only, MRF reduces System z central processor unit CPU usage time by about 60% versus SRF. When the query is executed on DB2 Analytics Accelerator, and the fetch is SRF, query CPU time on System z increases significantly, because there is inefficiency in the result set processing.

The good news is the new MRF support for accelerated query is efficient. We witnessed about 22 times System z CPU reduction and about 20 times query elapsed time reduction using MRF versus SRF for this query executed on DB2 Analytics Accelerator.

Elapsed time reduction varies depending on the factors mentioned previously, and on query complexity. The larger the result set processing time on System z versus the processing time on DB2 Analytics Accelerator, the larger the percentage improvement in query elapsed time.

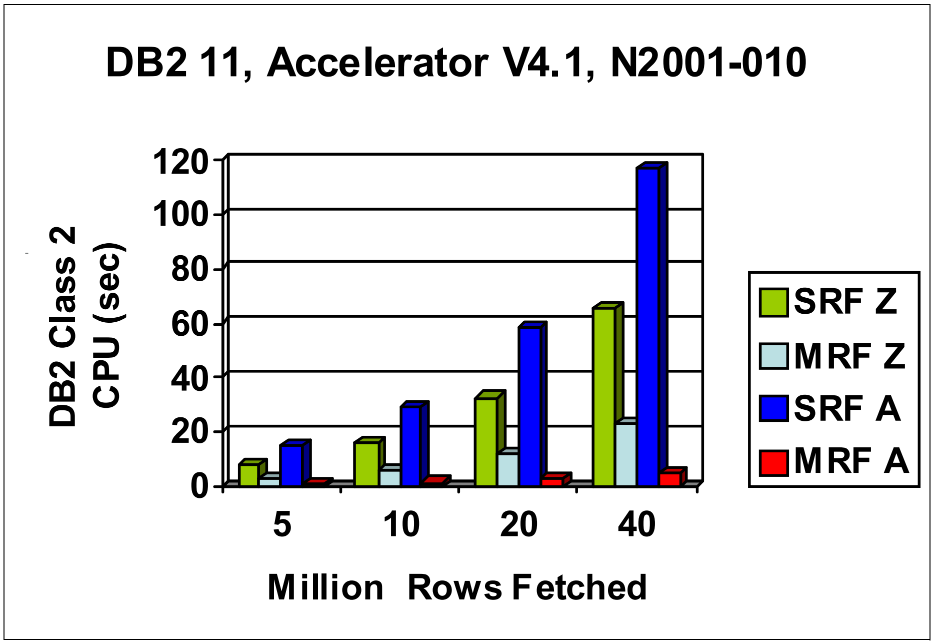


Figure 5-1 MRF versus SRF Improvements in DB2 11

5.5.2 Number of rows per fetch

The optimal number of rows per fetch in MRF depends on how many columns are fetched, column data types, how many rows are fetched, and other factors.

The following list describes the components in the measurement environment:

- Four zEC12 CPUs, 128 gigabytes (GB) memory
- z/OS V2.1
- DB2 10 or DB2 11
- DB2 Analytics Accelerator V4.1
- PureData for Analytics N2001-010
  - Netezza Performance Server (NPS) 702P9

Figure 5-2 on page 100 shows the effect of the number of rows fetched per fetch in MRF. This query fetches 40 million rows from the accelerated table. With SRF queries that fetch one row per fetch, query elapsed time is about 123 seconds and query CPU time was about 120 seconds.

With MRF fetching 50 rows per fetch, query elapsed time was 8 seconds and query CPU time was 5.8 seconds. Query CPU time was reduced by about 20 times. With MRF fetching 100 rows per fetch, query elapsed time was 6.7 seconds and query CPU time was 4.2 seconds. Query CPU time improved about 28 times.

Figure 5-2 shows results per number of rows fetched using MRF.

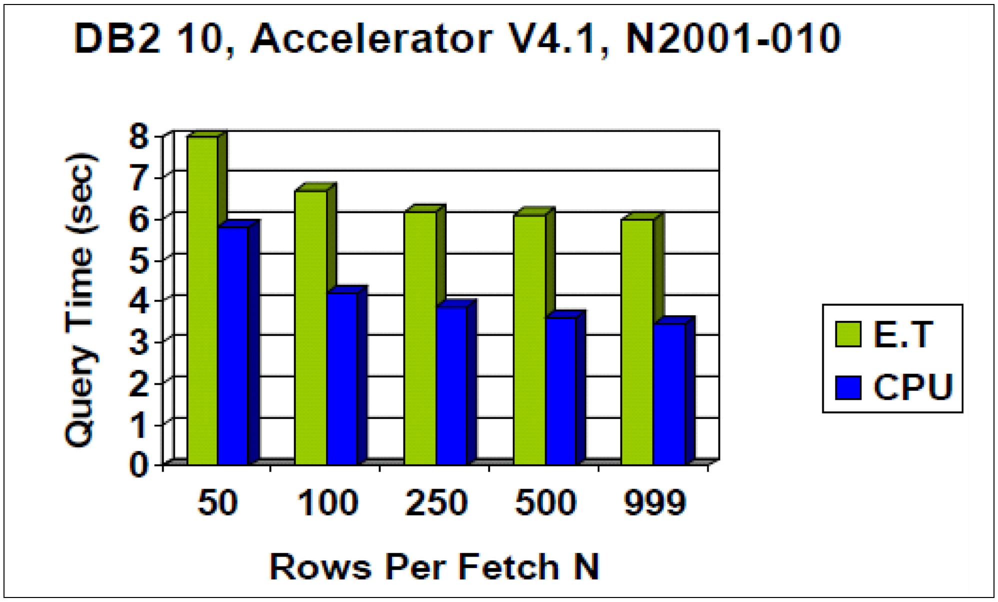


Figure 5-2 The optimal number of rows per fetch

5.5.3 Measurements of local versus remote fetch

Figure 5-3 on page 101 compares the query elapsed time and query CPU time for a simple query that fetches 40 million rows in four scenarios:

- The first scenario is remote query from requester to DB2 using DDF, and the query is executed on System z only (DDF Z).
- The second scenario is remote query from requester to DB2 using DDF, and the query is executed on DB2 Analytics Accelerator (DDF A).
- The third scenario is local query that is executed on DB2 server (Local Z).
- The fourth scenario is local query that is accelerated on DB2 Analytics Accelerator (Local A).

Local query that is accelerated on DB2 Analytics Accelerator (Local A) has the best query elapsed time and the best query CPU time among the four scenarios. Local query offloaded to DB2 Analytics Accelerator improved about 3.5 times compared to the local query executed on System z.

The improvement is mainly due to the massively parallel processing (MPP) on DB2 Analytics Accelerator that starts 240 parallel tasks to retrieve large result sets from 240 disks. The degree of query parallelism on System z is 20 for the local query executed on System z.

With MRF support for local accelerated query in DB2 Analytics Accelerator V4.1, query CPU time for the local accelerated query (Local A) is a little better than the same query from remote requester to DB2 (DDF A). Query elapsed times in the DDF Z scenario and in the DDF A scenario are higher than those in the Local Z scenario and in the Local A scenario, mainly due to the network time to send 40 million rows in the result set.

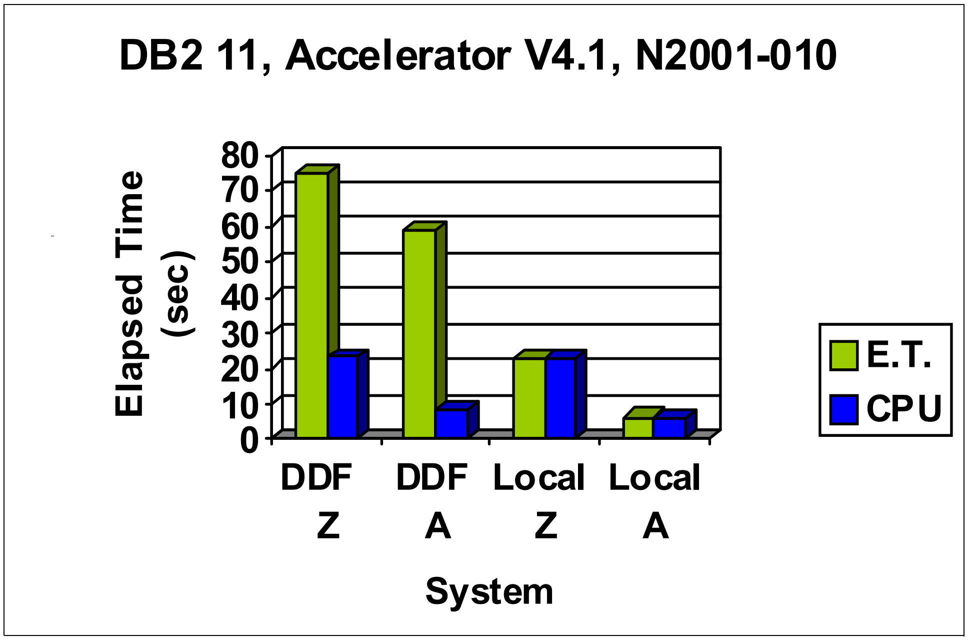


Figure 5-3 Local versus remote accelerated query in DB2 11

5.6 Static SQL queries acceleration

DB2 Analytics Accelerator V3.1 enables acceleration only for dynamic queries. DB2 Analytics Accelerator V4.1 adds the enablement of static queries to satisfy this requirement from many DB2 for z/OS users who rely heavily on static SQL. This enables clients to route more queries from a wider base of applications to DB2 Analytics Accelerator. In addition, this support can eliminate a great deal of the administrative tasks previously needed when identifying tables that had only dynamic SQL queries to be directed to DB2 Analytics Accelerator.

5.6.1 The BIND options for DB2 Analytics Accelerator

Additional bind options are supported for static queries and apply to the following operations:

- BIND, BIND COPY, or REBIND PACKAGE (local and remote binds)
- BIND DEPLOY (for SQL PL procedures), REBIND TRIGGER PACKAGE
- ALTER or CREATE PROCEDURE for native SQL PL procedures
- ALTER or CREATE FUNCTION for SQL PL scalar functions

When binding a package with ENABLE, ENABLEWITHFAILBACK, or ELIGIBLE, only some queries in a package might be bound for execution on DB2 Analytics Accelerator, depending on query characteristics. This would result in some kind of hybrid package with some statements executing in DB2 for z/OS, and some in DB2 Analytics Accelerator. It depends on the application if it can tolerate this behavior. Queries might obtain different results.

QUERYACCELERATION

The QUERYACCELERATION bind options specify whether a static SQL query is bound for acceleration, and if so, with what behavior.

The QUERYACCELERATION bind value overrides the QUERY_ACCELERATION subsystem parameter value, and provides the initial value for the CURRENT QUERY ACCELERATION special register.

Values for the new QUERYACCELERATION BIND options are:

NONE
: No static SQL query in the application is bound for acceleration, or will be accelerated when the application is run.

ENABLE
: The static SQL query is bound for acceleration if it satisfies the acceleration criteria, including the cost and heuristics criteria. The query is routed to an accelerator when the application runs. Otherwise, if the static query does not satisfy the acceleration criteria, the query is bound for execution in DB2.

ENABLE WITH FAILBACK
: The same as ENABLE, except if one of the error conditions occurs on the first OPEN of the accelerated static query when the application is run. In this case, rather than failing the static query and returning a negative SQL code to the application, DB2 performs a temporary statement-level incremental bind of the query, and runs the query in DB2.

ELIGIBLE
: The same as ENABLE, but the static SQL query is bound for acceleration if the query meets the basic acceleration criteria, regardless of the cost or heuristics criteria. The query is routed to DB2 Analytics Accelerator when the application runs.

ALL
: All of the static SQL queries in the application are bound for acceleration and routed to DB2 Analytics Accelerator when the application runs. If DB2 determines that a static query cannot be bound to run on DB2 Analytics Accelerator, and the query references a user base table or view, the BIND or REBIND PACKAGE operation fails with an error message for that query.

The SYSIBM.SYSPACKSTMT DB2 catalog table has a STATUS column value of “O” for static SQL queries that have been bound for acceleration. For cursor-based static queries, the DECLARE CURSOR, OPEN, FETCH, and CLOSE statements each have a SYSPACKSTMT.STATUS column value of “O”. For a static INSERT from SELECT statement, where the SELECT has been bound for acceleration, the SYSPACKSTMT.STATUS column value is also “O”.

GETACCELARCHIVE

The GETACCELARCHIVE bind option specifies whether a static SQL query that is bound for acceleration retrieves archived data on DB2 Analytics Accelerator, rather than active data. The GETACCELARCHIVE value overrides the GET_ACCEL_ARCHIVE subsystem parameter and provides the initial value for the CURRENT GET_ACCEL_ARCHIVE special register.

The following list describes the values for the new GETACCELARCHIVE BIND options:

NO
: No static SQL query is bound to retrieve archived data from DB2 Analytics Accelerator. If the static query also is not bound for acceleration, the query is bound to run in DB2.

: If the static query is bound for acceleration because the QUERYACCELERATION bind option was specified, the query is routed to DB2 Analytics Accelerator when the application runs. However, the query does not retrieve any archived data.

YES — Specifies that if all of the following criteria are met, the query is bound for acceleration and retrieves the archived data on DB2 Analytics Accelerator when the application runs:

The QUERYACCELERATION bind option is also specified.

The static SQL query references an accelerated table that has partitioned data archived on an accelerator.

The static query satisfies the acceleration criteria that is specified by the QUERYACCELERATION bind option.

Note that the special register does not affect static queries.

Default values for QUERY ACCELERATION and GETACCLEARCHIVE option and system parameters

The default value for both new bind options is `option not specified`, which means that the user is required to specify a value for the options to be effective. The corresponding DB2 system parameters QUERY_ACCELERATION and GET_ACCEL_ARCHIVE enable support for dynamic queries when no values are explicitly set for the CURRENT QUERY ACCELERATION special register by the SQL statement SET CURRENT QUERY ACCELERATION.

These system parameters do not dictate the default for static query acceleration. For static query acceleration, the ACCEL system parameter must be set to either AUTO or COMMAND. QUERY_ACCEL_OPTIONS option#2 needs to be set as 'YES' to enable the acceleration of static INSERT from SELECT statement. This is the same as for dynamic statements.

If you want queries in a static package to be accelerated, you must first add and enable the referenced DB2 tables to DB2 Analytics Accelerator. Then you BIND or REBIND the query package by specifying the wanted QUERYACCELERATION behavior to bind queries in that package for acceleration. QUERYACCELERATION BIND options are discussed in the previous figure.

If you want static queries to use data archived to DB2 Analytics Accelerator, you must first archive at least one partition of the referenced DB2 tables. Then you BIND or REBIND the query package by specifying GETACCELARCHIVE(YES) and the wanted QUERYACCELERATION behavior to bind queries in that package for acceleration, and to retrieve archived data. GETACCELARCHIVE BIND options are discussed in the following section.

BIND options QUERYACCELERATION and GETACCELARCHIVE

During BIND PACKAGE on active data with an ELIGIBLE or ENABLE acceleration option, DB2 Optimizer determines whether queries can be accelerated on DB2 Analytics Accelerator, based on the following requirements:

- The first requirement is that all tables in the query are added and enabled on DB2 Analytics Accelerator.
- The second requirement is that SQL functionality is supported on DB2 Analytics Accelerator.
- The third requirement is that the query passed the heuristics and cost checking when the acceleration option is ENABLE.
- The fourth requirement is that the behavior of bind options VALIDATE(RUN) and SQLERROR(CONTINUE) do not apply to the static query. Therefore, a static query requiring incremental bind at execution time will not be considered for acceleration.

At query execution time, a static query that is bound for acceleration is routed to DB2 Analytics Accelerator if DB2 Analytics Accelerator is started and active. If there is no DB2 Analytics Accelerator started or active, query execution fails with a negative SQLCODE. For a static query that does not qualify for acceleration, the query is executed in DB2.

For QUERYACCELERATION (ENABLEWITHFAILBACK), if the OPEN for a static query bound for acceleration fails when sent to the DB2 Analytics Accelerator, that static query will undergo an incremental bind to run in DB2 instead. No negative SQLCODE is returned to the application if the query runs successfully in DB2. After a COMMIT or on the next package execution, if that static query is run again, DB2 will try again to send that query to DB2 Analytics Accelerator, and no BIND or REBIND PACKAGE is needed.

For QUERYACCELERATION (ALL), during BIND PACKAGE, if any static query for a user base table or view is not eligible for acceleration, the BIND PACKAGE will fail with an error message even if VALIDATE(RUN) or SQLERROR(CONT) is used.

For a GETACCELARCHIVE(YES) and QUERYACCELERATION combination scenario, during BIND PACKAGE, DB2 will bind the query for acceleration and target the archived data on DB2 Analytics Accelerator if the following conditions are true:

- The target table of a static query has at least one data partition archived to a DB2 Analytics Accelerator (data is removed from DB2).
- The query qualifies for acceleration.

Note that there is a difference between active data on DB2 Analytics Accelerator versus archived data on DB2 Analytics Accelerator. If that static query does not qualify for acceleration, DB2 will fail the bind package with an error message or code. This is because GETACCELARCHIVE behavior was requested and the query’s table has at least one data partition archived (moved) to DB2 Analytics Accelerator, but DB2 could not accelerate the query.

For a scenario with GETACCELARCHIVE(NO) or not specified, and the DB2 table has archived data on a DB2 Analytics Accelerator scenario, the following outcomes are possible:

- If the static query is not bound to access archived data on a DB2 Analytics Accelerator, the query will run in DB2 and retrieve only active data, if any, that remains on DB2.
- If QUERYACCELERATION is specified and the static query is bound for acceleration, the query will run on DB2 Analytics Accelerator, but will only retrieve active data there.

5.6.2 DB2 catalog, pseudo-catalog, and EXPLAIN table

SYSACCEL tables are DB2 pseudo-catalog tables created by the user. The new bind options and values for static queries are stored in a new pseudo-catalog table (SYSACCEL.SYSACCELERATEDPACKAGES), not the standard catalog table (SYSIBM.SYSPACKAGE).

To support static SQL and workload balancing, the naming scheme for tables in the SYSACCEL.SYSACCELERATEDTABLES catalog table was changed. Entries in this pseudo-catalog table created by V3 or V2 of DB2 Analytics Accelerator do not provide the correct information that is required for a successful use of these features. An easy migration is provided by a new stored procedure (SYSPROC.ACCEL_MIGRATE). See 4.2.4, “Migrating SYSACCELERATEDTABLES table” on page 64 for details.

Run this stored procedure against each DB2 Analytics Accelerator that is supposed to participate in your workload balancing or static SQL scenario. The stored procedure always migrates the names of all tables on an accelerator.

If the new SYSACCEL table name does not exist, and the new bind options are specified, the BIND or REBIND PACKAGE fails with an error message or code.

If the new SYSACCEL table exists but is not defined correctly, any REBIND PACKAGE (or autobind) will fail even if the REBIND statement itself does not specify the new bind options. Note the following additional information regarding accelerated static queries:

- REBIND also attempts to pull any existing bind option values for the package from SYSIBM.SYSPACKAGE.
- The SYSIBM.SYSPACKSTMT column STATUS has a new value, “O”, for a static query that has been bound for acceleration during bind package.

This is one easy way to see static queries that are accelerated.

Unlike SYSIBM.SYSPACKSTMT, SYSIBM.SYSPACKAGE, and other DB2 catalog tables that are protected by DB2, user-created SYSACCEL tables can be dropped either intentionally or unintentionally.

Several non-trivial steps are needed to re-create and repopulate the SYSACCEL tables. It takes time to reload tables on DB2 Analytics Accelerator. Therefore, protecting SYSACCEL tables from being dropped by accident is important. Consider an ALTER TABLE operation to add ADD RESTRICT ON DROP:

```
ALTER TABLE SYSACCEL.xxxxxxxxx   ADD RESTRICT ON DROP;
```

The existing bind option EXPLAIN(YES | ONLY), specified along with the new bind options, causes the bind package process to insert a row for the static query in DSN_QUERYINFO_TABLE and PLAN_TABLE.

For REASON_CODE values other than 0 (query not qualified for acceleration), the DSN_QUERYINFO_TABLE rows for TYPE ='A' have the REASON_CODE value description in the QI_DATA column.

5.6.3 Which packages should be rebound for acceleration

To find those packages having static queries for a DB2 table that you want to accelerate and archive to DB2 Analytics Accelerator, you can run the following query:

```
SELECT DCOLLID, DNAME, BNAME, BQUALIFIER
FROM SYSIBM.SYSPACKDEP
WHERE BNAME = ‘table-name’ AND
BQAULIFIER = ‘table-qualifier’;
          DCOLLID = package collection-id, DNAME = package name
```

You can then determine which of these packages, programs, or queries you want to BIND or REBIND, specifying the new bind option QUERYACCELERATION (and GETACCELARCHIVE(YES) for archived data).

Suppose the package contains more than one query referencing the accelerated table, and you BIND or REBIND that package with QUERYACCELERATION (and, if applicable, GETACCELARCHIVE). During the BIND PACKAGE, DB2 applies the specified QUERYACCELERATION behavior to each of those specific static queries, and evaluates those queries for acceleration.

Be aware of hybrid packages (those with different types of SQL statements in a package that might be run in DB2 for z/OS and in DB2 Analytics Accelerator). For applications that read their own writes, this might lead to encountered inconsistencies. To make sure that all SQL statements in a given package execute in DB2 Analytics Accelerator, ALL is the BIND option to use.

5.6.4 Performance of accelerated static queries

From the perspective of DB2 Analytics Accelerator, it doesn't matter if the original query was executed as dynamic or static SQL in DB2, because all queries are executed dynamically in DB2 Analytics Accelerator.

BIND or REBIND processor resource usage in DB2 for acceleration decisions is query-dependent and workload-dependent. There are three phases during BIND or REBIND when DB2 optimizer determines whether a static query can be offloaded to DB2 Analytics Accelerator:

- First, the optimizer checks whether there is any restriction from a syntax or functionality perspective that prevents the query from being offloaded to DB2 Analytics Accelerator. If there is a restriction on the query, the query cannot be offloaded, and no further checking is done. The processor resource cost is very small in the first phase of checking.
- In the second phase in decision making, DB2 Optimizer checks whether the query passes the heuristic rules on query acceleration. If the query fails the heuristic checking, it cannot be offloaded, and no further checking is needed. The processor resource usage in this scenario includes the usage from both phase 1 and phase 2.
- In the third phase in decision making, the DB2 Optimizer applies costing on the query, and checks whether the estimated total processor and input/output (I/O) cost is greater than five seconds. A query can be offloaded to DB2 Analytics Accelerator if the estimated cost is greater than 5 seconds. In general, the processor resource cost in phase 3 is bigger than in phase 1 and phase 2.

Lab measurement showed less than 10% processor resource cost for query acceleration optimization in an in-house, complex online transaction processing (OLTP) workload that has many static packages. The workload contains 46 packages (static SQL PL). When acceleration is enabled, the queries are either too short to offload, or can't be offloaded because of UDF, or due to the query not being read-only. None of the queries is offloaded.

In general, a complex query has bigger bind resource costs than a simple query, but query speedup and System z processor usage reduction from complex query acceleration more than compensate for the bind resource cost. Bind resource cost for a small query is a one-time cost. No performance effect exists in the query execution on System z, *with* or *without* the query acceleration bind option.

Static query acceleration scenario # 1

In this scenario, queries have good selectivity, and tables on DB2 Analytics Accelerator have distribution and organization keys.

The first data warehousing workload has 22 static queries. Most queries have good selectivity. Figure 5-4 on page 107 shows the overall speedup in query elapsed time and the reduction in System z CPU time for these static queries bound with QUERYACCELERATION(ENABLE) versus QUERYACCELERATION(NONE).

With the QUERYACCELERATION(NONE) bind option, all of the queries were executed on System z with six CPUs and query parallelism enabled. There was no extensive query performance tuning done for these queries that ran on System z. With the QUERYACCELERATION(ENABLE) bind option, all of the queries were accelerated on PureData for Analytics N2001-010.

Because most queries have good selectivity on searched columns and joined columns, tables on DB2 Analytics Accelerator have distribution keys and organization keys to improve query performance. The overall query elapsed time speedup on DB2 Analytics Accelerator was 27 times, and the overall System z CPU usage reduction was 470 times.

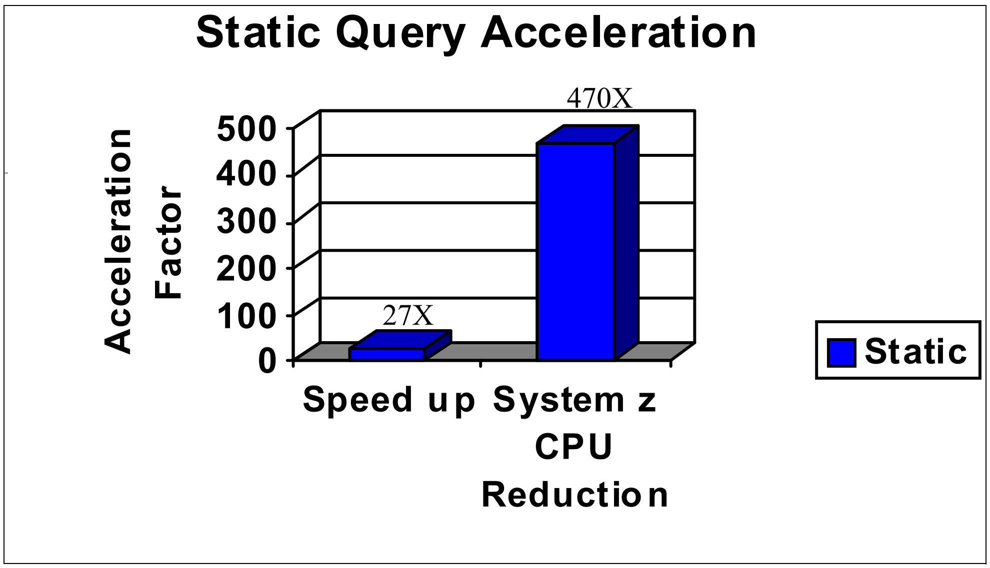


Figure 5-4 Query workload 1 elapsed time and CPU reduction

22 static queries in a Tivoli Storage Productivity Center for Replication data warehousing workload in the Lab have average 27 times query speed up and average 470 times DB2 for z/OS CPU usage reduction from query acceleration.

Queries that ran on DB2 for z/OS were not well tuned. Tables on DB2 Analytics Accelerator have distribution keys.

Static query acceleration achieves similar performance as dynamic query acceleration.

The static packages that do not use acceleration had no noticeable performance effect.

Static query acceleration scenario # 2

In this scenario, the queries are more complex, and tables on DB2 Analytics Accelerator have distribution and organization keys.

The second set of 12 static queries in the Tivoli Storage Productivity Center for Replication data warehousing workload are queries that ran longer than three minutes on System z. Most queries have good or fair selectivity. Figure 5-5 on page 108 shows the overall speedup in query elapsed time, and the reduction in System z CPU time for these static queries bound with QUERYACCELERATION(ENABLE) versus QUERYACCELERATION(NONE).

With the QUERYACCELERATION(NONE) bind option, all queries were executed on System z with six CPUs and query parallelism enabled. There was no extensive query performance tuning done for these queries that ran on System z.

With the QUERYACCELERATION(ENABLE) bind option, all queries were accelerated on PureData for Analytics N2001-010. Because almost all of the queries have good selectivity on searched columns and joined columns, tables on DB2 Analytics Accelerator have distribution keys and organization keys to improve query performance. The overall query elapsed time speedup on DB2 Analytics Accelerator was 48 times, and the overall System z CPU usage reduction was 1221 times, as shown in Figure 5-5.

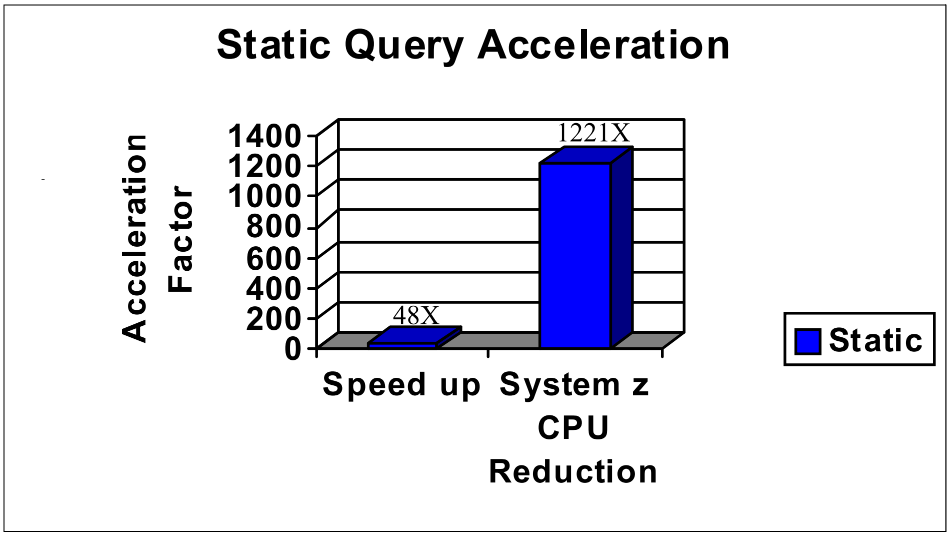


Figure 5-5 Query workload 2 elapsed time and CPU reduction

Queries run on DB2 for z/OS were not well tuned. Static query acceleration achieves similar performance as dynamic query acceleration. The static packages that do not use acceleration had no noticeable performance effect. Query elapsed time speedup and System z CPU reduction for each of the 12 static queries are shown in Figure 5-6 and Figure 5-7 on page 109.

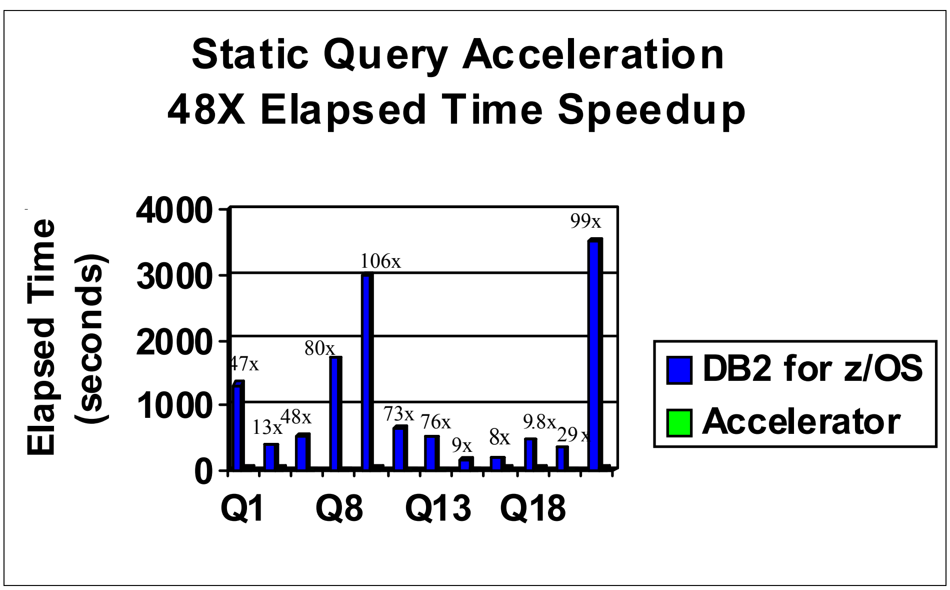


Figure 5-6 Query workload 2 elapsed time speedup

The 12 long-running static queries in a Tivoli Storage Productivity Center for Replication data warehousing workload in that Lab have average 48 times query elapsed time speedup from query acceleration. Elapsed time speedup varies 8 times - 106 times for these static queries.

Figure 5-7 shows the CPU usage reduction.

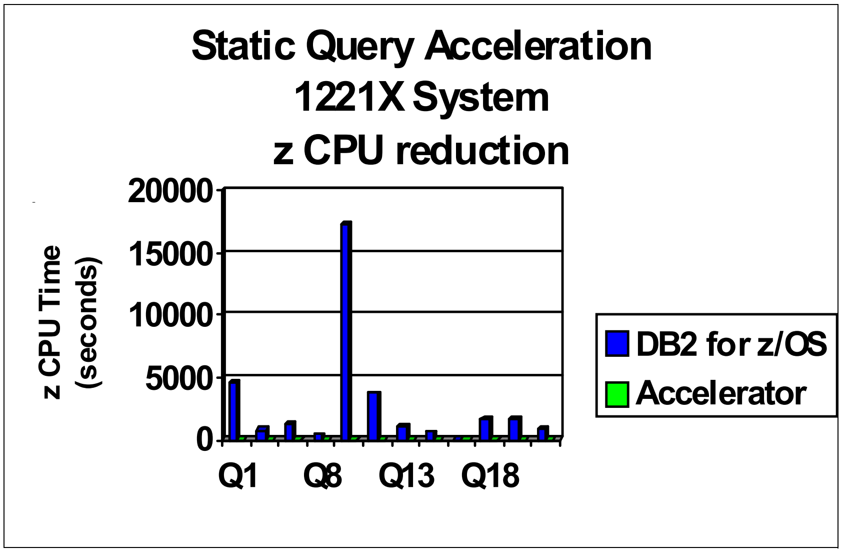


Figure 5-7 Individual query execution

The 12 long-running static queries in a data warehousing workload have on average 1221x CPU usage reduction on System z from query acceleration. Queries run on System z were not well tuned.

Typical System z CPU time for an accelerated query is in milliseconds (ms), unless the query has a large result set.

5.6.5 Static query acceleration scenario # 3

In this scenario, queries have poor or no selectivity, and tables on DB2 Analytics Accelerator have random distribution keys.

The average improvement for the third set of nine static queries in a Cognos data warehousing workload are shown in Figure 5-8 on page 110. Most queries have poor or no selectivity. The overall speedup in query elapsed time and the reduction in System z CPU time for these static queries bound with QUERYACCELERATION(ENABLE) versus QUERYACCELERATION(NONE) are shown in the chart.

With the QUERYACCELERATION(NONE) bind option, all of the queries were executed on z/OS with four CPUs and query parallelism enabled. There was no extensive query performance tuning done for these queries when they ran on System z.

With the QUERYACCELERATION(ENABLE) bind option, all queries were accelerated on PureData for Analytics N2001-010. Because almost all of the queries have poor or no selectivity on searched columns and joined columns, or they scan all of the rows in the tables, tables on DB2 Analytics Accelerator have random distribution with no distribution key. The overall query elapsed time speedup on DB2 Analytics Accelerator was 146 times, and the overall System z CPU usage reduction was 459,000 times.

Figure 5-8 shows the average improvement for static queries in a Cognos data warehousing workload.

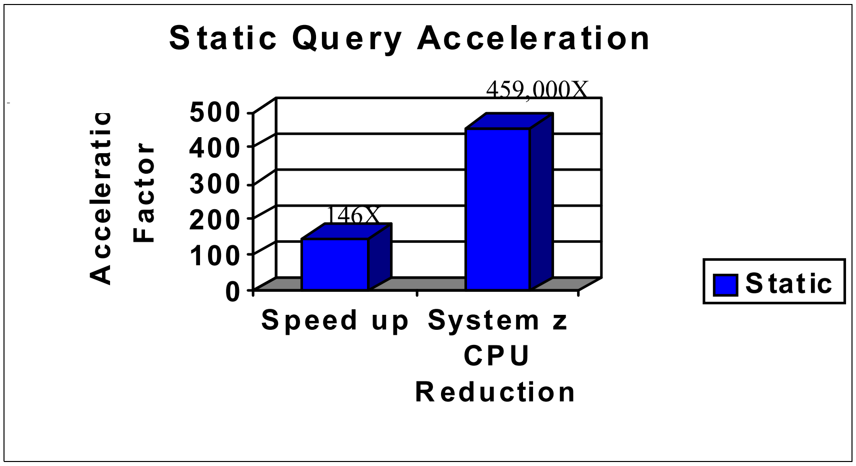


Figure 5-8 Query workload 3 elapsed time and CPU reduction

Static query acceleration achieves similar performance as dynamic query acceleration. The static packages that do not use acceleration had no noticeable performance effect.

Figure 5-9 shows the elapsed time speedup by query.

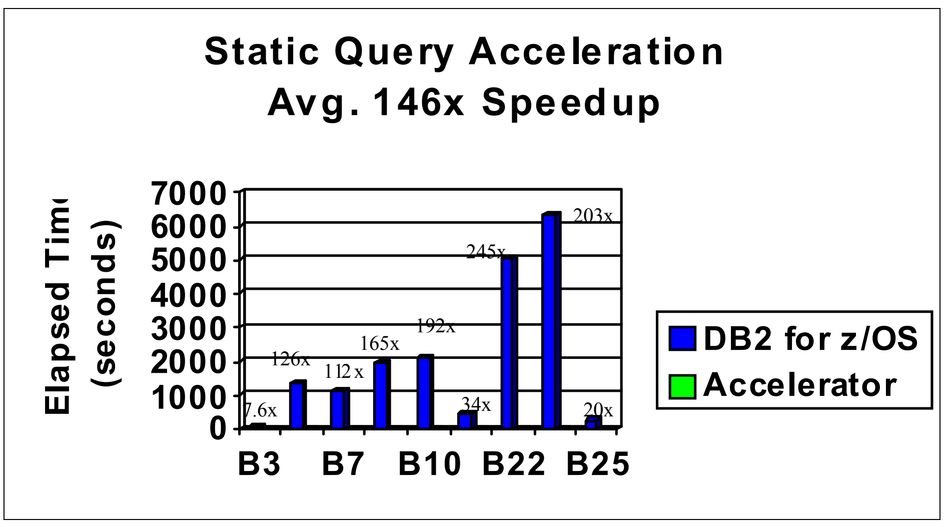


Figure 5-9 Individual query execution

Nine long-running static queries in a Cognos warehousing workload in the Lab have on average 146 times query elapsed time speedup from query acceleration. Queries run on PureData for Analytics N2001 were not tuned.

Elapsed time speedup varied from 7.6 times to 245 times for this set of static queries.

5.7 Large result sets and the cost of returned rows

DB2 11 reduces CPU cost in receiving DB2 Analytics Accelerator returned rows. DB2 11 has several enhancements in DDF and run time to reduce the number of data moves and code paths. Our workload shows an average of 8.4% DB2 CPU reduction in accelerated queries with large result sets.

The table in Figure 5-10 shows the results for a lab experiment of per-row DB2 CPU cost of receiving data from DB2 Analytics Accelerator. The numbers are based on a zEC12 machine with six CPUs. The numbers here are for reference. Different workloads on different DB2 environments might have different results.

Q1, Q2, Q3, Q4, and Q5 are all DB2 Analytics Accelerator accelerated queries with the same two tables joined. The difference is the type and number of selected columns:

- Q1 selects 5 date columns with a record length of 20 bytes.
- Q2 selects 5 string type columns with a record length of 182 bytes.
- Q3 selects 5 integer columns with a record length of 20 bytes.
- Q4 selects 12 columns of date, string, and integer types with a record length of 141 bytes.
- Q5 selects 25 columns of date, string, and integer types with a record length of 275 bytes.

QUERY	# ROWS	SELECT LIST			DB2 CPU second	DB2 CPU ms/row	Avg DB2 CPU ms/row
		DATA TYPE	#COLS	LENGTH			
Q1	10,000	DATE	5	20	0.031	3.150	
Q1	100,000	DATE	5	20	0.276	2.757	
Q1	1,000,000	DATE	5	20	2.734	2.734	
Q1	10,000,000	DATE	5	20	27.576	2.758	2.850
Q2	10,000	CHAR/VCHAR	5	182	0.030	2.952	
Q2	100,000	CHAR/VCHAR	5	182	0.286	2.856	
Q2	1,000,000	CHAR/VCHAR	5	182	2.847	2.847	
Q2	10,000,000	CHAR/VCHAR	5	182	28.203	2.820	2.869
Q3	10,000	INT	5	20	0.029	2.922	
Q3	100,000	INT	5	20	0.278	2.782	
Q3	1,000,000	INT	5	20	2.751	2.751	
Q3	10,000,000	INT	5	20	27.606	2.761	2.804
Q4	10,000	MIXED	12	141	0.034	3.380	
Q4	100,000	MIXED	12	141	0.314	3.137	
Q4	1,000,000	MIXED	12	141	3.150	3.150	
Q4	10,000,000	MIXED	12	141	31.486	3.149	3.204
Q5	10,000	MIXED	25	275	0.037	3.686	
Q5	100,000	MIXED	25	275	0.357	3.565	
Q5	1,000,000	MIXED	25	275	3.579	3.579	
Q5	10,000,000	MIXED	25	275	35.795	3.579	3.602

Figure 5-10 CPU cost per row

A fetch for *n* rows clause is used to control the number of returned rows, where *n* is 10,000, 100,000, 1,000,000, and 10,000,000 in the test.

The five queries have DB2 CPU time nearly proportionally aligned to the number of returned rows. The average DB2 CPU per-row cost is similar in Q1, Q2, and Q3 despite different selected data types and record length.

When the number of selected columns increases in Q4 and Q5, the average DB2 CPU per-row cost increases as well, but not proportionally.

6

Load and incremental update

This chapter explores scenarios for data ingestion behavior for moving data from IBM DB2 to the IBM DB2 Analytics Accelerator for z/OS (DB2 Analytics Accelerator), with a focus on changes that come with DB2 Analytics Accelerator V4.1. Data ingestion, whether batch loading or change data capture, can be resource-intensive.

Even though DB2 Analytics Accelerator is classified as an appliance, it is still a computing device, therefore has limited resources. Understanding how and where data ingestion techniques use resources is key to developing a data ingestion strategy.

The *Hybrid Analytics Solution using IBM DB2 Analytics Accelerator for z/OS V3.1*, SG24-8151 IBM Redbooks publication explored some basic load and incremental scenarios, to understand basic behavior of the batch load, and of incremental update. In this chapter, we explore changes in V4.1 to see what effect they have on data ingestion, and we also explore more advanced scenarios to gain a deeper understanding of batch loading and incremental update.

Note: This chapter describes the data ingestion functions included in DB2 Analytics Accelerator. The separately purchased tool, IBM DB2 Analytics Accelerator Loader for z/OS V1.1, is discussed in Chapter 10, “DB2 Analytics Accelerator Loader” on page 213.

This chapter contains the following:

- Batch load scenarios
- Incremental update changes
- Performance factors affecting incremental update
- Load versus incremental update

In this chapter, the terms *incremental update* and *replication* are used interchangeably.

6.1 Batch load scenarios

In DB2 Analytics Accelerator V4.1, there is no significant change for the batch loader function. In this section, we explore some additional scenarios for the batch load.

Before delving into the scenarios, a quick review might be helpful. For more detail, see the chapters about data latency management and incremental update in *Hybrid Analytics Solution using IBM DB2 Analytics Accelerator for z/OS V3.1*, SG24-8151.

The batch load populates data from DB2 tables to the associated accelerated table in DB2 Analytics Accelerator. It does this by using the concept of *load streams*. A load stream is created to operate on a single non-partitioned table or a single partition of a partitioned table. One load stream unloads data from one DB2 table partition to a pipe, streams it to DB2 Analytics Accelerator, and stream loads it into the corresponding DB2 Analytics Accelerator table.

If a table is a partitioned table, a load stream is created for each partition, in parallel, up to a maximum number defined in the AQT_MAX_UNLOAD_IN_PARALLEL environment variable found in AQTSAMP(AQTENV). If it set to four, for example, for any partitioned table the load function creates up to four concurrent load streams. If there are more than four partitions, when a load stream completes the loading of a partition, another partition is loaded.

The load is started using the SYSPROC.ACCEL_LOAD_TABLES stored procedure. This is done implicitly when you perform a load from DB2 Analytics Accelerator Studio, but it can also be called explicitly in a batch job or Restructured Extended Executor (REXX) script, for example. The next sections provide information about the following topics:

- Factors affecting DB2 Analytics Accelerator load throughput
- Concurrent table load behavior

6.1.1 Factors affecting DB2 Analytics Accelerator load throughput

This section provides an answer to questions such as, “Why does it take 15 minutes to load my 28 tables in serial, but it takes 3 hours when I load them in parallel using multiple jobs?”

First, we need to understand what resources are used when doing a load from DB2 to DB2 Analytics Accelerator. Resources are used on both the IBM System z side and on the DB2 Analytics Accelerator side. Having a resource constraint on either side limits the overall throughput that can be achieved. Currently, the network is typically not a limiting factor.

The chart in Figure 6-1 on page 115 lists many of the factors that affect the throughput of loading data to DB2 Analytics Accelerator. The three highlighted factors are the primary influencers:

- The number and size of partitions and tables
- The number and speed of processors
- IBM PureData for Analytics (formerly IBM Netezza) host

This section provides information about these factors. In this section, we use the term *partition* to refer to either *a single segmented table space* or *a single partition of a partitioned table space*. Essentially, they are equivalent in terms of their effect on overall DB2 Analytics Accelerator throughput.

Loading a single partition results in one load stream being established. The following steps then occur:

1. A call is made to the DSNUTILU stored procedure to start a DB2 unload utility to unload the data, in internal DB2 format, from DB2 to a pipe. Of course, DB2 a stored procedure address space must be established for the unload utility to run.
2. The pipe is established to receive data from the DB2 unload utility and to stream it across the private network to DB2 Analytics Accelerator.
3. Processes on the DB2 Analytics Accelerator host node receive the data stream from the network, and convert the data from internal DB2 format to a format appropriate for DB2 Analytics Accelerator database loader.
4. Host nodes stream the data to DB2 Analytics Accelerator loader, where it is inserted into the corresponding Accelerator table.

Steps 1 and 2 use resources on the System z side, but step 3 use resources on the DB2 Analytics Accelerator host node. Step 4 primarily affects DB2 Analytics Accelerator worker nodes, and is not usually a performance issue for a DB2 Analytics Accelerator load.

Accelerator Load Throughput Factors

Factors	System z Server	Accelerator
Data Characteristics	o Table size & number of partitions o # of concurrent load streams o Partitioning data distribution o RECLENGTH	o Table size & number of partitions o # of concurrent load streams o Data Types (ex. Float) o Order of DB2 input data as compared to defined Accelerator Distribution key
Resources Available	o Number of CPs, Model (MIPS) o Memory o # OSA cards / ports? o # 10 Gbit fibers to accelerator	o PureData System for Analytics Model o Host Server o SPUs o Disk o Number of Host Server ports utilized o Active / Active
Controls	o AQT_MAX_UNLOAD, NUMTCB o zWLM Policy o MNSPAS o TCP/IP Stack settings o MTU (Jumbo Frames)	o AQT_MAX_UNLOAD, NUMTCB o Data Maintenance priority (Console) o MTU (Jumbo Frames)
Recent Enhancements	o Unload Light (V31 PTF 1) o Load latency (V31 PTF 3) o DFSMS APAR OA41706 o DB2 APAR PM90151 o z/OS APAR OA43538, OA43068	Accelerator V41 PTF 3

Simply stated, it is a balancing act

Number of concurrent streams invoked on System z | ***Accelerator host processor load***

Figure 6-1 Factors influencing Accelerator load throughput

If a partitioned table is loaded, a separate concurrent load stream is established for each partition up to the number specified in the AQT_MAX_UNLOAD_IN_PARALLEL DB2 Analytics Accelerator environment variable. The default value is four (4).

If a table has more than four partitions, there are only four concurrent load streams at any one time for this table. Therefore, there are four concurrent DB2 unload utilities, four sets of pipes sending data to DB2 Analytics Accelerator, four processes on the DB2 Analytics Accelerator host for preparing the data, and four concurrent DB2 Analytics Accelerator loads.

Note: It is extremely important to understand that the environment variable, AQT_MAX_UNLOAD_IN_PARALLEL, applies only to partitioned tables, and applies to each partitioned table concurrently loading. Therefore, if you have two concurrent calls to the ACCEL_LOAD_TABLE stored procedure, and therefore have two partitioned tables loading, you can have two times the value of AQT_MAX_UNLOAD_IN_PARALLEL concurrent load streams. This would result in eight concurrent load streams using the default value of four.

For a detailed description of load streams, see *Hybrid Analytics Solution using IBM DB2 Analytics Accelerator for z/OS V3.1*, SG24-8151.

To increase the overall throughput, you want to increase the number of concurrent load streams. As the number of concurrent load streams increases, the more resources are used on both the System z and DB2 Analytics Accelerator host node. At some point, available resources on one side or the other become exhausted.

We have observed that, for IBM zEnterprise EC12 (zEC12), 3 - 5 load streams can be supported per central processing unit (CPU). If the System z is constrained, the number of concurrent load streams is limited by the available System z resources. If there are adequate System z resources, DB2 Analytics Accelerator, particularly the host node, becomes the limiting factor for the maximum number of concurrent load streams that can be supported.

The number of concurrent load streams that the DB2 Analytics Accelerator host can support is dependent on the number and speed of DB2 Analytics Accelerator host node CPU cores, which varies by DB2 Analytics Accelerator model. Other DB2 Analytics Accelerator workloads, queries, and incremental updates will also use resources with which the load will have to compete.

We observed that a PureData for Analytics model N1001-101 can support 8 - 10 concurrent load streams, and a PureData for Analytics N2001-010 can support 15 - 18 concurrent load streams. Half-rack models typically have slower host nodes, and multi-rack systems typically have faster host nodes. PureData for Analytics N2002 has newer host nodes but was not available in our test bed at this time.

Therefore, as shown in Figure 6-1 on page 115, this becomes a balancing act between the number of concurrent streams that can be supported on the System z side versus the number of concurrent streams that can be supported on DB2 Analytics Accelerator host.

Now to answer the original question. We really need more information to properly answer this question. We need to know how many tables are partitioned. We also need to know the value of the AQT_MAX_UNLOAD_IN_PARALLEL environmental variable.

In the best case, assume that all 28 tables are single-partitioned tables. This results in 28 load streams running concurrently. We need enough available resources on the System z side to support 28 concurrent DB2 unload utilities.

We know from observations that the maximum number of parallel streams that we can support on DB2 Analytics Accelerator without throughput degradation is in the range of 16 - 20 on a PureData for Analytics N2001-010 full rack system. Even in this case, we have over-saturated DB2 Analytics Accelerator with load streams. Assuming that we can support five concurrent load streams per CPU on the System z side, we would need nearly six available CPUs to support this load scenario.

What happens if all 28 tables were partitioned with at least four partitions each? Using the default value for AQT_MAX_UNLOAD_IN_PARALLEL of 4, this would result in 28 x 4 or 112 concurrent load streams. This is certainly far more streams than can be supported on DB2 Analytics Accelerator. On the System z side, at 5 concurrents loads per CPU, this scenario would require at least 22 available CPUs to support this level of concurrency.

Although we can't absolutely answer this question without more information, this points out that we have to be careful to design a batch load strategy that considers the available resources on the System z side, and doesn't over-saturate DB2 Analytics Accelerator.

6.1.2 Concurrent table load behavior

In *Hybrid Analytics Solution using IBM DB2 Analytics Accelerator for z/OS V3.1*, SG24-8151, we focused primarily on the behavior of single-partition table loads. In this section, we look at two other scenarios focused on loading of multiple tables.

Multiple single-partition tables versus single table with multiple partitions

The question is, "Is there any difference between loading a certain amount of data from multiple single-partition tables versus loading the same data from a single table with multiple partitions?" We explore this scenario to answer the question.

We start with a partitioned table containing 1 billion rows across 100 partitions. From that table, we create 20 single-partition tables, each having 50 million rows. The partitioned table is loaded using a AQT_MAX_UNLOAD_IN_PARALLEL value of 20. All 20 single-partition tables are loaded concurrently, with separate invocations of the ACCEL_LOAD_TABLE stored procedure. We are interested in comparing the overall throughput and the System z CPU usage cost.

Referring to Figure 6-2 on page 118, we see little difference between the two runs. We see that the throughput for the single multi-partitioned table is lower than for the 20 single table loads. Why might that be? Because there is a difference in the total number of partitions. The single table loads a total of 100 partitions, which means 100 load streams had to be instantiated. The multi-table example only had to instantiate 20 load streams.

The resource cost of creating 80 additional load streams caused the overall throughput to decrease. If the single table had 20 partitions, the throughput would have been much closer.

Similarly, for the System z CPU usage, the additional resource cost for creating the extra 80 load streams caused a higher CPU cost. Again, if the single table had 20 partitions, the cost would be closer.

Figure 6-2 shows the difference between the two test runs.

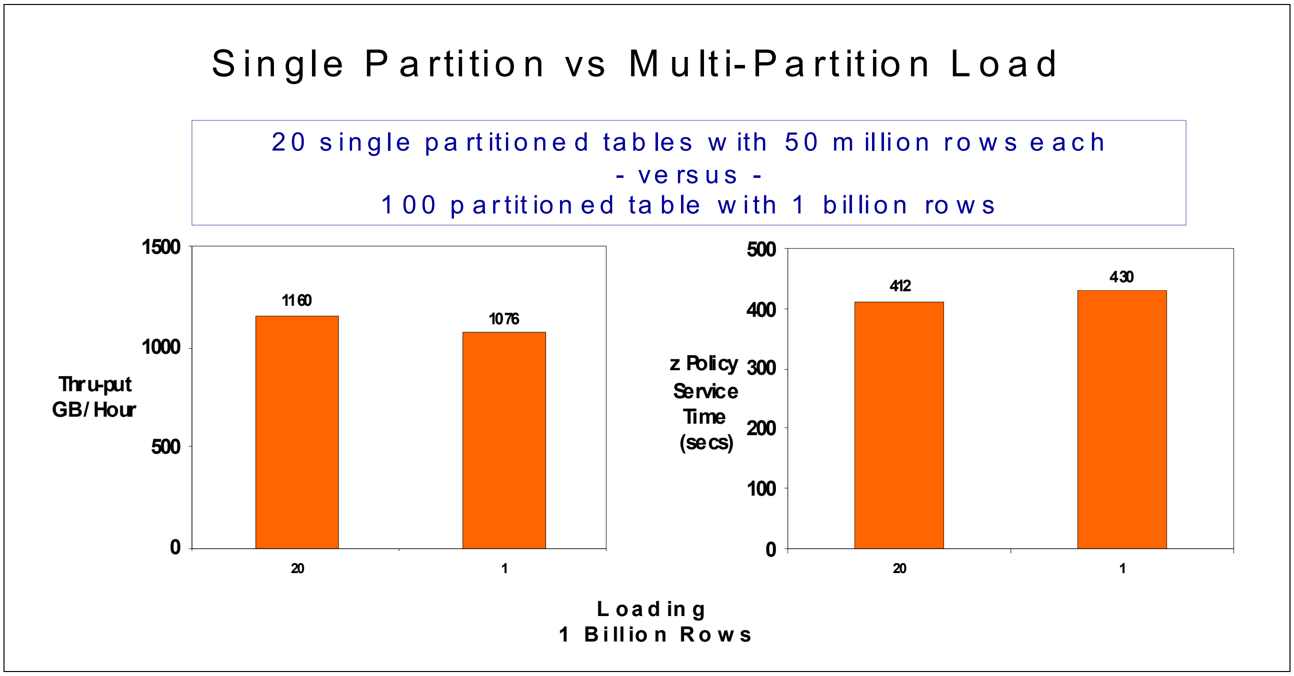


Figure 6-2 Single partition versus multi-partition load.

From this, we conclude that the important factor is the *number of load streams* created. It doesn't matter if those load streams are caused by a single table having multiple partitions or multiple tables running concurrently.

Concurrent table loads

Another experiment involving single partitions is to see what kind of concurrency would give us a peak throughput. Using tables with 50 million rows, we increase the number of concurrent table loads to see where we reach a peak throughput, and then continue to increase that number to determine how throughput changes. Note that the System z resources are not constrained, and there is no other work on DB2 Analytics Accelerator.

Figure 6-3 on page 119 shows the throughput, the System z CPU time, and DB2 Analytics Accelerator host usage across varying levels of load concurrency. The x-axis shows the level of concurrency used (1 - 40). The throughput line shows the increase in throughput from concurrency of 1, reaching a peak of 1160 gigabytes (GB) per hour at *concurrency of 20*.

At this point, DB2 Analytics Accelerator host usage is 93% - 95%. As we increase the concurrency above 20, we see a reduction in throughput and a decrease in the usage of the DB2 Analytics Accelerator host. Both slowly decline as the concurrency increases. This is likely due to the additional resource cost of managing the load streams.

Notice that the System z cost is primarily dependent on the amount of data. As the number of tables loaded increases, the System z CPU cost increases. The CPU time per load stream is consistent, 20 - 23 seconds per table.

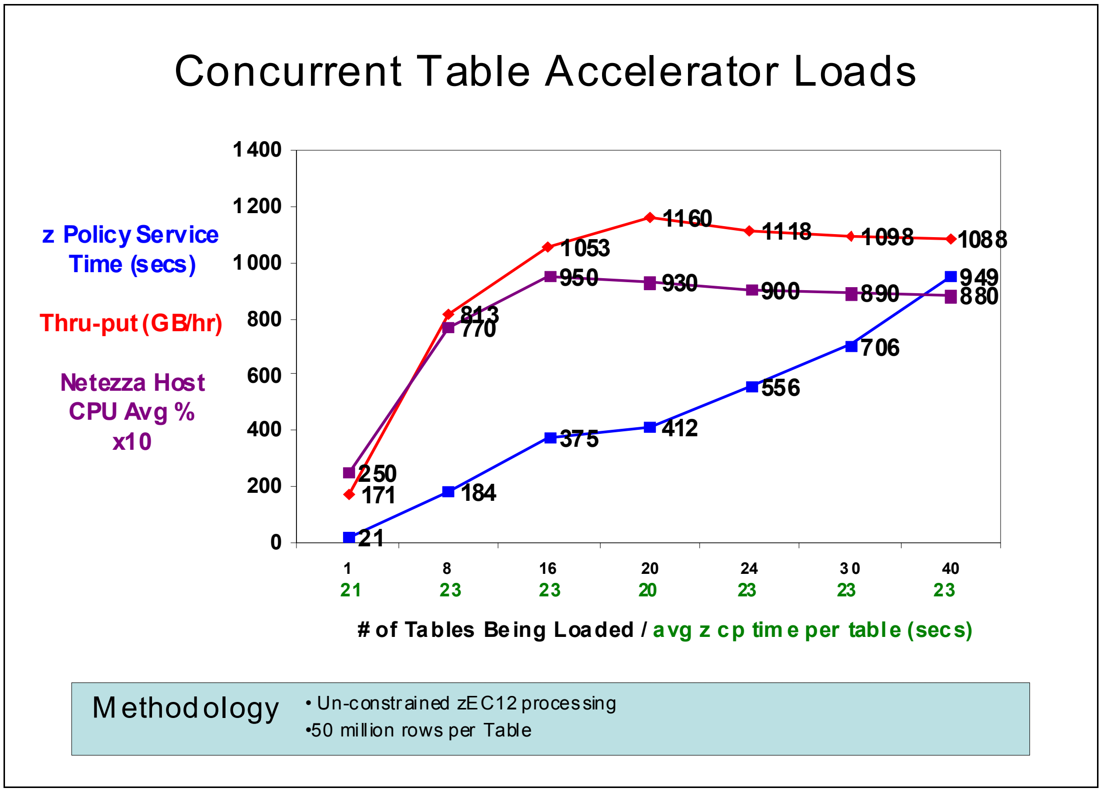


Figure 6-3 Concurrent single partitioned table loads.

6.2 Incremental update changes

IBM DB2 Analytics Accelerator V4.1 brings a significant change to the architecture of the incremental update (IU) apply function. This new architecture expands the number of source DB2 subsystems that can be supported on the target accelerator. Also, there are changes in DB2 Analytics Accelerator to how accelerator resources are managed for the incremental update apply function. On the System z source side, CPU use for the capture agent has been reduced for processing out of scope changes.

This section provides information about the following topics:

- Architectural changes
- Incremental update resource allocation
- Reduced System z CPU usage for incremental update

6.2.1 Architectural changes

Before DB2 Analytics Accelerator V4.1, one accelerator was limited to having two DB2 subsystems configured for incremental update. For each connected subscription, one instance of the apply agent was created for each connected DB2 subsystem, with each apply agent using 4 GB of memory. On the DB2 Analytics Accelerator host node, there was only 8 GB of free memory that could be allocated for incremental update, producing the limitation of two DB2 subsystems.

With DB2 Analytics Accelerator V4.1, there is only one apply agent created, as shown in Figure 6-4. This apply agent can use the entire 8 GB of free memory, and can support up to 10 DB2 subsystems configured for incremental update. This one agent processes the changes for all of the tables being replicated to this accelerator.

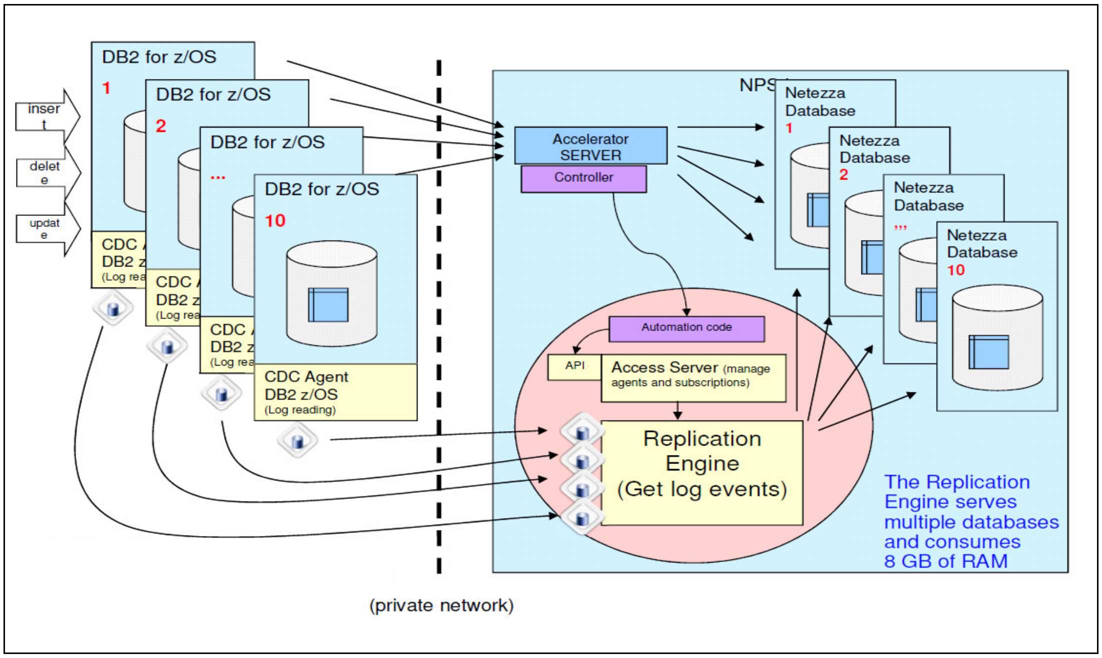


Figure 6-4 The new incremental update architecture supports up to 10 DB2 subsystems

6.2.2 Incremental update resource allocation

To understand how DB2 Analytics Accelerator V4.1 changes the resource allocation related to incremental updates, we provide a high-level view of how resources are allocated in DB2 Analytics Accelerator. For a more detailed description, see the chapter about resource allocation in *Hybrid Analytics Solution using IBM DB2 Analytics Accelerator for z/OS V3.1*, SG24-8151.

In DB2 Analytics Accelerator, the basis of workload management is the resource group. One resource group is created for each DB2 subsystem that is paired with DB2 Analytics Accelerator. Each resource group has a configurable minimum Guaranteed Resource Allocation. Therefore, no matter what else is happening in the system, this resource group is assured a certain amount of resources.

In one resource group, work coming from one DB2 subsystem is managed using priorities. There are four priorities: Low, Normal, High, and Critical. Of course, many queries running at higher priorities can reduce resources available to queries running at lower priorities.

Resource allocation values are defined using the DB2 Analytics Accelerator configuration console. An entry for each DB2 subsystem is paired with DB2 Analytics Accelerator. The minimum resource value is an absolute number in the range of 1 - 100. These values do not have to add up to 100. They are used in a calculation to determine a relative resource allocation percentage for each resource group.

In example 1 of Figure 6-5, the defined values are both 100, which results in a relative percentage of 50% for each DB2 subsystem's resource group. The work coming into each resource group is assured a minimum of 50% of the resources. Of course, if one DB2 subsystem is not using its allocated 50%, it is available for use for work from the other DB2 subsystem.

Modifying the minimum resources value changes the relative resource allocation percentage, as shown in example 2 of Figure 6-5. In this example, DBZZDDF is assured to get about two thirds of the resources at a minimum.

Index	Location Name	Minimum Resources	Relative Percentage
1	DBZZDDF	100	50.00
2	DZA1DDF	100	50.00

Resource allocation **example 1**

Index	Location Name	Minimum Resources	Relative Percentage
1	DBZZDDF	100	66.66
2	DZA1DDF	50	33.33

Resource allocation **example 2**

Figure 6-5 Resource allocation examples

Why is all of this important to incremental update? Well, applying changes is work, and that work has to be managed. The work must be assigned to some resource group and assigned a priority. Before DB2 Analytics Accelerator V4.1, all incremental update work for a DB2 subsystem was assigned to the same resource group associated with the DB2 subsystem, with a priority of *normal*.

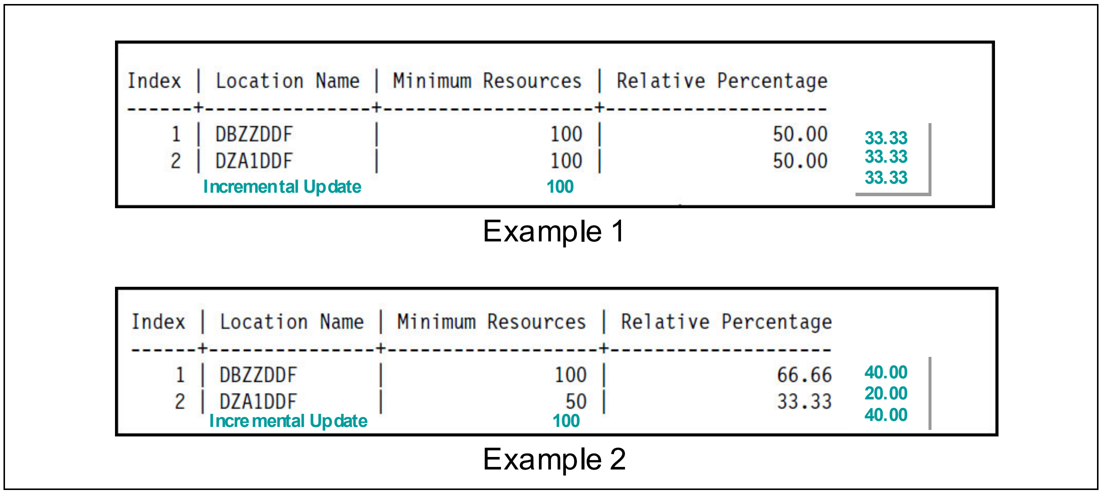
That means that IU work competed directly with queries for the amount of resources allocated to that resource group. Just as high and critical queries can keep lower-priority queries from getting resources, they can also keep resources from incremental update work.

Starting with DB2 Analytics Accelerator V4.1, as part of the architectural changes, there is now only one instance of the replication engine. A new independent resource group has been created for replication work. It has a minimum resource allocation value of 100, and all work is set to normal priority.

However, this resource group definition is not externalized in the configuration console, and therefore can't be seen or modified. It is important to understand that it exists, and that it affects the calculated relative resource percentages for resource groups.

Figure 6-6 shows the two earlier examples with incremental update considered. Notice that in Example 1, the relative minimum resource allocation changes to 33% for each of the resource groups. The relative resource allocation also changes accordingly in Example 2.

> **Tip:** The relative resource allocation for incremental update can be reduced by pairing additional DB2 subsystems to DB2 Analytics Accelerator, therefore creating additional resource groups. The relative resource allocation for incremental update can be increased by decreasing the resource values for the other resource groups.



Index	Location Name	Minimum Resources	Relative Percentage	
1	DBZZDDF	100	50.00	33.33
2	DZA1DDF	100	50.00	33.33
	Incremental Update	100		33.33

Example 1

Index	Location Name	Minimum Resources	Relative Percentage	
1	DBZZDDF	100	66.66	40.00
2	DZA1DDF	50	33.33	20.00
	Incremental Update	100		40.00

Example 2

Figure 6-6 The new incremental update resource group affects relative resource allocation

The result of this change is that incremental update is assured to get a minimum amount of resources, even in times of high resource contention. Figure 6-7 on page 123 shows four incremental update scenarios. Each graph is a timeline of incremental update activity, with a query workload starting around the middle of the timeline. We see how it accepts the replication workload.

In DB2 Analytics Accelerator V3.1, the query workload clearly has a detrimental effect on the replication activity, even to the point that critical queries did not permit the replication activity to obtain any resources. This is due to the queries and incremental update work being assigned to the same DB2 Analytics Accelerator resource group. Replication work is assigned a priority of normal, so queries with priorities of high and critical get the resources first.

With DB2 Analytics Accelerator V4.1, replication work has a separate Accelerator resource group independent of any query workloads. It gets a minimum amount of resources based on the calculated relative resource allocations, which results in replication work continuing even during times of high resource contention.

In the Version 4 charts, replication is able to continue to run even during the heavy query stream. In fact, the priority of the queries is no longer relevant to the resources for replication. Only the relative resource allocation for the incremental update resource group, compared to those of the other resource groups, affects the resource allocated to incremental update.

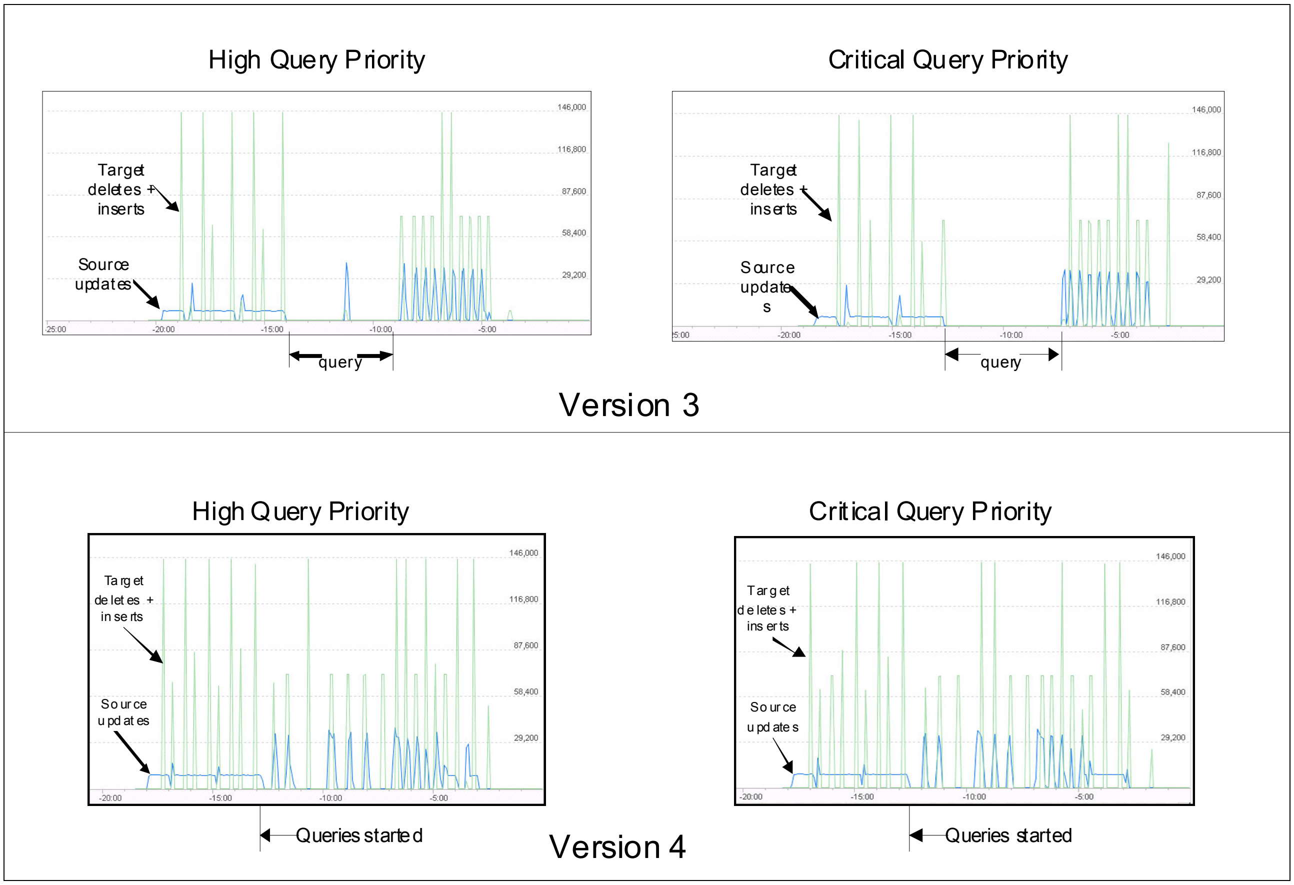


Figure 6-7 Accelerator resource management improved for incremental update

Of course, there is a cost for this. In times of heavy contention, if one workload gets more resources, another workload gets less. We observed an increase in the elapsed time for those queries running concurrently with incremental updates, as we see in seconds in Figure 6-8. Because incremental update now has a minimum ensured amount of the resources, there are fewer resources for processing the queries when there is high contention for those resources. Consequently, the query workload experienced longer elapsed times.

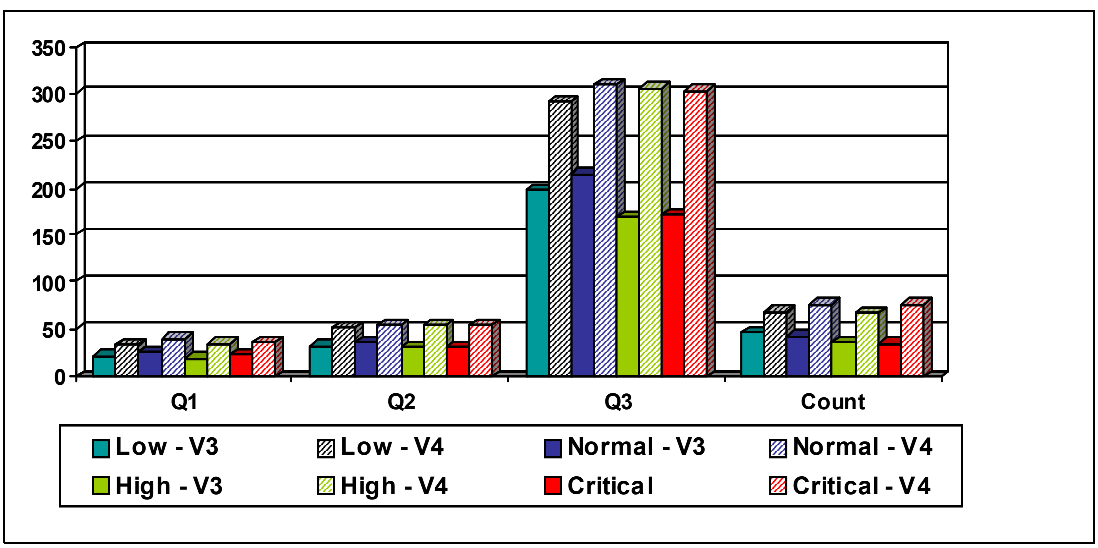


Figure 6-8 Effect on query times of concurrent incremental updates in DB2 Analytics Accelerator V4.1

6.2.3 Reduced System z CPU usage for incremental update

Some incremental update users have experienced higher than expected CPU usage for DB2 Analytics Accelerator capture agent running on System z. The CPU usage seems to be out of line with the amount of perceived replication activity.

The DB2 Instrumentation Facility Interface (IFI) is what the apply agent uses to gain access to change records from the DB2 log. Those records are passed back to the capture agent from IFI, and the apply then determines if that change applies to one of the tables of interest to this capture agent. These are called *in-scope changes*. These in-scope changes are then processed and sent to the target apply agent in DB2 Analytics Accelerator.

Change records that are not of interest to this capture agent are called *out-of-scope changes* and are discarded. The resources to receive, examine, and discard out-of-scope changes can be significant, depending on the number of out-of-scope records. This resource cost can appear to be very high if there is a high number of out-of-scope changes compared to in-scope changes.

With DB2 Analytics Accelerator V4.1, there is a possibility to reduce the CPU usage for processing out-of-scope changes. In this case, IFI will do the filtering of out-of-scope changes, and send to the capture agent only changes for tables that it needs.

> **Important:** IFI filtering requires IBM DB2 Analytics Accelerator V4.1 and DB2 11, or DB2 10 with DB2 authorized program analysis report (APAR) PM90568 (UK97013) and IBM InfoSphere Change Data Capture APAR PI08186 (program temporary fix (PTF) UI13632).

We ran two experiments where we processed five million updates to DB2 where all five million updates were out-of-scope, and again where all five million updates were all in-scope. We ran these tests before and after the IFI filtering PTFs were applied to DB2 10. For the out-of-scope test, we observed a 28.5% reduction in CPU usage for the capture agent. We also saw a minor drop in the CPU usage of around 6% in the in-scope scenario as well, which might possibly be attributed to normal test variation.

The amount of CPU reduction that you might see is dependent on the number of out-of-scope changes in relation to the number of in-scope changes. If there are many out-of-scope changes, the CPU savings will be higher.

6.3 Performance factors affecting incremental update

There are several factors that affect the performance of any kind of data replication scenario. These include the performance of the capture agent for the source data, the capacity of the network, and the performance of the apply agent for the target data. Incremental update for the IBM DB2 Analytics Accelerator is a specific implementation of a change data capture scenario. The capture agent does just a simple capture and forward of changes.

There are no changes to the data before being forwarded to DB2 Analytics Accelerator. We typically do not see the capture being a major performance bottleneck. The network between System z and DB2 Analytics Accelerator is a dedicated 10 GB per second (GBps) connection, which we have not seen as a typical bottleneck. Therefore, as typical in any data replication solution, it is the target system that is most likely to become the bottleneck. The *target apply system* on DB2 Analytics Accelerator is the focus of this section.

DB2 Analytics Accelerator database is different from most relational databases that are the usual target of replication, which brings some challenges to data replication technology. DB2 Analytics Accelerator is based on Netezza technology. The Netezza system is optimized for extremely quick processing of large analytic-style queries. An additional advantage of this architecture is that there is no need for creating performance-enhancing artifacts, such as indexes, because data is highly distributed and can be scanned quickly.

Having no indexes helps the insert/load process, because there are no indexes to maintain. The data is just appended to the end of the data pages. Of course, as with any design decision, there are some drawbacks. Having no indexes is detrimental to any update or delete activity. Rather than doing an index lookup to find the location of a row, the table must be scanned to locate the rows for the operation. This behavior causes some challenges to replicating deletes and updates to DB2 Analytics Accelerator.

One way that the incremental update function addresses this is by collecting the changes into mini-batches. These mini-batches are typically run at one-minute intervals, to reduce the number of table scans required.

Another way that incremental update uses to reduce the number of table scans is to convert DB2 updates into delete and insert pairs, so that there is only one insert and one delete operation per table per apply cycle. In addition, multiple changes to a row in one apply cycle are consolidated, and only the most recent change is applied.

When the apply cycle is run, for each table, a bulk delete operation is done, followed by a bulk insert operation. This is repeated for each table that has replication operations at the time of apply. After all of the changes have been applied, a commit is done. Performing a commit only at the end of the apply cycle ensures transactional consistency for those cases where changes occur across tables in a single DB2 unit of work.

The elapsed time that it takes for one apply cycle to complete depends on three primary factors:

- The sizes of the target tables
- The number of changes for each table
- The number of tables having change activity

The time that it takes for one apply cycle to complete determines the data latency that can be expected. Therefore, it is important to understand how each of these factors affects overall incremental update performance. In the next sections, we explore each of these factors to understand how variation in each of these factors affects the time to perform apply cycles:

- About the test environment
- Effect of target table size on incremental update
- Effect of change velocity on incremental update
- Effect of number of tables on incremental update
- One final example

6.3.1 About the test environment

The experiments in this section were done using a zEC12 with DB2 11. Furthermore, *DB2 Analytics Accelerator* refers to a PureData for Analytics N2001-010 full rack system using DB2 Analytics Accelerator V4.1 software.

The following tables exist in the test environment:

- 300 small tables of 100,000 rows of about 30 megabytes (MB) compressed in DB2 Analytics Accelerator.
- A medium-sized table of two billion rows that is about 200 GB uncompressed and 73.3 GB compressed in DB2 Analytics Accelerator.
- A large table of nine billion rows that is about one terabyte (TB) uncompressed and 296 GB compressed in DB2 Analytics Accelerator.

All of the tables have a primary key defined.

> **Tip:** It is highly advised that replicated tables have some set of columns that identify the row as unique. This is important in processing deletes, because the key columns are used to qualify the delete. This can help performance, but also can avoid duplicate rows being unintentional deleted.

6.3.2 Effect of target table size on incremental update

At apply time, one delete and one insert operation can be generated for each table, depending on the types of activity taking place in DB2. The elapsed time to apply a set of changes to a table is the sum of the elapsed time to perform the delete operation and the elapsed time to perform the insert operation.

As mentioned earlier, a DB2 update operation is transformed into a delete and insert pair. Decomposing a DB2 update operation into a delete and insert pair means that the apply only has to do delete and insert operations, no matter what kind of change happens in DB2. This results in having to do one scan per table per apply instance.

Because there are no indexes in a DB2 Analytics Accelerator database, the delete operation requires that the data slices of the table be scanned to find the qualifying rows for the operation. During the delete operation, the rows are logically deleted by setting an attribute on the deleted rows.

The row has to be read, the attribute modified, and the row written back out to disk. The rows are physically deleted when a background process called *groom* is run. DB2 Analytics Accelerator does this automatically when the table reaches a certain threshold of disorganization.

The time it takes to scan and logically delete rows depends on the compressed size of the table in DB2 Analytics Accelerator. The total size of the table includes the space that holds logically deleted rows that have not been removed by the groom process, the number of data slices, and the scan speed. Similar to queries, having more and smaller data slices reduces the time to scan the table. The number of data slices and the scan speed varies among DB2 Analytics Accelerator models.

For this test, we perform some number of updates to the tables. We use updates because that will produce 50% deletes and 50% inserts on DB2 Analytics Accelerator. The focus of these experiments is to measure the time that it takes the DB2 Analytics Accelerator database to perform the delete and insert operations done on behalf of incremental update.

Our observations show that inserts are efficient. The inserted rows go to the end of the data slice to which they are assigned. We have not observed any major bottlenecks with inserts. The size of the table and the number of inserts seem to have little effect upon incremental update compared to deletes. In fact, in our experiments, we have observed that the insert operation to one table takes an average of about one second elapsed time.

In a few cases where there are a high number of inserts, over 300,000 inserts to one table in one apply cycle, we've observed that the apply might take two seconds. Therefore, we focus on delete processing in the rest of this section.

For delete processing, the size of the table affects the time that it takes to scan the table. A larger table will require more time to scan. To get a rough approximation of the scan time, divide the size of the table in DB2 Analytics Accelerator by the scan speed of your model of DB2 Analytics Accelerator.

This gives you a rough estimate of the scan time, which is the basis for calculating the elapsed time for processing the delete operation of incremental update. Another component of the delete time is the number of rows to be deleted, which is discussed in 6.3.3, "Effect of change velocity on incremental update" on page 128.

Table 6-1 shows the scan speed and number of data slices for each configuration of the PureData for Analytics N2001. If a certain amount of data is spread across more disks, this would result in smaller data slices. A smaller data slice takes less time to scan. For example, a 200 GB table on a half rack would take roughly 200/16 = 12.5 seconds to scan. That same table on a four-rack system would take 200/128 = 1.6 seconds to scan.

Table 6-1 Scan speed and number of data slices for models of the N2001

PureData for Analytics N2001	Approximate scan speed (GBps)	Number of data slices
Half rack	16	120
Full rack	32	240
Two racks	64	480
Four racks	128	960

Of course, there is more to this scenario, which is the concept of *zone maps*. Zone maps are internal DB2 Analytics Accelerator tables that track the minimum and maximum values for each column, of certain data types, per page and extents. Zone maps can be used to avoid reading pages that don't have data of interest. This happens automatically. If your data happens be somewhat organized based on the columns of the primary key, this might help the delete processing to skip reading some pages.

In addition to the time that it takes to locate the row, there is additional time to read the row, update the delete attribute, and write it back to disk. The more rows to be deleted, the larger this time becomes. We explore this topic in 6.3.3, "Effect of change velocity on incremental update" on page 128. In this section, we use a minimum number of changes.

To illustrate this, we made 100 updates to one of the 30 kilobyte (KB) tables, the 200 GB table, and the 1 TB table. The results are documented in Table 6-2 on page 128. We show the calculate scan time based on a 32 GBps scan rate.

We saw that inserts took one second for all three tables. This makes sense, because inserts are all done at the end of the data slice.

Deletes, however, did increase with the table size. For the small table, the observed time to process the delete was one second and was probably much less than that. This table is just too small to get a detailed measurement.

> **Note:** Our resolution of measurement for the tools we used is one second. We cannot accurately measure any interval smaller than that. However, as seen in 6.3.4, “Effect of number of tables on incremental update” on page 131, one second seems to be the minimum time for incremental update to prepare for and run a Structured Query Language (SQL) statement.

The calculated scan time for the 200 GB table is 2.3 seconds. We would expect, then, that the total delete time would be higher than that when considering the time to update the delete attribute and write it back to disk. However, the delete time is only one second. This indicates that, indeed, our delete benefitted from zone maps.

It is inconclusive as to whether the 1 TB table benefitted from zone maps. The total delete time is just a bit more than the scan time, as shown in Table 6-2.

Table 6-2 Minimum time to process deletes and inserts on behalf of incremental update

Table	Insert time (sec)	Calculated table scan time (sec)	Delete time (sec)
30 KB	1	.0009	1
200 GB	1	2.3	1
1 TB	1	10.4	11

How can we make sure the we get a benefit from zone maps? To get the best use of zone maps, we should define distribution and organization keys on the accelerated table. Both keys should match the primary key. See 2.2, “Zone maps (automatic partitioning)” on page 31, and 4.5, “Tuning distribution and organizing keys” on page 78, for additional information about choosing the distribution and organizing keys. We see the effect on our scenario in 6.3.3, “Effect of change velocity on incremental update”.

To summarize, the amount of time to insert data is not dependent on the table size. The delete time, however, is related to the table size, which is modified by any benefit seen by being able to use zone maps.

6.3.3 Effect of change velocity on incremental update

In 6.3.2, “Effect of target table size on incremental update” on page 126, we see how the table size affects incremental update, particularly delete processing. Another major factor that plays into overall incremental update performance is the number of changes to be applied for a table in any given apply cycle.

In this experiment, we explore how the velocity of changes affects the performance of the incremental update apply function. We use the two larger tables, 200 GB and 1 TB, to vary the velocity in terms of number of DB2 updates per second. Because deletes are the primary component in apply operations, we focus on how delete processing varies as the velocity of change increases.

Figure 6-9 shows the time to process deletes for the 200 GB and 1 TB tables across increasing numbers of rows to be deleted during one apply cycle. Given that inserts take an average of one second, we can clearly see that the delete process is the major driver in determining the amount of time to apply changes to a table.

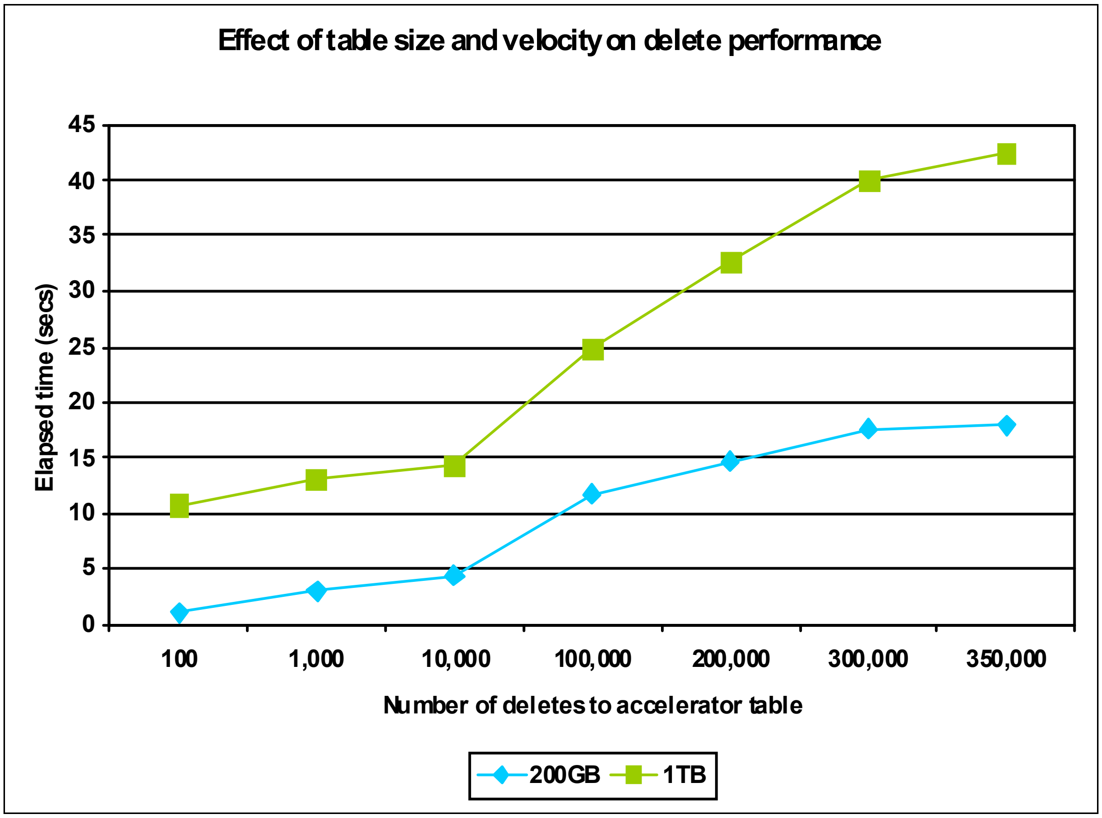


Figure 6-9 Effect of change velocity on apply time

There are some things we might be able to do to improve the delete performance against large tables.

We saw that we might get some benefits of zone maps if the data is somewhat ordered by the primary key. We can define distribution and organization keys, on DB2 Analytics Accelerator tables, that are the same as the DB2 primary key. This will distribute and organize the data so that zone maps might be much more effective. See 4.5, "Tuning distribution and organizing keys" on page 78 for a more detailed description.

Applying this to our 200 GB and 1 TB table and repeating the experiment, we see the results in Figure 6-10. For both tables, there was decrease for the elapsed time for processing the delete part of the apply cycle. The 1 TB table saw the most decrease in absolute elapsed time, particularly at the lower velocities. However, the decrease was fairly significant for all data points in the 1 TB table. The 200 GB table also showed improvement at all points as well.

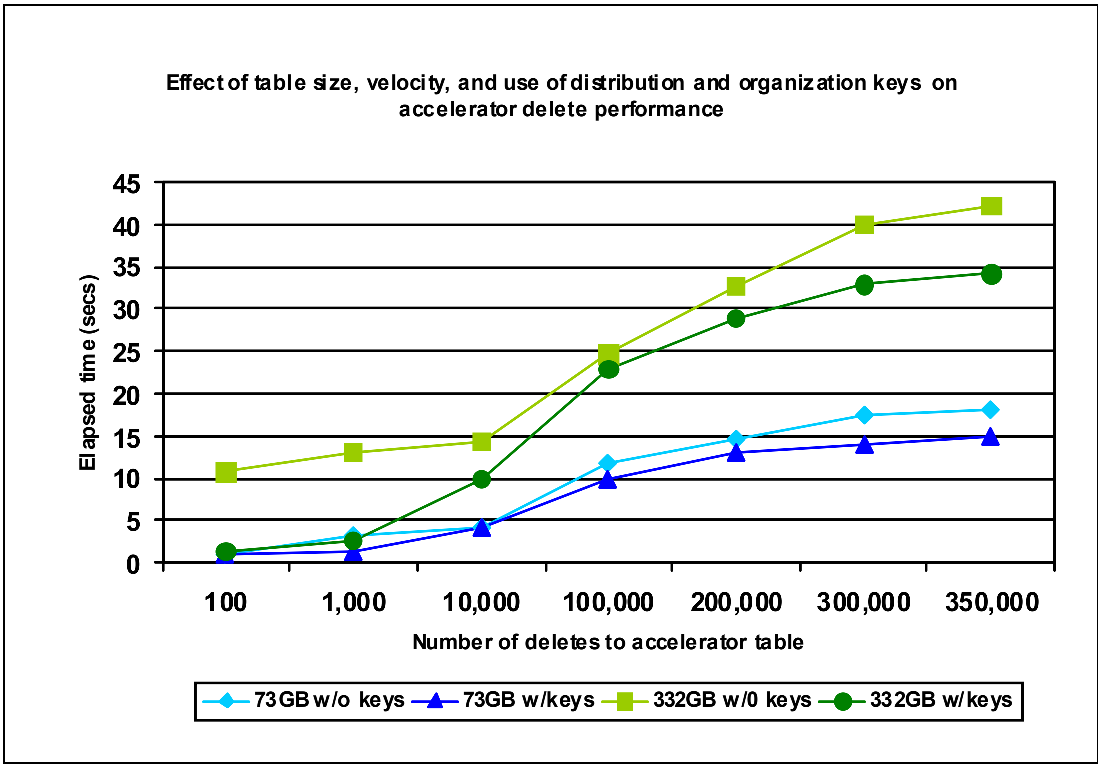


Figure 6-10 Effect of adding distribution and organization keys to Accelerator table

These numbers are just for one apply cycle, and should accumulate over time. To verify this, we take probably the worse case, where there is a high velocity of changes: Three million DB2 updates over about a five-minute time frame. This scenario ran three times each, with and without distribution and organization keys. This resulted in a total of nine million DB2 updates across a 15-minute total run time for each table.

Accumulating the total elapsed time for delete processing for both scenarios gives the results shown in Figure 6-11. There was a decrease in delete processing time of 85 seconds for the 200 GB table and 155 seconds for the 1 TB table. If the velocity of changes was lower, the savings would be greater, particularly for the 1 TB table.

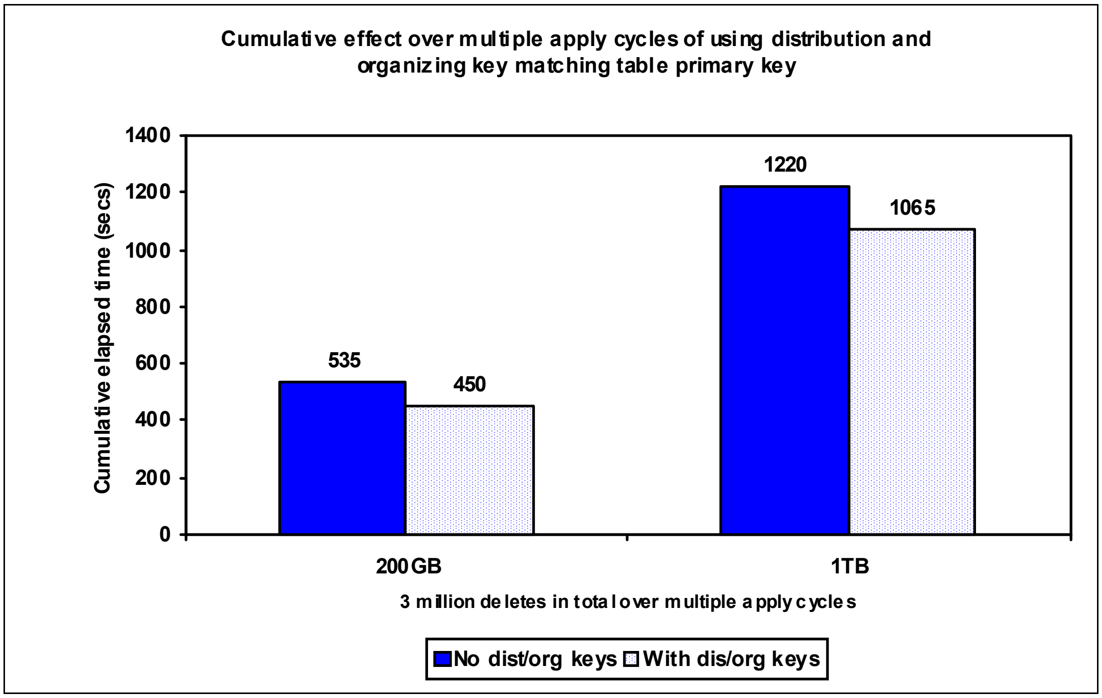


Figure 6-11 Cumulative effect using distribution and organization keys on delete processing

If you have large tables, it is advisable to implement distribution and organization keys on the DB2 Analytics Accelerator table matching the primary key of the DB2 table.

Another way to decrease the delete time for large tables is to reduce the scan time. The way to do that is to increase the capacity of DB2 Analytics Accelerator. This will increase the number of data slices and reduce the size of the data slices, in effect increasing the scan rate. Moving from a single-rack system to a double-rack system will double the number of data slices to 480, therefore increasing the effective scan speed from 32 GBps to 64 GBps.

6.3.4 Effect of number of tables on incremental update

The two previous sections looked at individual tables, particularly large tables. It is not unusual to have a large number of small tables. Therefore, we are interested in the time it takes to replicate changes for these small tables to DB2 Analytics Accelerator. We know that any DB2 updates or deletes result in delete processing on DB2 Analytics Accelerator. We know that delete processing on a DB2 Analytics Accelerator table results in some amount of table scanning due to the lack of indexes.

What we have not discussed yet is the fact that the tables during an apply cycle are processed serially. For each table that has changes during an apply cycle, the apply has to prepare the changes that exist in the apply staging area, generate and run the appropriate SQL statements, possibly a delete and insert for each table, and maintain the staging area and associated metadata. In this section, we explore how the number of tables affect incremental update.

Table 6-3 shows the minimum amount of apply time for each type of DB2 change. If we have DB2 inserts only or DB2 deletes only for a table, the minimum amount of time for the apply cycle for this table is one second. However, if we have any DB2 updates, or a combination of DB2 insert and delete activity, there is a minimum apply elapsed time of two seconds.

Table 6-3 Minimum SQL processing time in DB2 Analytics Accelerator on behalf of incremental update

DB2 change type	**DB2 Analytics Accelerator insert time**	**DB2 Analytics Accelerator delete time**	**Total time**
INSERT	1	0	1
UPDATE	1	1	2
DELETE	0	1	1

Those numbers are for one table. What if we have many tables, how does that affect the apply elapsed time? The following list includes some examples. Remember that a DB2 update becomes an DB2 Analytics Accelerator delete and insert:

- Five tables with DB2 inserts only:
  - Five DB2 Analytics Accelerator inserts
  - 5 x 1 = 5 seconds, minimum, per apply cycle
- Five tables with DB2 deletes only:
  - Five DB2 Analytics Accelerator deletes
  - 5 x 1 = 5 seconds, minimum, per apply cycle
- 5 tables with DB2 updates only:
  - Five DB2 Analytics Accelerator inserts and Five DB2 Analytics Accelerator deletes
  - (5 x 1) + (5 x 1) = 10 seconds, minimum, per apply cycle
- 10 tables with DB2 insert and delete activity:
  - 10 DB2 Analytics Accelerator inserts and 5 DB2 Analytics Accelerator deletes
  - (10 x 1) + (5 x 1) = 15 seconds, minimum, per apply cycle
- 10 tables with DB2 insert only and 5 tables with DB2 update only:
  - 15 DB2 Analytics Accelerator inserts and 5 DB2 Analytics Accelerator deletes
  - (10 x 1) + (5 x 2) = 20 seconds, minimum, per apply cycle
- 50 tables with DB2 insert, update, and delete activity:
  - 50 DB2 Analytics Accelerator inserts and 50 DB2 Analytics Accelerator deletes
  - (10 x 50) + (10 x 50) = 100 seconds, minimum, per apply cycle

These are the minimum times. Of course, additional time will be required for large tables, as discussed in the previous sections.

We can see that having hundreds of small tables in a replication scenario can be problematic, causing extremely long apply times. For example, if there are 200 tables that all have updates in one apply cycle, the apply cycle will take at least 400 seconds. So, what can you do?

First, examine the latency needs of those tables. Include only critical tables in incremental update. Perhaps a nightly refresh might be sufficient for some number of the tables.

Second, examine if some type of a frequent refresh scenario might suffice. We know that it will take at least 10 seconds, possibly 20, seconds for an apply cycle for our 10 small tables.

A batch refresh of those 10 small tables takes eight seconds with one invocation of the table load stored procedure, which loads the tables serially. You can probably reduce that eight seconds by starting multiple stored procedures, and load the tables in parallel.

Depending on your latency requirements, you could automate this full refresh to run at an appropriate interval, maybe every 5 minutes, 10 minutes, 15 minutes, and so on. In fact, it's conceivable that you could reduce the latency time for these 10 tables compared to them participating in a larger incremental update scenario.

If the number of tables can't be reduced enough to meet the latency requirements, another option would be to split the data across multiple accelerators. This, of course, would require that there is a logical organization of the tables, so that they could be split and not affect the ability to run queries.

6.3.5 One final example

Consider an example with our 200 GB table, our 1 TB table, and 50 small tables. The expected average change activity in most apply cycles is as follows:

- 30 small tables are expected to have at least one insert and update per apply cycle.
- 20 small tables have an average of one insert per apply cycle.
- The 200 GB table is expected to have 5000 inserts, 100 updates, and 5 deletes in one apply cycle.
- The 1 TB table is expected to have only 120,000 inserts per apply cycle.
- None of the tables have distribution or organization keys.

Based on these measurement results, we can estimate the total elapsed time (expected latency) for one apply cycle. By virtue of the 50 small tables having DB2 update activity, we know that will result in one insert and one delete per table during the apply, for an elapsed time of (50 x 2) = 100 seconds.

The 200 GB table results in one insert and one delete during apply time. We know that inserts average one second elapsed time. Referring to Figure 6-9 on page 129, we can estimate that the 105 deletes will take about three seconds, for a total apply time of four seconds for the 200 GB table.

The 120,000 inserts to the 1 TB table will take one second, and there will be no deletes.

Summing 100 seconds for the small tables, 4 seconds for the 200 GB table, and 1 second for the 1 TB table, we estimate that the elapsed time for one apply cycle is 105 seconds.

What happens if we were wrong in our assumption about the 1 TB table and, in fact, there are updates, averaging 1000, per apply cycle? We now have a delete operation to the 1 TB table during apply that we did not have before. Again, referring to Figure 6-9 on page 129, we can estimate that deleting those 1000 rows will add about 13 seconds to our apply cycle, for a new total of 118 seconds.

This example is fairly small. Actual examples might be much larger and more complicated.

We can see that having a large number of tables during any apply cycle can really elongate the time for completing the apply cycle.

6.4 Load versus incremental update

Should you choose a batch refresh or an incremental update strategy to maintain data in DB2 Analytics Accelerator? There are many factors to consider.

Of primary importance, what are your business requirements? What kind of latency does the business require? Is there sufficient business value for having a lower latency data ingest strategy? Alternatively, are the analytics needs such that a longer data latency is fine?

When the business requirements are known, is there the technical capability to support the required latency? DB2 Analytics Accelerator certainly cannot support a latency that is not available in the DB2 source tables. There are technical strengths and limitations to each technology as well. Figure 6-12 compares the technical characteristics of batch refresh and incremental update.

Batch refresh	Incremental Update
•**Utilizes batch streaming load method based on the DB2** UNLOAD **utility**	•**Uses separate log-reading capture and apply technology**
•**High performance for larger data volumes**	•**High performance for frequent, small, and arbitrary changes**
•**Granularity: table or partition**	•**Granularity: single row grouped by unit of work – lower latency**
•**High throughput: up to 1.5 Terabyte per hour**	•**Lower throughput: <= 18 GB/hr**
•**Can handle data changes that are not logged**	•**Will not see changes that are not logged**
•**Lower processor cost per MB**	•**Higher processor cost per MB**
•**Explicitly invoked**	•**Runs continuously propagating data resulting in lower latency for moving changes**

Figure 6-12 Comparison of batch refresh and incremental update characteristics

Sometimes, the decision can come down to the System z CPU costs for one versus the other. For a given amount of data, the cost to batch load that amount of data is going to be less than replicating the same amount of data. Therefore, the System z CPU cost depends on the amount of data transferred with each technology.

If the percentage of changed data is low compared to the size of a table or partition, the cost of incremental update will most likely be less than loading the entire partition or table. If the percentage of changed data is large compared to the size of the table, perhaps batch load will be less costly in terms of System z CPU.

In Figure 6-13, we see an example of calculating the cost of incremental update versus batch load for varying percentages of changes. We want to determine at what point it costs more to replicate data versus a batch load.

We make the following assumptions:

- The partition is 10,000 MB.
- Changes are DB2 updates only.
- The cost of the load is 0.5 seconds per 100 MB.
- The cost of replicating updates is 34 seconds per 100 MB.
- There are no out-of-scope changes.

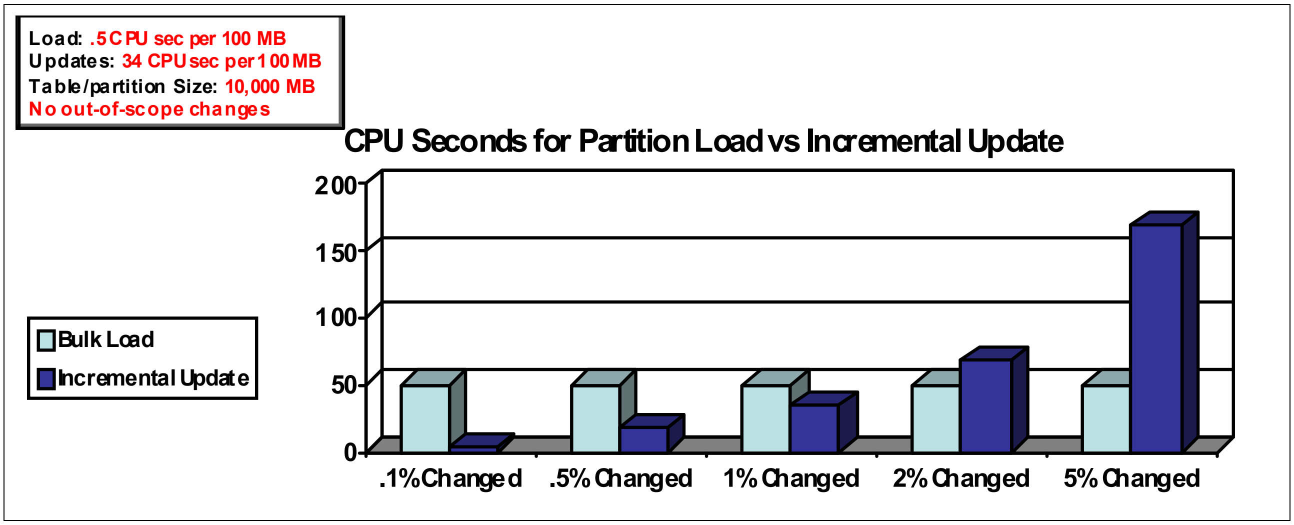


Figure 6-13 Cost of incremental update versus batch load example

The cost of batch load is always the same, because all of the data is loaded in each instance (at about 50 seconds in this example). The cost of incremental update increases as the number of changes increase. When the percentage of changed data is very low, the cost of replicating just those changes is less than the cost of loading all of the data.

However, in this example, somewhere between 1% changed data and 2% changed data, the costs of loading the entire partition becomes less expensive. After that, the cost of replication data increases rapidly.

DB2 Analytics Accelerator resource usage patterns for batch load and incremental update differ. As shown in Figure 6-14 on page 136, load can cause heavy CPU resource use on the DB2 Analytics Accelerator host node.

Incremental update, on the other had, can cause a heavy load on DB2 Analytics Accelerator worker nodes. Heavy change activity can cause much data scanning to process the deletes that occur on DB2 Analytics Accelerator on behalf of DB2 updates and deletes.

Figure 6-14 shows load and incremental update resource usage on host and worker nodes.

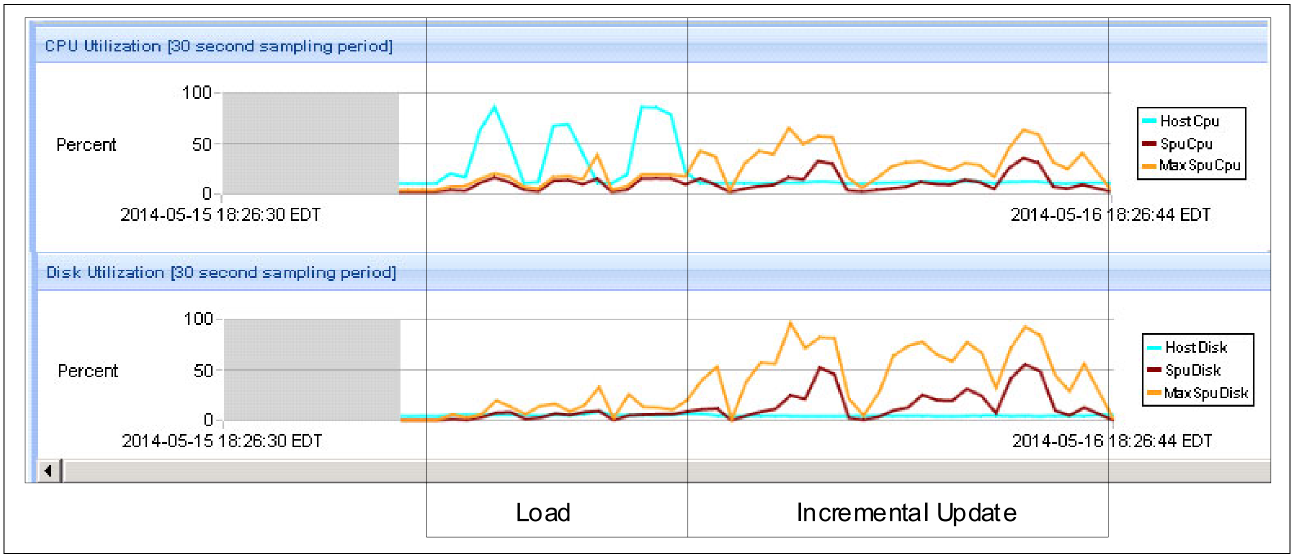


Figure 6-14 Load and incremental update has different usage patterns on DB2 Analytics Accelerator resources

In many DB2 Analytics Accelerator deployments, data ingestion requirements are not discussed until late in the deployment process. Sometimes, the data ingestion requirements will be the driving factor in the initial sizing of DB2 Analytics Accelerator. Try to understand data ingestion requirements as early as possible.

7

Online data archiving

This chapter introduces the IBM DB2 Analytics Accelerator for z/OS (DB2 Analytics Accelerator) High Performance Storage System (HPSS) feature and its functional capabilities. HPSS enables users to archive partitions of a table, or the entire table, from the DB2 environment to DB2 Analytics Accelerator.

HPSS is available in DB2 Analytics Accelerator V4.1, and provides enhanced features, such as better access control to HPSS archived partitions, improved flexibility and protection of image copies created by HPSS, built-in restore for HPSS, and HPSS archiving for multiple accelerators.

The following topics are described in this chapter:

- HPSS overview
- Online data archive scenarios
- Online restore of archived partitions

7.1 HPSS overview

The HPSS feature of DB2 Analytics Accelerator uses the data and access characteristics of the enterprise data warehouses (EDWs) and the operational data stores (ODSs). It provides the ability to move less-frequently used and changed partitions or tables from DB2 into DB2 Analytics Accelerators.

The data belonging to older partitions is moved to DB2 Analytics Accelerator and purged from DB2. From this perspective, the feature can be considered an archiving option, and the partitions that are only in DB2 Analytics Accelerator are often referred to as *archived partitions*. However, note that there is no negative aspect regarding reduced availability of data, which is usually a by-product of archiving. The partitions that are in both DB2 and DB2 Analytics Accelerator are referred to as *non-archived partitions*.

HPSS offers several benefits:

- Reduced usage in physical disk storage in IBM System z.
- With the data being removed from the DB2 table, the size of corresponding indexes shrinks, which further reduces disk storage requirements. However, more importantly, removing the data improves performance when accessing the remaining data.
- Administrative operations on the data remaining in DB2, such as backup and recovery times, become faster because of the reduced data volume.
- The data moved to DB2 Analytics Accelerator is still available, and it can be accessed through Structured Query Language (SQL) queries. Access to the archive data in DB2 Analytics Accelerator delivers the usual high performance that DB2 Analytics Accelerator (as the query Accelerator) has always had.
- HPSS is one of the most advanced multi-temperature data solutions, and the next step toward an integrated online transaction processing (OLTP) and online analytical processing (OLAP) system.
- HPSS tables can exist on multiple accelerators, which provides high availability (HA).
- HPSS V2.0 provides the ability to archive multiple HPSS tables on multiple accelerators.
- HPSS is simple to use.

7.1.1 Archiving enhancements

The enhanced features of HPSS are only available in DB2 Analytics Accelerator V4.1 and later. The following list describes the archiving enhancements:

- DB2 partitions are set to persistent read-only (PRO) status after the partition is moved to DB2 Analytics Accelerator, which prevents future inserts or updates to the partitions.
- Created image copies are protected by the PRO status of the DB2 partitions. No further image copies can be created. Image copy entries in SYSCOPY are kept even when MODIFY RECOVERY is run. However, physical image copy data sets are not protected.
- Up to four copies per partition are possible. This improves disaster recovery (DR) scenarios.
- A flexible naming schema for the image copies can be defined. This simplifies the restart process or re-archiving, because the image copies can no longer be deleted manually beforehand.

- The ability to archive a table on multiple DB2 Analytics Accelerators is added. Archive does not fail if the table is already archived on another accelerator. Instead, the stored procedure will detect that the table is already archived on another accelerator, and will switch to `Archive from image copy` mode.
- The capability to restore archived partitions or tables moved with HPSS is added. This process restores the moved data to its original locations in DB2. Restore makes the data available in DB2 Analytics Accelerator for regular accelerated queries.

Following are the basic software requirements for the use of HPSS:

- DB2 Analytics Accelerator V4.1 or later
- IBM DB2 10 for z/OS or later
- IBM DB2 Analytics Accelerator Studio V4 or later

There are two stored procedures required to automate the HPSS archive and restore processes. They are SYSPROC.ACCEL_ARCHIVE_TABLES and SYSPROC.ACCEL_RESTORE_ARCHIVE_TABLES:

- Stored procedure SYSPROC.ACCEL_ARCHIVE_TABLES moves table partitions from DB2 for z/OS to a storage area on a DB2 Analytics Accelerator.
- Stored procedure SYSPROC.ACCEL_RESTORE_ARCHIVE_TABLES restores the data of moved partitions to their original locations. It works on partitions that were moved with HPSS.

  This stored procedure is available to automate the process to change one or more partitions from archived partitions (stored in DB2 and DB2 Analytics Accelerator). The restore is done from image copy. This feature is useful to update data and archive again.

To be able to mark accelerators with maintenance needed, and control the offloading of queries, the states as a result of the archive or restore operations are now defined in the ENABLE and ARCHIVE columns of SYSACCEL.SYSACCELERATEDTABLES, listed in Table 7-1 on page 140.

Table 7-1 lists the possible states resulting from archive or restore operations.

Table 7-1 ENABLE and ARCHIVE columns of SYSACCEL.SYSACCELERATEDTABLES

Column	State	Description	Query offloading
ARCHIVE	blank	The table is not archived in a DB2 Analytics Accelerator server.	DB2 Analytics Accelerator can answer normal queries.
	A	The table is archived in the DB2 Analytics Accelerator server that is specified by the DB2 Analytics ACCELERATORNAME column value of this table. The DB2 Analytics Accelerator server contains active and archived data.	DB2 Analytics Accelerator can answer normal and archive queries.
	C	The table is archived in other DB2 Analytics Accelerator servers. The DB2 Analytics Accelerator server that is specified by the DB2 Analytics AcceleratorNAME value contains only active data.	DB2 Analytics Accelerator can answer normal queries.
	B	The table is partially archived on the DB2 Analytics Accelerator server that is specified by the DB2 Analytics AcceleratorNAME value. DB2 Analytics Accelerator can answer queries with active data.	DB2 Analytics Accelerator can answer normal queries.
	X	The table was restored on another accelerator. The DB2 Analytics Accelerator that is specified by the DB2 Analytics AcceleratorNAME value cannot answer queries.	n/a The ENABLE column is set to “T” in this case.
ENABLE	Y	Enabled	Yes
	N	Disabled	No
	T	Transition	No

7.1.2 HPSS archive data process and operations

The archive partition process uses the SYSPROC.ACCEL_ARCHIVE_TABLES stored procedure to move data from one or more table partitions in DB2 to a DB2 Analytics Accelerator. The stored procedure triggers or completes the following tasks against the partition or table to be moved:

- Locks the partition or table
- Unloads data from the partition or table
- Creates the image copies
- Creates (adds) a table in DB2 Analytics Accelerator
- Transfers data into DB2 Analytics Accelerator
- Sets the partition to the `UT` state
- Removes data from DB2 table (pruning)
- Invalidates the DB2 dynamic statement cache for the tables

During the archive process, DB2 image copies are created automatically. These image copies are used to restore the original data to DB2.

The stored procedure enables up to four new image copies to be created for each moved partition. The stored procedure uses the DB2 COPY utility to create up to two local images and two recovery images.

The naming convention for the image copy data sets is controlled by the values specified in the new environment variables in the AQTENV data set:

- AQT_ARCHIVE_COPY1
- AQT_ARCHIVE_COPY2
- AQT_ARCHIVE_RECOVERYCOPY1
- AQT_ARCHIVE_RECOVERYCOPY2

The following requirements apply to the values:

- Variables can be used as documented for the DB2 COPY utility. The &SEQ, &LIST, and &DSNUM variables are not valid in the DB2 Analytics Accelerator context.
- The values must evaluate to qualifiers that are mapped with DFSMS to a suitable data class (as in DB2 Analytics Accelerator V3.1 with the static high-level qualifier (HLQ) prefix).
- The values must ensure uniqueness of names among all archived partitions. The suggestion is to use the &PART and &UNIQ variables.
- The values must evaluate to valid IBM z/OS data set names.

When a partition is moved to DB2 Analytics Accelerator, accelerated queries can only be run against the data in that partition if the GET_ACCEL_ARCHIVE=YES special register or subsystem parameter is used. The table data can no longer be updated, not in the original DB2 partition, and not on DB2 Analytics Accelerator, because DB2 partitions are set to the PRO state.

Tables are processed sequentially. However, partitions in a table can be moved in parallel. By default, the data is transferred four at a time. This can be changed by adjusting the AQT_MAX_UNLOAD_IN_PARALLEL environment variable in the AQTENV data set.

> **Note:** It is possible to run the accelerated queries while partitions are being moved. However, it cannot be determined if the result is included in the archived data.

The removal process is called *pruning*, and it requires exclusive access to the tables in a partition. The stored procedure issues the `START DATABASE ACCESS(UT)` command to make the partition available only for DB2 online utilities (and the SQL DROP statement).

The UT lock remains active when the stored procedure fails in the pruning process, but enables the process to resume where it left off. The number of retries, and the time interval between retries, are controlled by the AQT_MAX_RETRIES_DSNUTILU and AQT_SECONDS_BEFORE_RETRY_DSNUTILU environment variables.

7.1.3 Restore HPSS archived data process and restrictions

HPSS provides a full-scale, built-in restore function for HPSS data. The restore operation uses the SYSPROC.ACCEL_RESTORE_ARCHIVE_TABLES stored procedure to recover data from image copies created in the HPSS archive process. It ensures that indexes are up-to-date, and that data is consistent on DB2. It moves the partitions into the *online storage* on DB2 Analytics Accelerator. After HPSS restore, partitions become visible for any kind of queries.

HPSS can be used to restore the data of one or more explicitly specified partitions, or of an entire table. These partitions need to be defined on a DB2 Analytics Accelerator, and DB2 Analytics Accelerator has to be active (in online status). If DB2 Analytics Accelerator is unavailable, or if the table had been removed from DB2 Analytics Accelerator, you can only recover the DB2 data manually from the DB2 copies.

If the archived partitions have been maintained in the PRO state after the partition move, it is assured that the RECOVER TO LASTCOPY operation will restore the data back to DB2 using the image copies from the archived process. A sample job in the SAQTSAMP(AQTSJI04) is provided for your reference.

Restores of multiple partitions in a table are processed in parallel. However, each table is restored one at a time. By default, the data of 1 - 4 partitions can be restored concurrently. This default can be changed by adjusting the AQT_MAX_UNLOAD_IN_PARALLEL environment variable in the AQTENV data set.

If an operation fails on a single partition, only the failing partition is affected. All changes to previously processed partitions are kept and not rolled back. For those partitions that failed during the restore process, you must call the SYSPROC.ACCEL_RESTORE_ARCHIVE_TABLES procedure again, after fixing detected problems, to resume processing.

You can use the Restore partitions to DB2 function in DB2 Analytics Accelerator Studio to display the restore error status of each partition. You can also start the SYSPROC.ACCEL_GET_TABLES_INFO stored procedure to detect tables with incompletely restored partitions, and the SYSPROC.ACCEL_GET_TABLES_DETAILS stored procedure to obtain information about partitions that must be processed again because the restore operation was interrupted.

If an error occurs during the restore process, indexes are not rebuilt. They are put into a rebuild pending state. This might cause performance degradation on queries using these indexes, and cause the static SQL queries to fail due to unavailable resources.

During the recovery of moved partitions, the following utilities are run in the order indicated:

1. The RECOVER utility restores the original DB2 data from the image copies created at the time the partitions are moved. The utility works on a partition level. Several threads can run in parallel if multiple partitions are specified.
2. The REBUILD INDEX utility is used to rebuild all of the indexes of the tables that were recovered. The utility is run once per table, and rebuilds all data for partitioned indexes and non-partitioned indexes.
3. The CHECK DATA utility is started automatically for partitions that are in a check-pending state after a restore operation. This checks the consistency of the restored data.

The environmental variable AQT_SORDEVT enables you to explicitly specify a device name, such as SYSDA, that is used in the REBUILD INDEX and CHECK DATA utility statement.

7.2 Online data archive scenarios

This section provides information about environment setup and the user scenarios, to explain the details of online archiving. We begin with the environment setup and preparation for archiving partitions, and then proceed with the data archive of several partitions. After the data move, we show how the query results are affected by the archive process.

7.2.1 Prepare for archiving partitions and tables

In this section, we look at setting up the environment variables that are referenced by the SYSPROC.ACCEL_ARCHIVE_TABLES stored procedure to determine how many local copies are created for each moved partition, and how many recovery copies, and their data set names. It is required that at least one copy must be set. No default value exists for this variable.

Environment variables

In our environment, we chose to set up all of the variables, two copies each for local and recovery image copies. This is shown in Figure 7-1.

```
AQT_ARCHIVE_COPY1=D10KRED1.&DB..&TS..P&PART..&UNIQ.
# Dataset name template for first local copy image created per partition
# by SYSPROC.ACCEL_ARCHIVE_TABLES.
AQT_ARCHIVE_COPY2=D10KRED1.&DB..&TS..P&PART..&UNIQ.
# Dataset name template for second local copy image
AQT_ARCHIVE_RECOVERYCOPY1=D10KRED1.&DB..&TS..P&PART..&UNIQ.
# Dataset name template for first recovery copy image
AQT_ARCHIVE_RECOVERYCOPY2=D10KRED1.&DB..&TS..P&PART..&UNIQ.
# Dataset name template for second recovery copy image
```

Figure 7-1 Data set name template for archive in our environment

Important: The AQT_ARCHIVECOPY_HLQ environment variable that was present in the earlier version of the HPSS is deprecated. Remove it from your AQTENV data set.

To avoid any issue in the allocation of storage space larger than 65,535 tracks, we had defined and used a data class in the storage management subsystem (SMS) that supports larger sequential data sets. We also modified the automatic class selection (ACS) routine to associate the new data class with the range of template data set names for image copies.

DB2 tables and partitions

There are certain conditions to be met before you can move the partitions with the SYSPROC.ACCEL_ARCHIVE_TABLES stored procedure:

- The DB2 table must be partitioned by range.
- The table must have been defined or added to DB2 Analytics Accelerator.

- The table must be in InitialLoadPending or Loaded state on DB2 Analytics Accelerator.

  Important: This implies that incremental updates must be disabled for the table, or that incremental updates must be stopped for the entire DB2 subsystem.

- Databases and table spaces that contain the table partitions to be moved must not be in read-only state, stopped state, or any other state that prevents an update of the data in the partition. Ideally, the database or partition should be in read/write (RW) state, as shown in Figure 7-2.

For our scenario, we want to move the last five partitions of the BOOKDB.CUSTOMER_DIM table. This is a range-partitioned table on the TSDCUST table space. To verify its status on DB2 for z/OS, we issue the following DB2 display command:

-BZA1 DIS DB(BOOKDB) SPACE(TSDC*)

The output of this command is captured on Figure 7-2.

```
-BZA1DIS DB(BOOKDB) SPACE(TSDC*)
DSNT360I  -BZA1 ***********************************
DSNT361I  -BZA1 *  DISPLAY DATABASE SUMMARY 862
                *    GLOBAL
DSNT360I  -BZA1 ***********************************
DSNT362I  -BZA1     DATABASE = BOOKDB  STATUS = RW 864
                   DBD LENGTH = 88826
DSNT397I  -BZA1 865
NAME     TYPE PART  STATUS            PHYERRLO PHYERRHI CATALOG
-------- ---- ----- ----------------- -------- -------- -------
TSDCUST  TS    0001 RW
    -THRU      0100
TSDCUST  TS
******* DISPLAY OF DATABASE BOOKDB   ENDED      ***************
```

Figure 7-2 Table space BOOKDB.TSDCUST is partitioned and in read-write status

There are a few more requirements to be met before we can archive the partitions in the table:

- The table must exist on DB2 Analytics Accelerator.
- The status should either be InitialPending or Enabled.

The following section describe the steps to meet these requirements:

1. From Data Studio, connect to the BZA1TFIN and then to BZA1STR.
2. Add the table BOOKDB.CUSTOMER_DIM on BZA1TFIN, then to BZA1STR.
3. Load the data into both DB2 Analytics Accelerators and check the status.

Figure 7-3 shows, using the Data Accelerator Studio, that the last Load Acceleration status for BOOKDB.CUSTOMER_DIM table is Enabled on both accelerators.

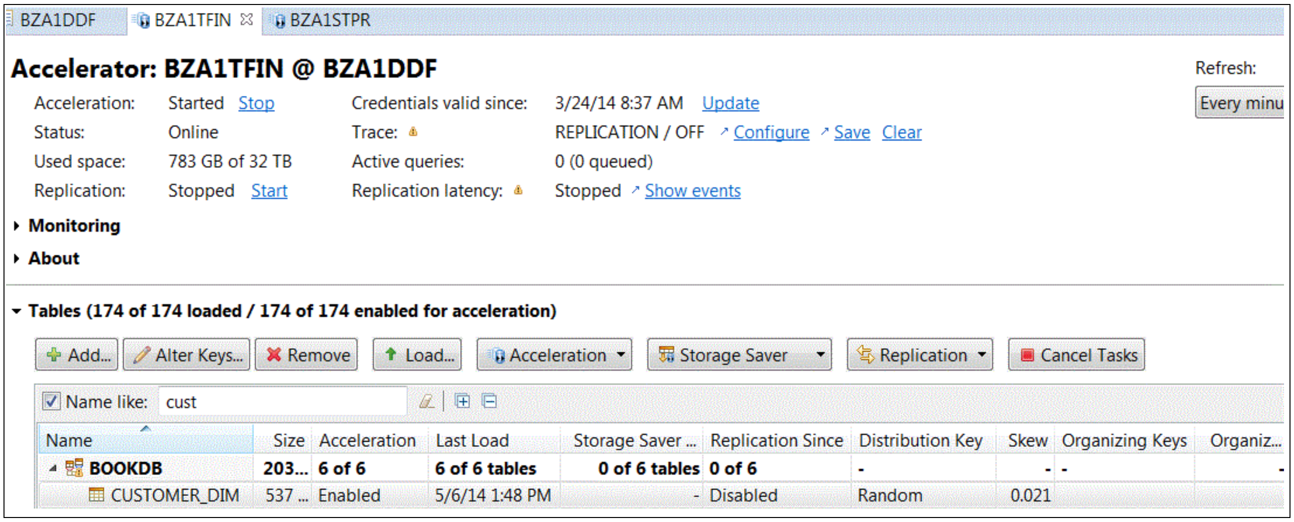


Figure 7-3 Last Load status for BOOKDB.CUSTOMER_DIM table is Enabled

Figure 7-4 shows the status of BOOKDB.CUSTOMER_DIM after all partitions of the table were copied (loaded) into DB2 Analytics Accelerator.

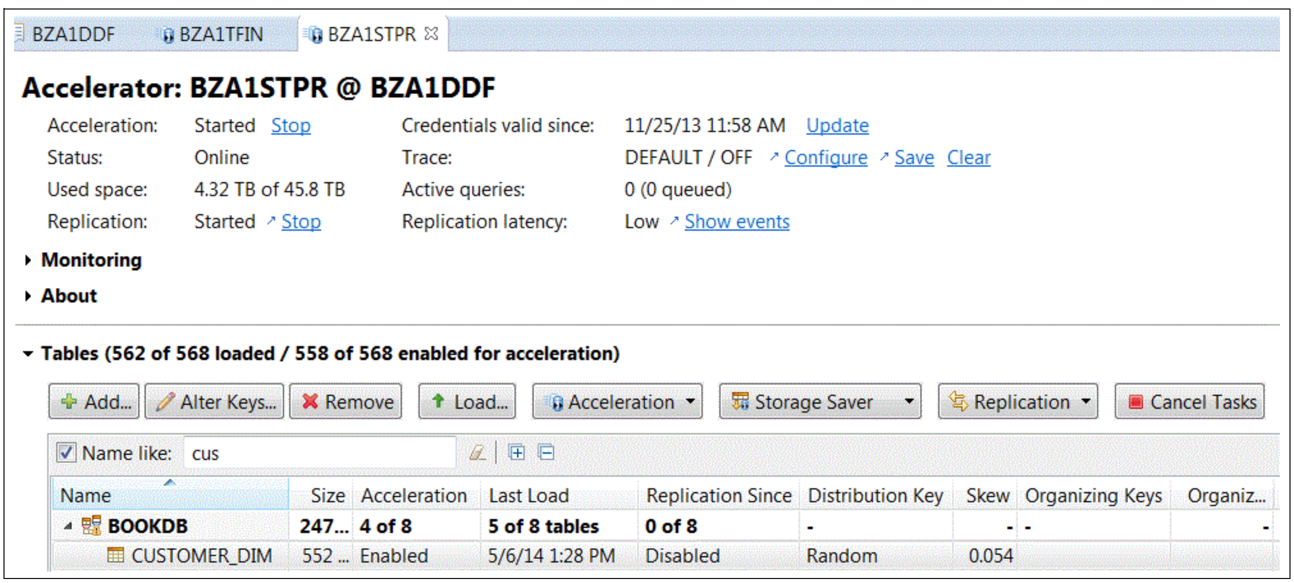


Figure 7-4 Status of BOOKDB.CUSTOMER_DIM after loading all partitions

7.2.2 Archive partitions on Accelerator 1 (BZA1TFIN)

In this section, we describe the steps for archiving the last five partitions of the BOOKDB.CUSTOMER_DIM table.

> **Requirement:** DB2 Analytics Accelerator requires that the table is *not* enabled for replication.

To archive partitions 96 to 100 of the BOOKDB.CUSTOMER_DIM table into BZA1TFIN, follow these steps:

1. In Data Studio, highlight the CUSTOMER_DIM table, then click the down arrow on the Storage Saver box.
2. Choose **Move Partitions to Accelerator**. You get a warning that describes what happens to the data after the move, as shown in Figure 7-5. Click **Yes**.

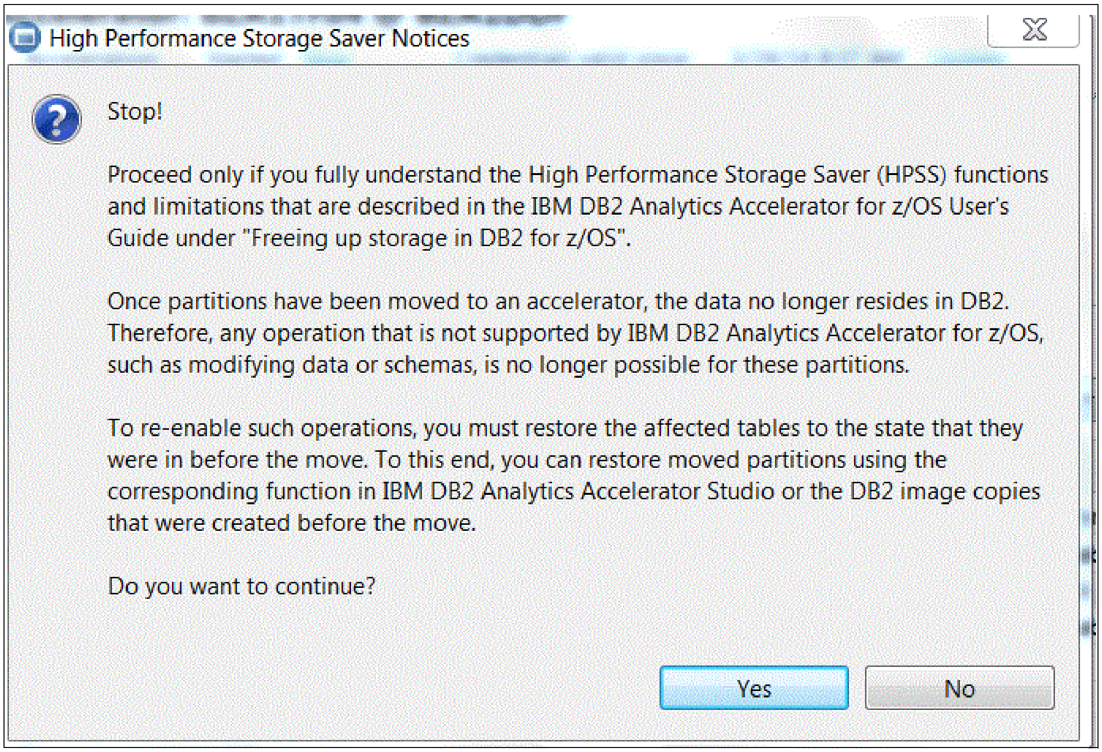


Figure 7-5 Warning about what happens to the data after the move

3. There are three options in the Move Selected Partitions pane shown in Figure 7-6. Select **Manually select partitions**, and select partitions 96 - 100. Click **OK**.

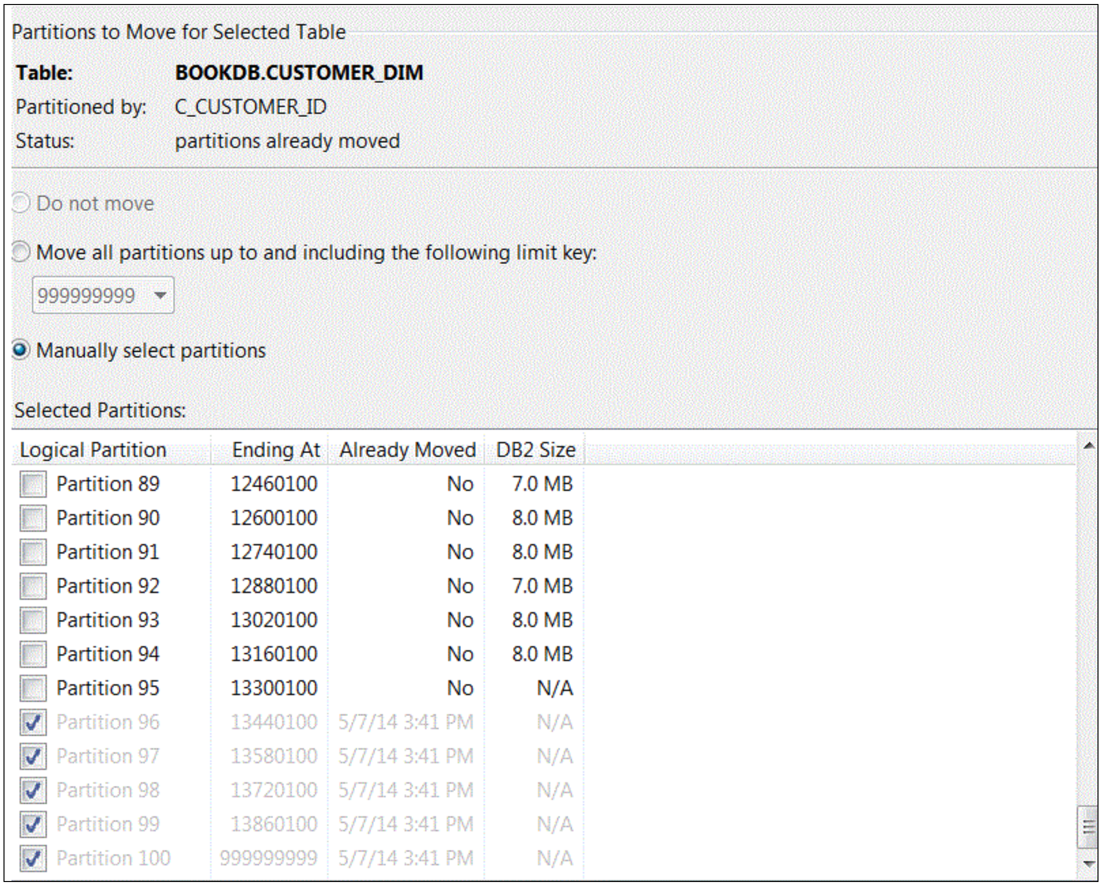


Figure 7-6 Move Selected Partitions pane

> **Note:** After partitions 96 to 100 are archived, data from the archived partitions will no longer be available in DB2 for z/OS. The archived partitions are still available in both accelerators for query. These partitions are still active in BZA1STPR.

DB2 image copies now exist for the archived partitions, two each for local and recovery, using the naming conventions in the environment variables set in AQTENV. See Example 7-1.

Example 7-1 Image copies for the archived partitions

```
D10KRED1.BOOKDB.TSDCUST.P00096.EPPSUUZQ
D10KRED1.BOOKDB.TSDCUST.P00096.EPPSUU0J
D10KRED1.BOOKDB.TSDCUST.P00096.EPPSUU1A
D10KRED1.BOOKDB.TSDCUST.P00096.EPPSUU1U
```

Partitions 96 to 100 of the BOOKDB.CUSTOMER_DIM table are now in a PRO state to prevent any update on these partitions. See Figure 7-7.

```
NAME     TYPE PART  STATUS            PHYERRLO PHYERRHI CATALOG  PIECE
-------- ---- ----- ----------------- -------- -------- -------- -----
TSDCUST  TS    0096 RW,PRO
    -THRU      0100
TSDCUST  TS
******* DISPLAY OF DATABASE BOOKDB   ENDED      *********************
```

Figure 7-7 CUSTOMER_DIM table in PRO status after partitions were archived on BZA1TFIN

When you display the accelerated table using the DB2 Administration Tool, it shows two different statuses for BZA1TFIN and BZA1STPR. The statuses are A (active and archived) and C (archived on another accelerated). This is captured in Figure 7-8.

```
DB2 Admin ----------- BZA1 Display Accelerated Tables ------- Row 608 from 742
Command ===>                                                   Scroll ===> CSR

Line commands:
 I - Interpret  AC - Accelerator  T - Table

   Table              Table    Server       Remote   Remote
S  Name               Schema   Name     E A Name     Schema   Refresh Time
   CUST*              BOOKDB*  *        * * *        *        *
-- ------------------ -------- -------- - - -------> -------- ---------------->
   CUSTOMER_DIM       BOOKDB   BZA1TFIN Y A CUSTOMER BOOKDB   2014-05-06-13.27.
   CUSTOMER_DIM       BOOKDB   BZA1STPR Y C CUSTOMER BOOKDB   2014-05-06-13.28.
```

Figure 7-8 Status of BOOKDB.CUSTOMER_DIM table on the two Accelerators

Query execution on Accelerator with HPSS online archive

With the introduction of HPSS online archive, before each query request on DB2, the DB2 query optimizer considers query processing against recent, non-archived partitions, or archived partitions. The optimizer checks if the following conditions are met:

- Query acceleration is enabled.
- All tables referenced by the query have been loaded into DB2 Analytics Accelerator.
- The query qualified and meets the routing criteria that it supports query constructs.
- The same heuristics decisions, optimal query access plan, and cost-based optimization for query routing apply.

Access to archived partitions can be enabled or disabled:

- By setting the CURRENT GET_ACCEL_ARCHIVE special register
- In the GET_ACCEL_ARCHIVE subsystem parameter value in the DSNZPARM
- By the corresponding BIND parameter for static SQL

> **Important:** If CURRENT GET_ACCEL_ARCHIVE = YES, the query can only run in DB2 Analytics Accelerator. DB2 optimizer heuristics check and query do not apply.

First scenario

In our first scenario, the last five partitions of the BOOKDB.CUSTOMER_DIM table were archived on BZA1TFIN. We run the following query to select the first 10 rows in the last partition of the table. We set CURRENT QUERY ACCELERATION = ALL. The query returned the first 10 rows of the table, as shown in Figure 7-9.

```
SELECT * FROM BOOKDB.CUSTOMER_DIM
  WHERE C_CUSTOMER_ID > 13860100
  FETCH FIRST 10 ROWS ONLY WITH UR;
---------+---------+---------+---------+---------+---------+---------+---------
+
C_CUSTOMER_ID  C_FIRST_NAME                    C_LAST_NAME
C
---------+---------+---------+---------+---------+---------+---------+---------
+
     13860125  Leslie                          Ryon
7
     13860182  K A                             Grim
1
     13861440  M                               Airth
2
     13860290  John R & Virginia               Macedo
2
     13860365  Donna                           Moses
5
     13860707  A L                             Fox
1
     13860716  W                               Biggers
3
     13860757  Hope                            Mitchell
4
     13860805  V P                             Nicks
3
     13860820  Thomas A                        Carreras
6
DSNE610I NUMBER OF ROWS DISPLAYED IS 10
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 100
```

Figure 7-9 First 10 rows of archived partition 96 in BOOKDB.CUSTOMER_DIM table

Second scenario

The second scenario is to force the query to run the query on DB2 with GET_ARCHIVE = YES. We set the QUERY_ACCELERATION value to NONE. This query returned an error. It got a SQLCODE -4742 and reason code 2. The archived partitions can no longer be available for query on DB2. Figure 7-10 shows the query and its result for this scenario.

```
SET CURRENT QUERY ACCELERATION = NONE;
---------+---------+---------+---------+---------+---------+---------+---------
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
---------+---------+---------+---------+---------+---------+---------+---------
SET CURRENT GET_ACCEL_ARCHIVE = YES;
---------+---------+---------+---------+---------+---------+---------+---------
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
---------+---------+---------+---------+---------+---------+---------+---------
SELECT * FROM BOOKDB.CUSTOMER_DIM
  WHERE C_CUSTOMER_ID > 13860100
  FETCH FIRST 10 ROWS ONLY WITH UR;
---------+---------+---------+---------+---------+---------+---------+---------
DSNT408I SQLCODE = -4742, ERROR:  THE STATEMENT CANNOT BE EXECUTED BY DB2 OR
       IN DB2 Analytics Accelerator (REASON 2)
DSNT418I SQLSTATE   = 560D5 SQLSTATE RETURN CODE
DSNT415I SQLERRP    = DSNXODML SQL PROCEDURE DETECTING ERROR
DSNT416I SQLERRD    = -534 0  0  -1  0  0 SQL DIAGNOSTIC INFORMATION
DSNT416I SQLERRD    = X'FFFFFDEA'  X'00000000'  X'00000000'  X'FFFFFFFF'
         X'00000000'  X'00000000' SQL DIAGNOSTIC INFORMATION
DSNE618I ROLLBACK PERFORMED, SQLCODE IS 0
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
```

Figure 7-10 Query result output with error

Important: Note that accelerated queries run on either DB2 Analytics Accelerator server, *if* both of the servers are started, and *if* the table exists in both. DB2 does not consider data currency when routing the query to the servers.

7.2.3 Archive partitions on Accelerator 2 (BZA1STPR)

In this section, we describe the process of archiving some partitions of the same table that exists on another accelerator with archived partitions. We also show how the query is routed when data exists on both accelerators:

1. To start the process, from DB2 Analytics Accelerator Data Studio, we verify that BOOKDB.CUSTOMER_DIM table has an Enabled status on the second accelerator, BZA1STPR. However, we notice that Replication is started for this table, as shown in Figure 7-11 on page 151.

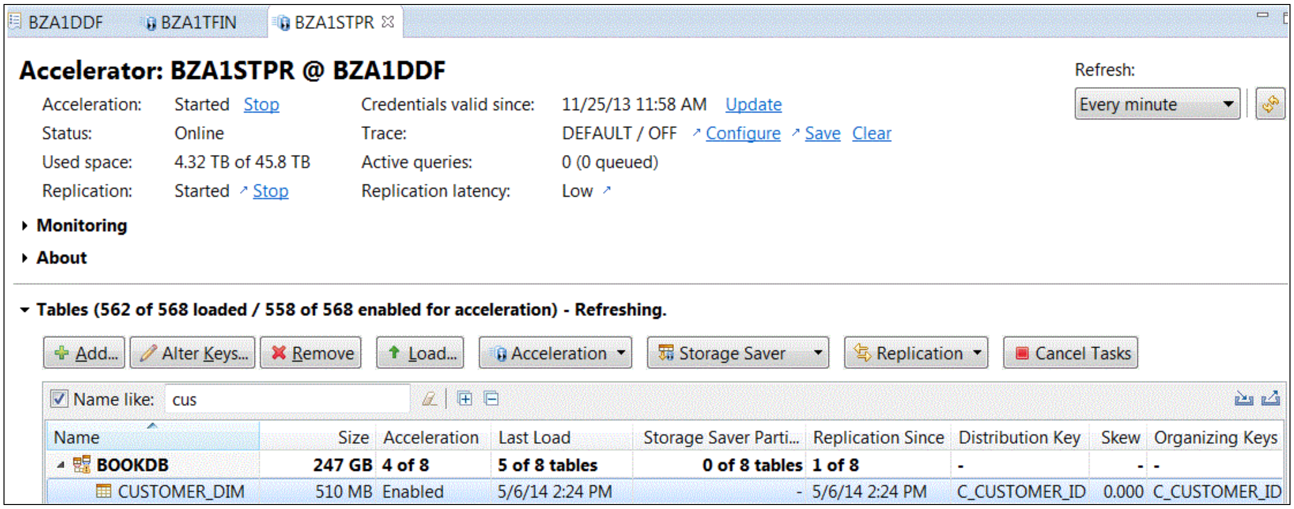


Figure 7-11 Replication is active

Figure 7-8 on page 148 shows that the BOOKDB.CUSTOMER_DIM table is replication-enabled.

2. There are two ways to disable replication, at the system level or the table level. For this experiment, we disable replication only for the CUSTOMER_DIM table, as shown in Figure 7-12.

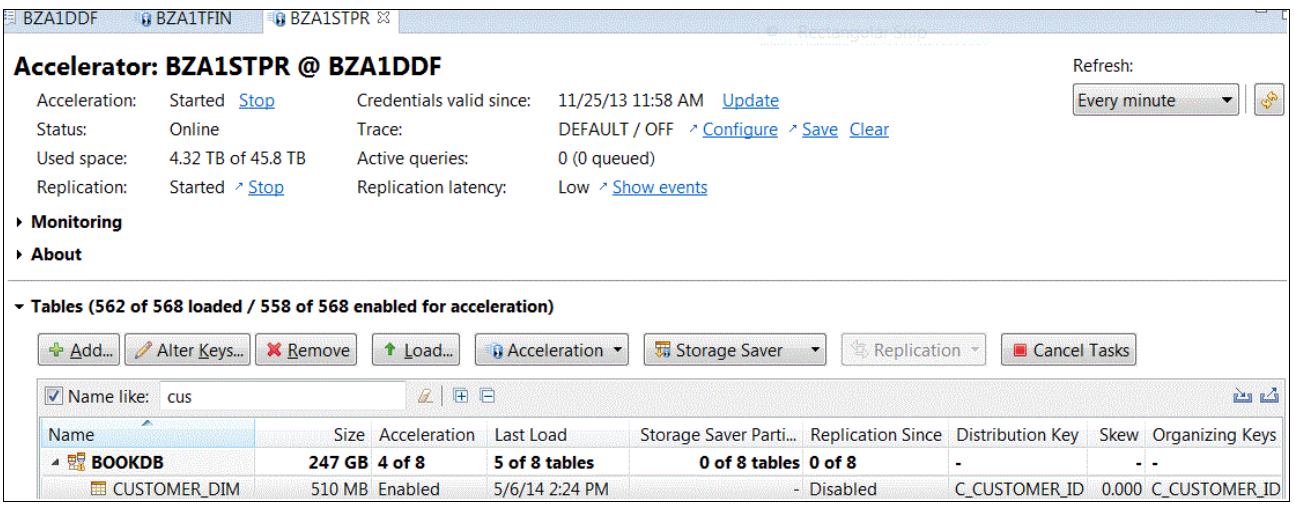


Figure 7-12 Replication is disabled for BOOKDB.CUSTOMER_DIM table

To archive partitions 96 to 100 of the CUSTOMER_DIM table, follow these steps:

1. In Data Studio, highlight the CUSTOMER_DIM table, and click the down arrow on the Storage Saver box.
2. Choose **Move Partitions to Accelerator**. You get a warning that describes what happens to the data after the move. Click **Yes**.
3. Select **Manually select partitions** and partitions 96 to 100. Click **OK**.

Because partitions 96 to 100 were previously archived, image copies already exist. Rather than creating another set of image copies of the partitions, DB2 uses the existing image copy to load the data into partitions 96 to 100 of the table.

For example, Figure 7-13 shows that the created and the reference dates for the D10KRED1.BOOKDB.TSDCUST.P00100.EPPSUUZS image copy data set are April 22, 2014 (date partition 100 was archived on BZA1TFIN) and May 1, 2014 (the date when partition 100 was archived on BZA1STPR).

```
Data Set Name . . . . : D10KRED1.BOOKDB.TSDCUST.P00100.EPPSUUZS

Creation date . . . : 2014/04/09
Referenced date . . : 2014/05/06
Expiration date . . : ***None***
```

Figure 7-13 DB2 uses existing image copy if the partition is archived on another Accelerator

Using the DB2 Administration Tool on z/OS, when we display the status of the same table in two different accelerators, it shows the statuses of the CUSTOMER_DIM table on BZA1TFIN and BZA1STPR are both A, which means active and archived. Additional information about the tables is shown in Figure 7-14.

```
DB2 Admin--- BZA1 Interpretation of an Object in SYSACCELERATEDTABLES
Command ===>

Details for accelerated table (label): BOOKDB.CUSTOMER_DIM

Table name  . . . : CUSTOMER_DIM
Table schema  . . : BOOKDB
Server name . . . : BZA1TFIN
Enabled . . . . . : Yes
Archive status  . : A - Active and archived
Remote name . . . : CUSTOMER_DIM-UID_01260003
Remote schema . . : BOOKDB

Details for accelerated table (label): BOOKDB.CUSTOMER_DIM

Table name  . . . : CUSTOMER_DIM
Table schema  . . : BOOKDB
Server name . . . : BZA1STPR
Enabled . . . . . : Yes
Archive status  . : A - Active and archived
Remote name . . . : CUSTOMER_DIM-UID_01260003
Remote schema . . : BOOKDB
```

Figure 7-14 Existing archived partitions for BOOK DB.CUSTOMER_DIM table

The DB2 optimizer does not consider data currency when it routes the query to DB2 Analytics Accelerator, which contains archived partitions. It considers the queue length and capacity to distribute the workload to DB2 Analytics Accelerators.

Figure 7-15 shows the access path of the query when some of the partitions of the table are archived.

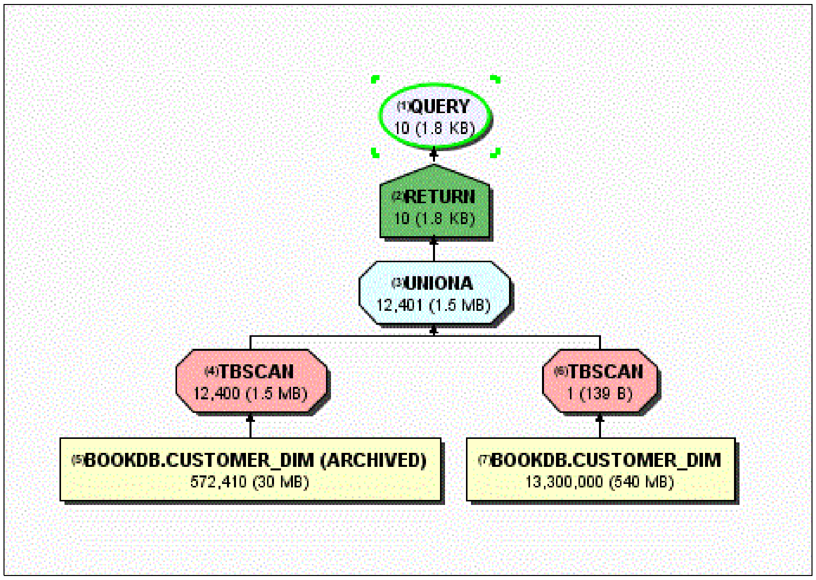


Figure 7-15 Access path of a query when some partitions are archived

7.3 Online restore of archived partitions

The full-scale built-in restore of archived partitions is one of the enhanced features of HPSS. In this section, we describe the step-by-step process of restoring the archived partition from one and two accelerators. We also run an Explain on a query to understand how the access path changes when partitions are archived versus when they are online.

7.3.1 Prepare the restore of archived partitions

The built-in restore feature of HPSS supports restoring the data for one or more explicitly specified partitions, or for the whole table. The HPSS restore can only restore data from a table that has been defined on a DB2 Analytics Accelerator, and that has archived partitions.

> **Important:** If DB2 Analytics Accelerator is unavailable, or if the table is accidentally removed from DB2 Analytics Accelerator without restoring the moved data, you can only recover the DB2 data manually.

To prepare for the restore of an archived partition, there are several things that you need to verify:

- The status of the archived partition is PRO.
- No other utility is running against the table space of the archived partition.
- The table is defined on DB2 Analytics Accelerator.
- DB2 Analytics Accelerator is active (status is online).
- Replication is disabled for the table with the archived partition.

During recovery of moved partitions, the following DB2 utilities are run in the order listed:

1. The RECOVER utility restores the original DB2 data from the image copy that was created at the time that the partitions were moved. The utility works at the partition level. Multiple threads can run in parallel if multiple partitions are specified.
2. The Rebuild index utility rebuilds all of the indexes of the table that were recovered (and are in rebuild-pending state). The utility is run once per table. It rebuilds all data-partitioned secondary index (DPSI) and nonpartitioned secondary index (NPSI) indexes.
3. Check data utility is started automatically for partitions that are in check-pending state after a restore operation. This checks consistency of the restored data.

7.3.2 Restore archived partition on Accelerator 1 (BZA1TFIN)

In this section, we describe the steps to restore the archived partitions of the BOOKDB.CUSTOMER_DIM table:

1. To start the restore process, verify that partitions 96 to 100 of the table are in a PRO state.
2. Use the **DISPLAY DATABASE** DB2 command to verify the table space status, as shown in Figure 7-16.

```
-BZA1DIS DB(BOOKDB) SPACE(TSDC*)
DSNT360I  -BZA1 ***********************************
DSNT361I  -BZA1 *  DISPLAY DATABASE SUMMARY 672
                *    GLOBAL
DSNT360I  -BZA1 ***********************************
DSNT362I  -BZA1     DATABASE = BOOKDB  STATUS = RW 674
                   DBD LENGTH = 88826
DSNT397I  -BZA1 675
NAME     TYPE PART  STATUS            PHYERRLO PHYERRHI CATALOG  PIECE
-------- ---- ----- ----------------- -------- -------- -------- -----
TSDCUST  TS    0001 RW
    -THRU      0095
TSDCUST  TS    0096 RW,PRO
    -THRU      0100
TSDCUST  TS
******* DISPLAY OF DATABASE BOOKDB   ENDED      *********************
```

Figure 7-16 Partitions 96 to 100 are in PRO and no DB2 utility is active on the table space

3. From DB2 Analytics Accelerator Data Studio, connect to the BZA1TFIN accelerator.
4. Expand the Tables section and highlight the BOOK.CUSTOMER_DIM table.
5. Click the down arrow of the Storage Saver box.

6. Select **Restore archived partitions**, as shown in Figure 7-17.

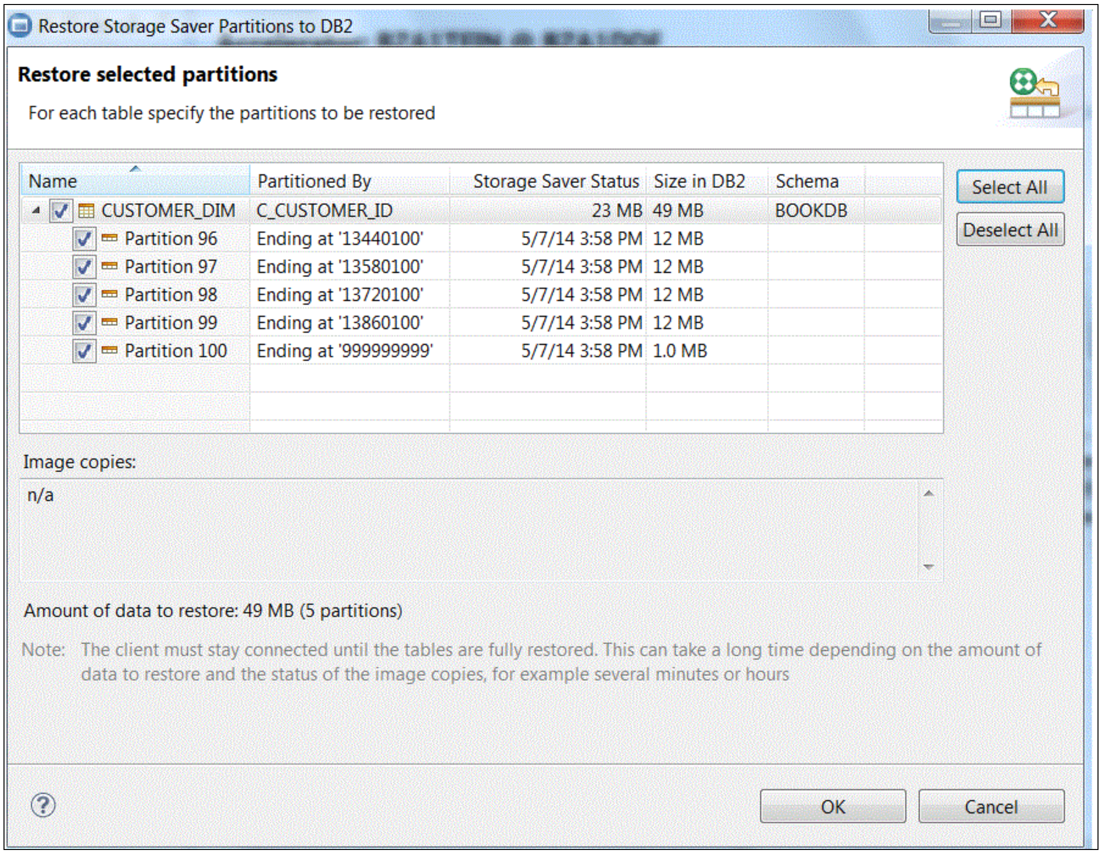


Figure 7-17 Restore partitions

Notice the following output from DB2 Administration Tool, sourced from z/Os, after the restore archived partition is completed, shown in Figure 7-18 on page 156:

– The image copy data set used to recover the partitions.

– The status for partitions 96-100 was updated to RW.

– The archive status of the remote table on BZA1TFIN was updated to blank (not archived).

– The archive status of the remote table on BZA1STPR was updated to X (table was restored on another server).

– The enabled value for the remote table on BZA1TFIN was Y.

– The enabled value for the remote table on BZA1STPR was T, which means that it is in a transition state. Queries are not sent to the DB2 Analytics Accelerator server.

Figure 7-18 shows the output from DB2 Administration Tool.

```
DB2 Admin--- BZA1 Interpretation of an Object in SYSACCELERATEDTABLES
Command ===>

Details for accelerated table (label): BOOKDB.CUSTOMER_DIM

Table name  . . . : CUSTOMER_DIM
Table schema  . . : BOOKDB
Server name . . . : BZA1TFIN
Enabled . . . . . : Yes
Archive status  . : Blank - Not archived
Remote name . . . : CUSTOMER_DIM-UID_01260003
Remote schema . . : BOOKDB

Details for accelerated table (label): BOOKDB.CUSTOMER_DIM

Table name  . . . : CUSTOMER_DIM
Table schema  . . : BOOKDB
Server name . . . : BZA1STPR
Enabled . . . . . : T
Archive status  . : X
Remote name . . . : CUSTOMER_DIM-UID_01260003
```

Figure 7-18 Displaying objects from DB2 Administration Tool

7. Run the query against the BOOKDB.CUSTOMER_DIM table. Although data has been restored on the BZA1TFIN accelerator, DB2 still chooses to route the query to the BZA1STPR accelerator, where some partitions are archived. This shows that DB2 evaluates the request and determines the workload routing based on queue length and capacity, and not on data currency, as shown in Figure 7-19.

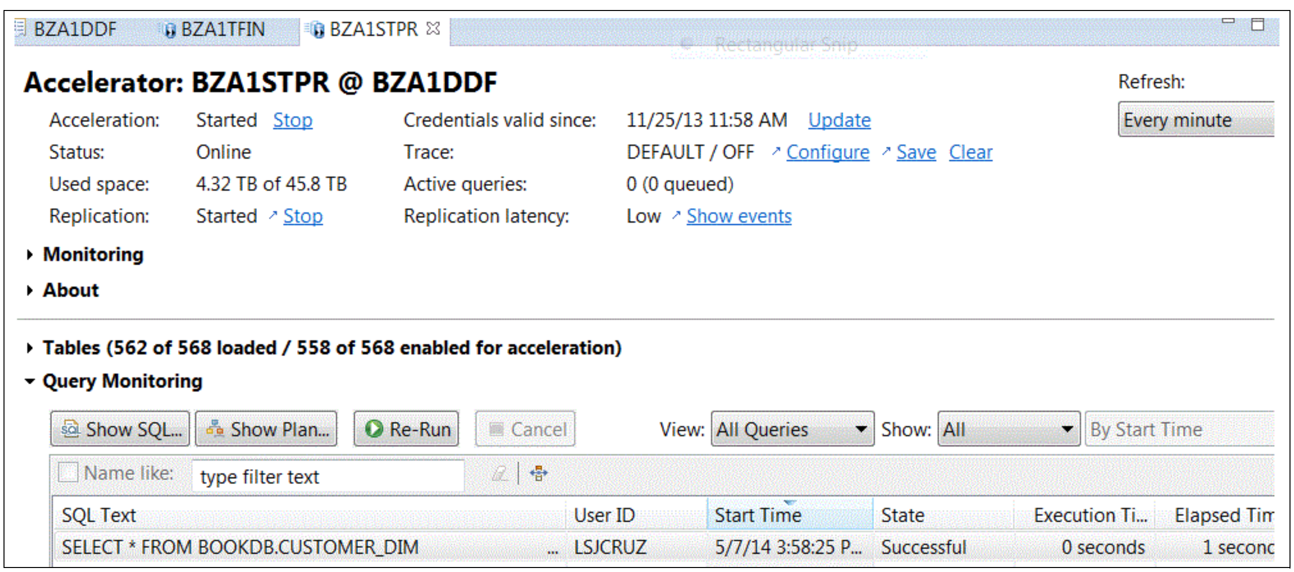


Figure 7-19 Routing the query based on qlength and capacity

7.3.3 Restore archived partition on Accelerator 2 (BZA1STPR)

The built-in HPSS restore enables you to archive partitions in multiple DB2 Analytics Accelerators. These archived partitions should be the same across multiple accelerator servers. The archived partitions can be restored from one accelerator only.

In our scenario, we archived partitions 96 to 100 on both accelerators, BZA1TFIN and BZA1STPR. Then we restored all five (5) partitions from one accelerator, BZA1TFIN. Upon completion of the successful restore process, DB2 updates the archived partitions status to RW, preventing the restore of the same partitions from other accelerators.

After restore, the remote table status is set to *T (transition state)* on the other accelerators**.** This status is an indicator to DB2 that the table was restored on another accelerator, and that this remote table is not enabled for accelerated queries.

If you try to restore the same partitions on a DB2 Analytics Accelerator server, after they have been previously restored from another, you get an error. This error is shown in Figure 7-20.

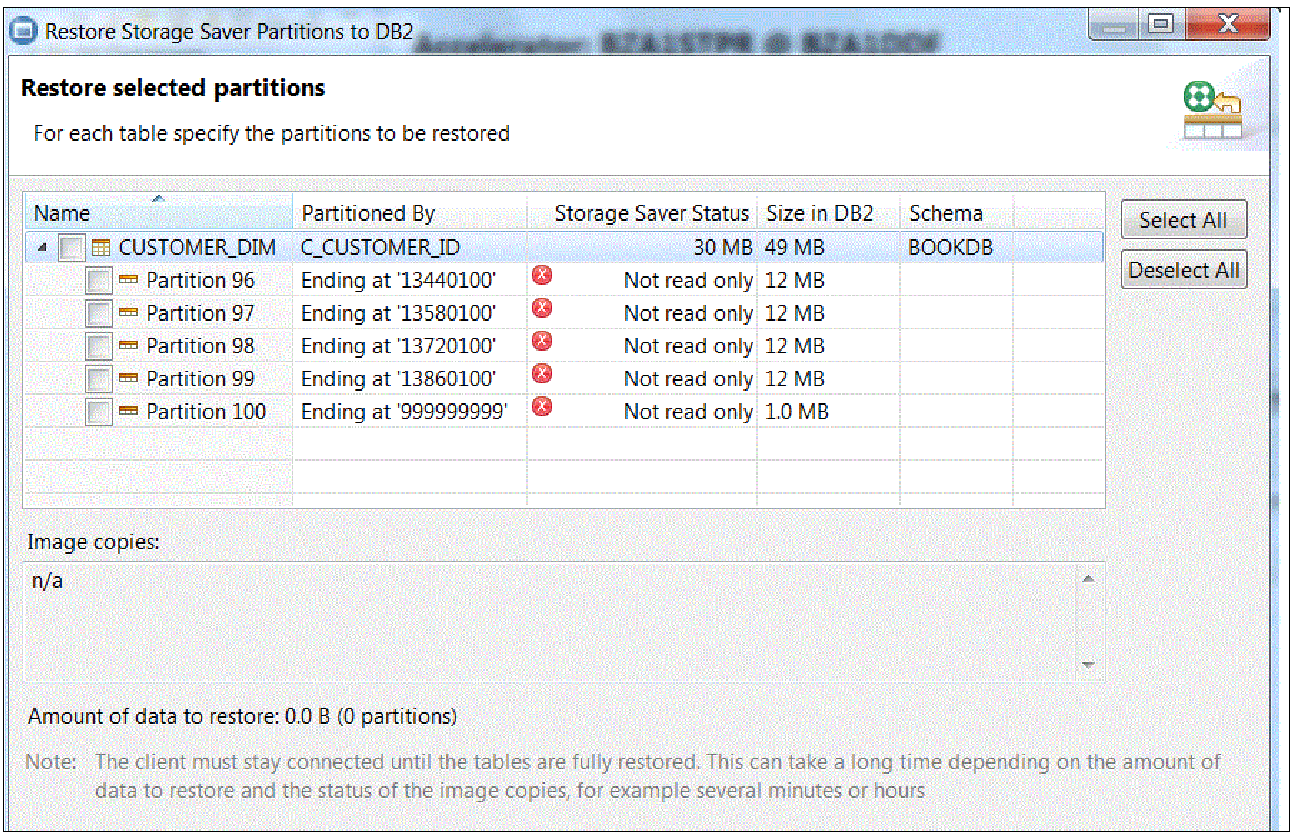


Figure 7-20 Error on Restore when partitions were already restored from another accelerator

If you choose to restore archived partitions that have been restored from another accelerator, you get the error message shown in Figure 7-21.

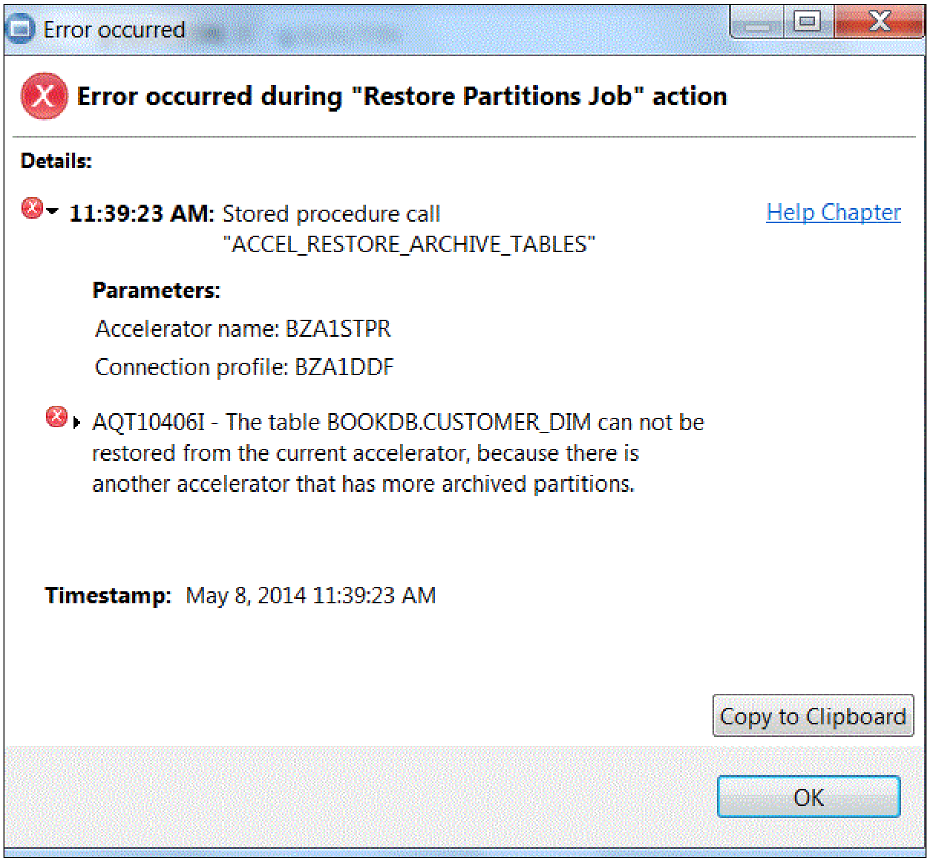


Figure 7-21 Error message on restore

8

High availability, disaster recovery, and workload balancing

In this chapter we provide information about high availability (HA), disaster recovery (DR), and workload balancing (WLB) concepts relative to multiple IBM DB2 Analytics Accelerator for z/OS (DB2 Analytics Accelerator) appliances.

This chapter should be read in conjunction with *IBM DB2 Analytics Accelerator: High Availability and Disaster Recovery*, REDP-5104 for basic concepts about HA, DR, and WLB, and how DB2 Analytics Accelerator can be configured to complement existing IBM System z topologies for HA and DR.

In this chapter we also provide results and observations from experiments examining some of the DB2 Analytics Accelerator WLB, HA, and DR functions.

Although we did not have the luxury of a complete DR infrastructure, such as IBM Geographically Dispersed Parallel Sysplex™ (IBM GDPS®) and Metro Mirror (Peer-to-Peer Remote Copy, PPRC), we examined some of these concepts in our two-system z/OS Sysplex DB2 data sharing environments, with multiple DB2 Analytics Accelerators.

We conducted experiments covering the following:

- High availability of DB2 Analytics Accelerators
- Disaster recovery of DB2 Analytics Accelerator
- Environment
- Basic DB2 Analytics Accelerator workload balancing
- High availability behavior with workload balancing
- Built-in HA DB2 Analytics Accelerator host server
- Remote DB2 Analytics Accelerator experiments

8.1 High availability of DB2 Analytics Accelerators

The primary objective of an HA solution from a DB2 Analytics Accelerator standpoint is to minimize the effect of failures of components (or the entire DB2 Analytics Accelerator itself) while the DB2 for z/OS environment is available. HA addresses the ability of a DB2 Analytics Accelerator solution to endure failure of part or all of its components:

- The private network interfaces
- System Modification Program (SMP) host
- S-Blade
- Redundant Array of Independent Disks (RAID) storage
- The entire DB2 Analytics Accelerator appliance

HA is usually described through service level agreements (SLAs), and achieved through architectural design. In the context of accelerators, HA has a different meaning than that of the conventional DB2 for z/OS databases. With accelerators, failure means acceleration is not possible, but the operational data might still be available in DB2 for z/OS. When mission-critical queries are required to be accelerated, then the availability of DB2 Analytics Accelerator becomes a critical requirement for the business.

DB2 Analytics Accelerator consists of multiple components that contribute to HA, and that are inherited from the underlying IBM PureData System for Analytics (formerly IBM Netezza) architecture, as shown in Figure 8-1. The following list describes these components:

- Netezza Performance Server (NPS) hosts, known as *SMP hosts*
- S-Blades
- RAID storage

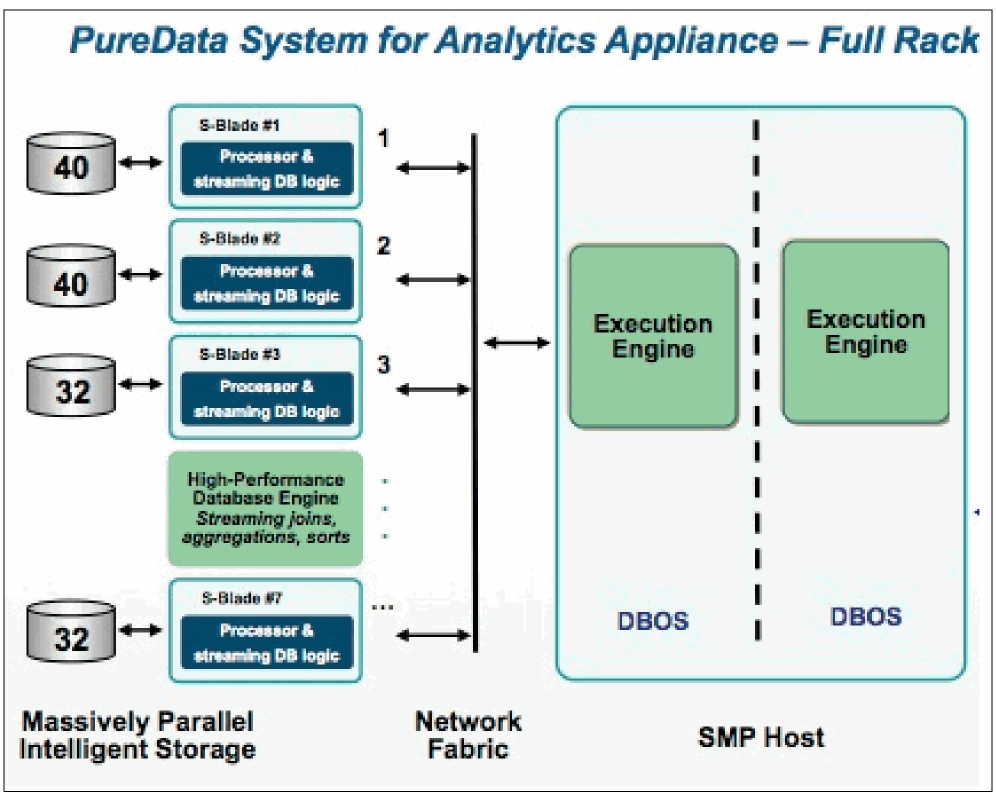


Figure 8-1 PureData System for Analytics N2001-010 model architecture with built-in HA

The redundancy and excess capacity in the internal design of the DB2 Analytics Accelerator, and the dual-port 10 Gigabit Ethernet (10 GbE) interfaces used on the private network, ensure HA of the DB2 Analytics Accelerator solution. This section also describes HA configuration to quickly forego a failed DB2 Analytics Accelerator and switch to a surviving accelerator for continuous query acceleration in the event of total failure.

8.1.1 Redundant SMP host for high availability

Each DB2 Analytics Accelerator appliance is equipped with two SMP hosts. The SMP host acts as the interface with System z. One SMP host is always active, and the other host is in stand-by mode to take over if the first host fails.

The experimental scenario of simulating the failure of an SMP host is discussed in 8.6, “Built-in HA DB2 Analytics Accelerator host server” on page 178.

8.1.2 Redundant S-Blade capacity for high availability

The PureData System for Analytics N2001-010 model has seven S-Blades, and each S-Blade is sized to be able to handle data from up to 40 disks, 16 central processing unit (CPU) cores, and 16 Field Programmable Gate Array (FPGA) engines, as shown in the architecture diagram in Figure 8-1 on page 160.

There should be no significant performance loss when running with one failed S-Blade. This is because the way that the disks are allocated across the S-Blades (2 S-Blades with 40 disks each, and 5 S-Blades with 32 disks each) is specifically designed to enable continuity. For example, after the failure of one S-Blade, all of the 6 surviving S-Blade will be handling data from 40 disks. This is accomplished by internal re-assignment of disks from the failed S-Blade to the surviving S-Blades without any manual intervention.

8.1.3 Redundant array of independent disks

Each DB2 Analytics Accelerator is equipped with multiple disk drives that use RAID technology, which enables data mirroring. Therefore, if one disk fails, the data is preserved. Out of the 288 disks per rack on a PureData for Analytics N2001-10 model, 240 are active and assigned to S-Blades, 34 are spare, and 14 are used actively for swap and log space.

The spare disks provide another layer of HA, because they are automatically reconfigured (in the background) so that DB2 Analytics Accelerator can continue to operate seamlessly after a disk failure.

8.1.4 Private network configuration for high availability

The suggested HA network setup uses two Open Systems Adapter (OSA) Express (OSA-Express) cards, two switches, and the required number of additional cables. In such an environment, the links between the switches ensure that a path from System z to the surviving host can always be found.

If one switch fails, incoming signals are routed through the other switch, which can also connect to both SMP hosts, as shown in Figure 8-2.

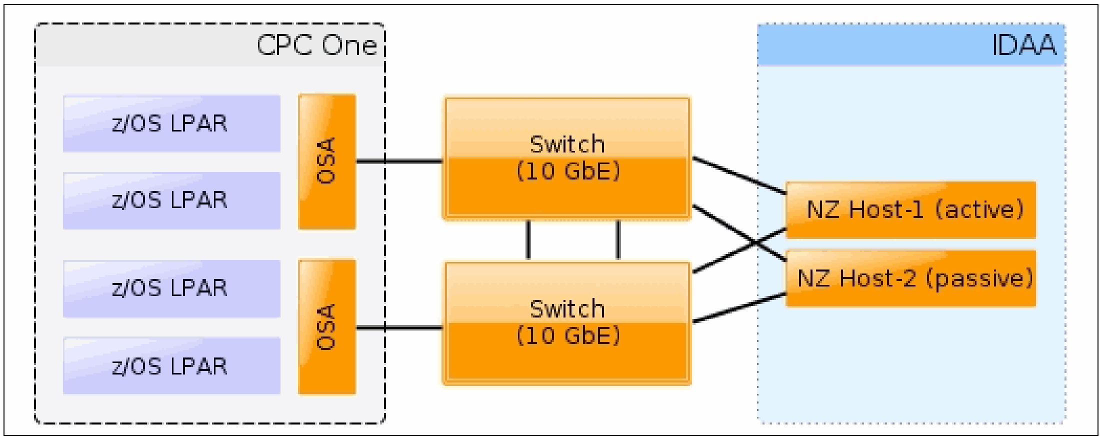


Figure 8-2 Suggested high availability network configuration

Additional central processor complexes (CPCs, formerly known as CECs) can be easily integrated into Figure 8-2. For example, a sample HA network configuration with two CPCs and two DB2 Analytics Accelerators is shown in Figure 8-3. This configuration can be thought of as an extension to the DB2 for z/OS DR configuration (before DB2 Analytics Accelerator was introduced). The second accelerator (labeled *HA for Production Accel*) would take over the entire workload when the production DB2 Analytics Accelerator fails.

This configuration is further discussed in section 8.1.5, "High availability configuration to handle failure of an entire accelerator" on page 163.

The terminology used in the network setup shown in Figure 8-2, and further information about HA network setups, is available in the IBM white paper *Network Configurations for IBM DB2 Analytics Accelerator for z/OS*:

http://www.ibm.com/support/docview.wss?uid=swg27028171

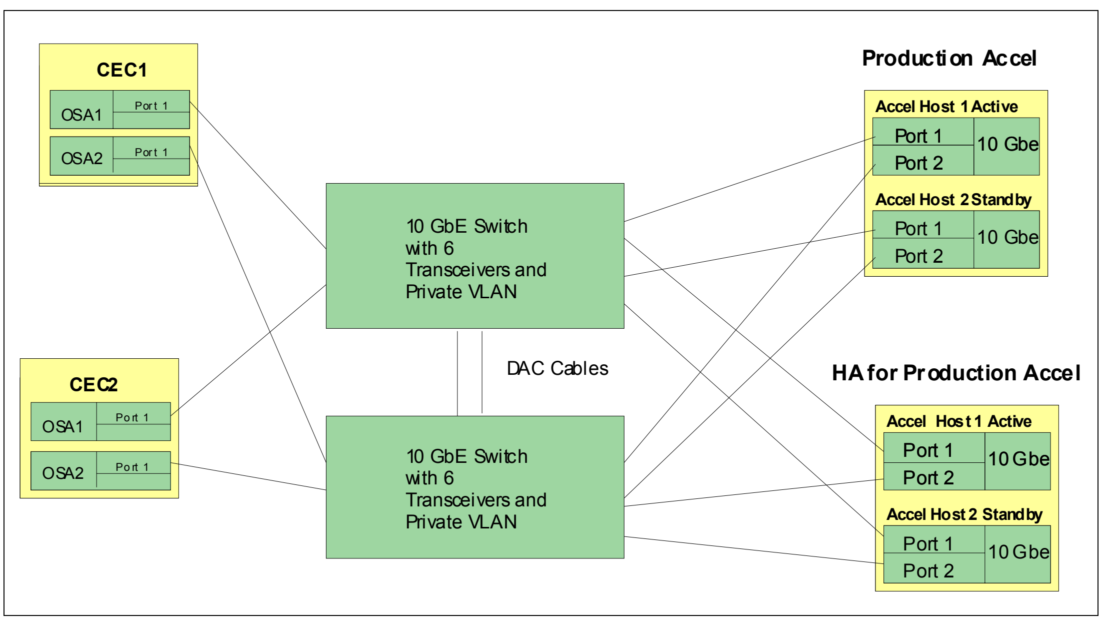


Figure 8-3 Sample HA network configuration: Two CPCs and two DB2 Analytics Accelerators

8.1.5 High availability configuration to handle failure of an entire accelerator

The HA network topology with two DB2 Analytics Accelerators to deal with the total failure of one accelerator is shown in Figure 8-4. The network topology for both data-sharing (right side of Figure 8-4) and non-data-sharing environments is basically the same. In this configuration, additional cables and switches can be added for more redundancy (if required).

In addition to the redundancy to handle failure of network components, such as OSA card, cable, switch, and network card, the configuration depicted in Figure 8-4 also addresses failure of an entire DB2 Analytics Accelerator box. This is accomplished with the help of the heartbeat signal (happening every 20 seconds) between DB2 for z/OS and the DB2 Analytics Accelerator.

> **Note:** Two links from the switch to DB2 Analytics Accelerator are mandatory to support failover to a standby host.

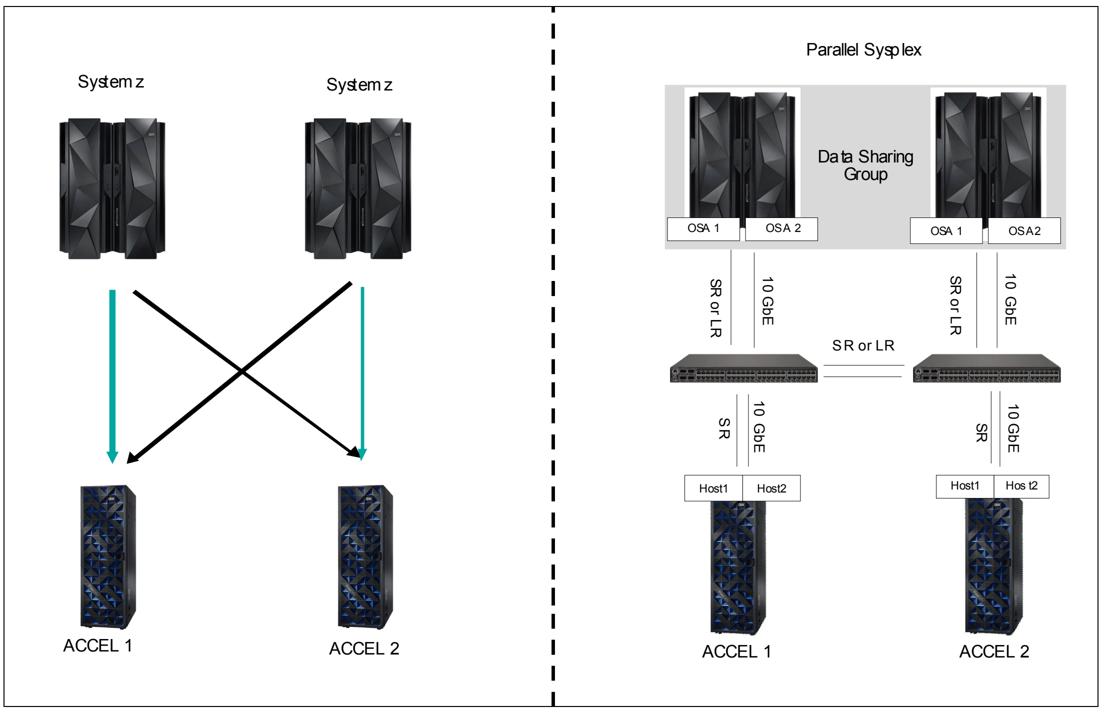


Figure 8-4 High availability configuration with two accelerators

After a complete failure of one of the DB2 Analytics Accelerators, the existence of the heartbeat between DB2 for z/OS and DB2 Analytics Accelerator is used to determine which accelerator is still active, and in turn to route the subsequent queries to that accelerator.

In this HA configuration with two DB2 Analytics Accelerators, when both DB2 Analytics Accelerators are active and functioning correctly, WLB can be achieved if all of the tables are loaded and enabled for acceleration. WLB is discussed in detail in section 8.4, "Basic DB2 Analytics Accelerator workload balancing" on page 168.

Data maintenance considerations for high availability configurations

The data maintenance strategy should be based on your recovery time objective (RTO) and recovery point objective (RPO) requirements. You can follow one of the three options for HA with reference to the sample configuration shown on the left side of Figure 8-4:

Active-standby accelerators Load and use only ACCEL1 during normal operation. Load and use ACCEL2 only when ACCEL1 is down. If WLB is not needed, this configuration can work well.

Active-active accelerators When loading data to DB2 Analytics Accelerator, always load it twice (into ACCEL1 *and* ACCEL2). This option results in a balanced usage of both DB2 Analytics Accelerators, and the best possible response times by even workload distribution during normal operation.

Mix-and-match Load important tables into both ACCEL1 and ACCEL2. After detecting failure of the primary DB2 Analytics Accelerator (by heartbeat), load the remaining tables and accelerate the queries.

The load operation for each DB2 Analytics Accelerator can be run from any member of the data sharing group. The tables are enabled after successful load to make them readily available when needed. Incremental update (IU), or InfoSphere Change Data Capture (InfoSphere CDC), can also be used to update tables on both ACCEL1 and ACCEL2 to reduce the latency on the primary accelerator and to reduce the recovery time while switching to the standby accelerator.

Other considerations (including the DB2 Analytics Accelerator High Performance Storage Saver (HPSS) feature) regarding data maintenance strategies are discussed in *IBM DB2 Analytics Accelerator: High Availability and Disaster Recovery*, REDP-5104:

http://www.redbooks.ibm.com/abstracts/redp5104.html?Open

If the DB2 data sharing environment spans multiple logical partitions (LPARs), in case of failure of one LPAR, the CDC agent on the other LPAR can start and continue operation. The configuration details are documented in the *High Availability Setup for the Incremental Update Capture Engine* Product Documentation:

http://www.ibm.com/support/docview.wss?uid=swg27037912

8.2 Disaster recovery of DB2 Analytics Accelerator

DR addresses the readiness to re-establish availability of a DB2 Analytics Accelerator after a data center outage. It refers to restoring all of the associated systems and data to a previous acceptable state in the event of partial or complete failure of an DB2 Analytics Accelerator environment. The possible failure includes the DB2 for z/OS environment, due to man-made or natural causes.

Recovery is the most critical part of a disaster. The first major question to answer in DR planning is, “How much data loss and downtime are acceptable in a disaster?” These metrics must be derived from the business decisions made, based on the options available and relative expenses.

Table 1 (Summary of options) in the IBM Redpaper™ publication titled *IBM DB2 Analytics Accelerator: High Availability and Disaster Recovery*, REDP-5104, summarizes different options. There are options for various RTO and RPO characteristics and for different (existing) DB2 for z/OS DR options, which could help business leaders make appropriate decisions.

Similar to how a good backup strategy is important for a successful recovery of DB2 for z/OS databases, the DB2 Analytics Accelerator's recovery relies on proper data maintenance and synchronization strategies. Data synchronization options are discussed in section 8.2.3, "Options for DB2 Analytics Accelerator recovery" on page 166.

8.2.1 Rolling migration in a HA/DR environment configuration

We have introduced rolling migration in 4.1.5, "Rolling migration in an HA and disaster recovery environment" on page 57. Figure 8-5 shows a sample DB2 Analytics Accelerator DR scheme that would support a typical DR configuration of a DB2 for z/OS environment. This schematic also addresses the HA and WLB needs.

In this scheme, it is assumed that the DB2 data sets are asynchronously replicated by GDPS to the DR site (passive site) using PPRC or IBM z/OS Global Mirror (zGM, formerly Extended Remote Copy, or XRC). When DB2 at the passive site comes up (as part of the recovery process), it would have the same subsystem identifier (SSID), the same catalog tables (including the authentication token for DB2 Analytics Accelerator), and the same logs as the active site.

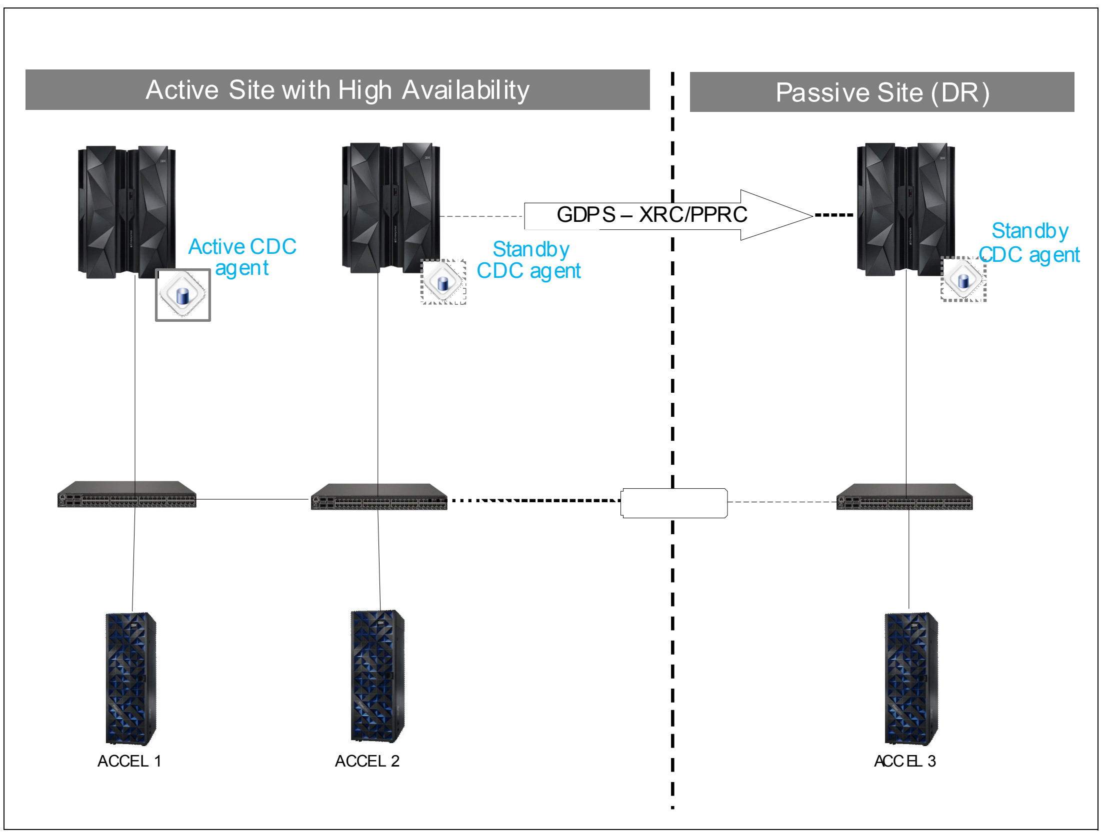


Figure 8-5 Configuration for HA/DR with incremental update and WLB options

8.2.2 Incremental update considerations

Dynamic virtual Internet Protocol (IP) addresses (DVIPAs) for the InfoSphere CDC agent on z/OS are used. The active InfoSphere CDC agent runs on (any) one of the DB2 members of the data sharing group, and “hot-standby” InfoSphere CDC agents exist on other members. In case of failure, the InfoSphere CDC agent on the surviving LPAR automatically continues the incremental update operation to ensure an HA scenario, as discussed in “Data maintenance considerations for high availability configurations” on page 164.

If replication from and to the DR site is possible (sufficient bandwidth exists), an InfoSphere CDC agent on the DR site continues after DB2 has been started. The right side of Figure 8-5 on page 165 shows a typical DR scheme at a passive site.

8.2.3 Options for DB2 Analytics Accelerator recovery

Depending on your data synchronization strategy, DR can be carried out using one of the following options:

- Reload data from DB2 into ACCEL3 at about 1 terabyte per hour (TB/h). In this case, ACCEL3 might also be down before activation.
- Regularly load data from the active site to ACCEL3, and reload the changed parts only after DB2 is running again (partition update, re-sync since last refresh).
- Keep ACCEL3 in sync with DB2 using incremental update (starting the capture agent after DB2 is active).

8.3 Environment

The experiments in this chapter were conducted on a two-system sysplex, two-way DB2 data sharing environment connected to two accelerators, all in Poughkeepsie, NY. Table 8-1 provides the key hardware and software levels used.

Table 8-1 Environment for WLB/HA experiments

Hardware		
Server/Appliance	**Model**	**Detail**
System z	IBM zEnterprise EC12 (zEC12) Model 2827-739	LPAR P59 - 2 CPs, 1 zIIP LPAR P60 - 2 CPs, 1 zIIP
Accelerators	PureData System for Analytics N1001-010	Name: BZAGTFIN Installed software: ► Accelerator server: 4.1.0.201311141445 ► NPS 7.0.2(P6) [Build 32037] ► IBM Netezza Firmware Diagnostics and Tools (FDT) 2.6 ► IBM Netezza Host Platform (HPF) 5.2 ► Access Server 10.2.1.2220 ► Replication Engine 10.2.1 [Build CCTRJYPP_20_4]
	PureData System for Analytics N2001-010	Name: BZAGSTPR Installed software: ► Accelerator server: 4.1.0.201311141445 ► NPS 7.0.2(P9) [Build 33062] ► FDT 3.0.5.1 ► HPF 5.3.2 ► Access Server 10.2.1.2220 ► Replication Engine 10.2.1[Build CCTRJYPP_20_4]
Software		
Product	**Version / Release**	**Detail**
z/OS	2.1	Both LPARs
DB2	11	Both LPARs
IBM InfoSphere Data Replication CDC	10.2.1	LPAR P59
IBM Tivoli OMEGAMON XE for DB2 Performance Expert (PE) on z/OS	5.2.0	Both LPARs
IBM DB2 Analytics Accelerator for z/OS	4.1 program temporary fix 2 (PTF2)	DB2 Analytics Accelerator stored procedures: 4.1.0.2013111-1826 (both LPARs)

In addition to this, for our remote accelerator experiments, we also used a third DB2 Analytics Accelerator V4.1 appliance (another PureData for Analytics N1001-010) in San Jose, California.

Figure 8-6 is the HA/DR diagram of the environment used in various experiments described in this section.

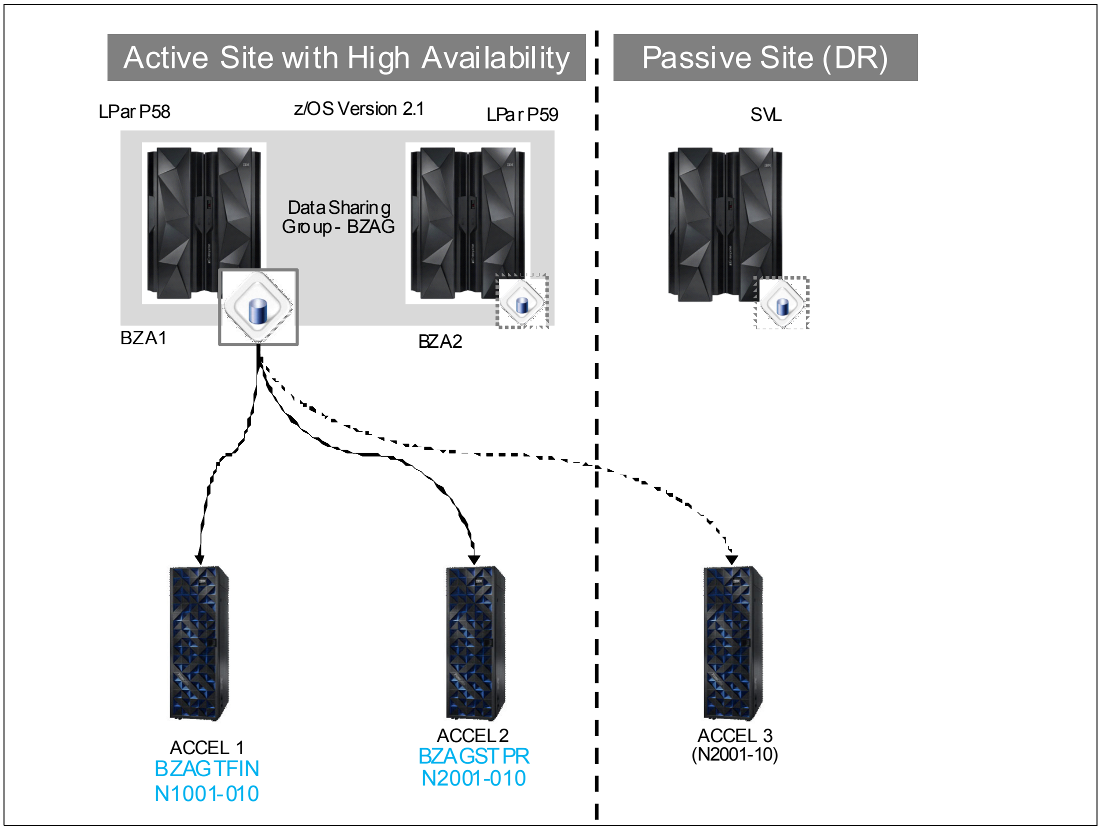


Figure 8-6 High availability and disaster recovery configuration used for our scenarios

8.4 Basic DB2 Analytics Accelerator workload balancing

This section describes what WLB is, and exhibits results from a potential client scenario implementing WLB.

8.4.1 Workload balancing

The objective of WLB is to intelligently distribute the accelerated query workload across all qualified accelerators. A DB2 Analytics Accelerator is *qualified* if it contains the data required by a given query, and it is running V4.1 or later. The workload balancing algorithm intelligence is based on the capacity and utilization of all of the qualified accelerators.

Accelerator capacity is determined by the number of data slices for a given DB2 Analytics Accelerator. A *data slice* contains pieces of each accelerated table. Generally, the more data slices, the more DB2 Analytics Accelerator parallel processing power. The number of data slices for a given accelerator can be found in DB2's instrumentation facility component identifier (IFCID) 002, field name Q8STNMDS.

Accelerator utilization is the number of queries concurrently running on DB2 Analytics Accelerator. This includes queries that are queued, running, and fetching. The number of queries concurrently running is captured in DB2's IFCID 002, field name Q8STACTV.

It is important to note that *accelerator utilization*, as used for the WLB algorithm, does not include accelerator CPU use of worker nodes, coordinator nodes, or disk input/output (I/O) usage. See Figure 8-7 for how these fields are displayed in the DB2 Analytics Accelerator section of an OMEGAMON XE for DB2 PE Statistics Long report, when running Version 4 of DB2 Analytics Accelerator.

```
BZAGSTPR         FOR SUBSYSTEM ONLY                 QUANTITY    BZAGSTPR         TOTAL ACCELERATOR                  QUANTITY
-----------------------------------  --------------------    -----------------------------------  --------------------
QUERIES SUCCESSFULLY run                         5320.00    QUERIES SUCCESSFULLY run                         5320.00
QUERIES FAILED TO run                               0.00    QUERIES FAILED TO run                               0.00
CURRENTLY running QUERIES                           2.61    CURRENTLY running QUERIES                           2.61
MAXIMUM running QUERIES                            37.00    MAXIMUM running QUERIES                            37.00
CPU TIME running QUERIES                    55:22.590000    CPU TIME running QUERIES                    55:22.990000
CPU TIME LOAD/ARCHIVE/RESTORE                   0.000000    CPU TIME LOAD/ARCHIVE/RESTORE                   0.000000

CONNECTS TO ACCELERATOR                          6075.00    DISK STORAGE AVAILABLE (MB)                  48000959.98
REQUESTS SENT TO ACCELERATOR                    11545.00      IN USE FOR ACCEL DB - ALL DB2 (MB)          9468592.04
  TIMED OUT                                         0.00      IN USE FOR ACCEL DB - THIS DB2(MB)          4704865.04
  FAILED                                            0.00
BYTES SENT TO ACCELERATOR                    40943575.00    MAXIMUM QUEUE LENGTH                               26.00
BYTES RECEIVED FROM ACCELERATOR             268174387.00    CURRENT QUEUE LENGTH                                0.00
MESSAGES SENT TO ACCELERATOR                    59425.00    AVG QUEUE WAIT ELAPSED TIME                     0.030067
MESSAGES RECEIVED FROM ACCEL                    59425.00    MAX QUEUE WAIT ELAPSED TIME                 13:47.500427
BLOCKS SENT TO ACCELERATOR                          0.00
BLOCKS RECEIVED FROM ACCELERATOR                  905.00    WORKER NODES                                        7.00
ROWS SENT TO ACCELERATOR                            0.00    WORKER NODES DISK UTILIZATION (%)                  18.67
ROWS RECEIVED FROM ACCELERATOR                      0.00    WORKER NODES AVG CPU UTILIZATION (%)               32.41
                                                            COORDINATOR CPU UTILIZATION (%)                    12.42
TCP/IP SERVICES ELAPSED TIME                5:26:26.528310    PROCESSORS                                       224.00
ELAPSED TIME IN ACCELERATOR                12:52:20.659574    DATA SLICES                                      240.00
WAIT TIME IN ACCELERATOR                      1:19.042339

CPU TIME FOR REPLICATION                        3.624823    CPU TIME FOR REPLICATION                    37:13.828838
LOG RECORDS READ                                  248.00    LOG RECORDS READ                             84377596.74
  LOG RECORDS FOR ACCEL TABLES                    248.00      LOG RECORDS FOR ACCEL TABLES               84377596.74
LOG RECORD BYTES PROCESSED                      16368.00    LOG RECORD BYTES PROCESSED                17775885165.15
INSERT ROWS FOR ACCEL TABLES                        0.00    INSERT ROWS FOR ACCEL TABLES                        0.00
UPDATE ROWS FOR ACCEL TABLES                        0.00    UPDATE ROWS FOR ACCEL TABLES                 43646764.98
DELETE ROWS FOR ACCEL TABLES                        0.00    DELETE ROWS FOR ACCEL TABLES                       10.00
REPLICATION LATENCY                             0.171255
REPLICATION STATUS CHANGE              05/17/14 16:27:22.76    ACCELERATOR SERVER START              05/15/14 15:34:37.91
                                                            ACCELERATOR STATUS CHANGE             05/15/14 15:35:01.53
```

Figure 8-7 DB2 Analytics Accelerator data for Version 4 in Statistics Long Report

The following list provides the associated field descriptions, as documented in the `Chap_49_Statistics_Accelerator_Data_V4.pdf` file update that came with PM98092 in December 2013, which is available for download from the following website (scroll down):

http://www-01.ibm.com/support/docview.wss?uid=swg27040297

DATA SLICES

The number of data slices at DB2 Analytics Accelerator. This equals the degree of parallel I/O channels.

Field Name: Q8STNMDS

CURRENTLY running QUERIES

The number of currently (actively) running queries in DB2 Analytics Accelerator on behalf of all DB2 systems (field name for DB2 Analytics Accelerator for z/OS Version 4: Q8STACTV_64).

Field Name: Q8STACTV

Important: In the DB2 Analytic Accelerator version 4 Statistics Long report, OMEGAMON makes it much clearer which accelerator metrics are related to the given DB2 subsystem (left side of report labeled FOR SUBSYSTEM ONLY) versus all DB2 subsystems (right-side of report labeled TOTAL ACCELERATOR).

A quality reference for the current IBM Tivoli OMEGAMON XE for DB2 PE/PM on z/OS can be located on the following website:

http://www-01.ibm.com/support/docview.wss?uid=swg27040297

Based on these two metrics, DB2 will re-calculate a *capacity weight* every 20 seconds, for each connected V4.1 DB2 Analytics Accelerator. In turn, DB2 uses these capacity weights to determine which accelerator to route the next qualified query to.

Additional considerations

If there are less than two qualified V4.1 accelerators (Example: One V4 accelerator and one V3.1 accelerator that both contain the data being queried), the query is routed to the first qualified accelerator listed in the DB2 catalog, and no V4.1 capacity weight intelligence is used to distribute.

For static Structured Query Language (SQL), the opportunity to use WLB occurs at execution time, and at that point WLB behavior is the same as for dynamic SQL.

8.4.2 Workload balancing migration considerations

The naming scheme for tables in the SYSACCEL.SYSACCELERATEDTABLES pseudo catalog table was changed to support WLB. Run the new SYSPROC.ACCEL_MIGRATE stored procedure against each accelerator that is supposed to participate in WLB. See 4.2.4, “Migrating SYSACCELERATEDTABLES table” on page 64 for details.

8.4.3 Workload balancing experiment

We developed three experiments, attempting to somewhat simulate the experience a client with a maturing analytics workload might witness. The premise is starting with a production analytics workload running on a PureData for Analytics N1001 with DB2 Analytics Accelerator. This work now becomes more mission-critical, and it’s expected to grow.

Therefore, a second accelerator is added for capacity, for production and test, but also to provide HA to the production workload. We ran the following three experiments to exhibit the WLB behavior associated with these changes:

Test Case A Single V4 DB2 Analytics Accelerator (N1001), 10 concurrent user queries

Test Case B Multiple V4 DB2 Analytics Accelerators (N1001 and N2001), 10 concurrent user queries

Test Case C Multiple V4 DB2 Analytics Accelerators (N1001 and N2001), 20 concurrent user queries

Before reviewing results, we first explain the query workload we used for these experiments, in addition to most of the experiments in this chapter.

The residency workload balancing query workload

The focus being evaluation of HA and WLB, we used a relatively simple query workload consisting of multiple users running a mix of intermediate or complex queries.

Table 8-2 includes the two query types were chosen for these experiments.

Table 8-2 WLB workload: Queries stand-alone performance on PureData System for Analytics N2001-010

Name	Description	Stand-alone Elapsed Time (seconds)
Q11A	Intermediate complexity: Selective report requiring predicate evaluation over 200 gigabyte (GB) fact table joined with small dimension tables.	2
Q1B	Complex, resource-intensive, non-selective join between 200 GB fact table and small dimension tables.	17

Depending on the experiment, we varied the number of users submitting these queries. The following list describes the basic query workloads:

- WLB_LoadA: 10 simulated concurrent users:
  - 8 users submitting Q11A. Each user submits a query on average every 20 seconds.
  - 2 users submitting Q1B. Each user submits a query on average every 60 seconds.
- WLB_LoadB: 20 simulated concurrent users:
  - 16 users submitting Q11A. Each user submitting the query on average every 20 seconds.
  - 4 users submitting Q1B. Each user submitting the query on average every 60 seconds.

To keep the workload arrival rate somewhat consistent, we set an arrival rate of approximately 20 seconds for Q11A, and 60 seconds for Q1B. Only “somewhat consistent” because, in cases where DB2 Analytics Accelerators were not able to finish the query in the set arrival rate, these users would be delayed in submitting their next query, therefore affecting the arrival rate.

These queries were submitted by distributed data facility (DDF), using a command-line interface (CLI)-based query submission tool running on a Windows server. Although it is not necessarily representative, but to prevent occasional errors with our simulation tool, each simulated user reconnects to DB2 for each query.

Data collection methodology

Standard DB2 accounting and statistics traces were captured for all experiments:

- Accounting 1, 2, 3
- Statistics 1, 3, 4, 5, 6

The trace data was subsequently reduced using Tivoli OMEGAMON for DB2 Performance Expert V520. We predominantly made use of the relatively new capability of saving fields of interest into comma-separated values (CSV) files.

Workload balancing experiment results

We evaluated the results of these experiments from two perspectives.

First, given the WLB factors, we examined how the WLB algorithm distributes the incoming work. In our case, we have heterogeneous accelerators, so they are perhaps even more interesting or challenging to balance. Secondly, we wanted to gain some appreciation from a performance perspective, so we monitored average query response times when we added a second DB2 Analytics Accelerator and increased workload.

Workload balancing distribution of queries

The first factor in determining the capacity weight is *accelerator capacity*. Table 8-3 depicts the workload balancing *accelerator capacity* for the two V4.1 DB2 Analytics Accelerators used for the residency. This factor is constant.

Table 8-3 Workload balancing capacity

WLB capacity (# of data slices)	
N1001-010	**N2001-010**
92	240

Using the other accelerator statistics field, CURRENT QUEUE LENGTH (Q8STCQL), we estimated the following factors:

- Queries actively running/fetching (Q8STACTV - Q8STCQL)
- Queries currently queued on DB2 Analytics Accelerator (Q8STCQL)

  Q8STCQL is a new field captured using IFCID 002 with V41 of DB2 Analytics Accelerator. Q8STCQL is a *snapshot* value, where Q8STACTV is an *aggregated* value, so our previous calculation of queries actively running/fetching is not exact, but good enough to estimate queuing versus running/fetching activity on DB2 Analytics Accelerator.

In addition to *accelerator capacity* and *accelerator utilization*, we also monitored the number of connections per accelerator, per minute. This allowed us to see how the queries were getting distributed relative to the WLB factors. For connections per accelerator per minute, we summarized the field Q8STCONN (which is a per-DB2-subsystem metric), for both of our DB2 subsystems. See Figure 8-8 for a Statistics Long report exhibiting these additional two fields.

```
BZAGSTPR        FOR SUBSYSTEM ONLY                 QUANTITY    BZAGSTPR        TOTAL ACCELERATOR                 QUANTITY
-----------------------------------  --------------------    -----------------------------------  --------------------
QUERIES SUCCESSFULLY run                        5320.00    QUERIES SUCCESSFULLY run                        5320.00
QUERIES FAILED TO run                              0.00    QUERIES FAILED TO run                              0.00
CURRENTLY running QUERIES                          2.61    CURRENTLY running QUERIES                          2.61
MAXIMUM running QUERIES                           37.00    MAXIMUM running QUERIES                           37.00
CPU TIME running QUERIES                   55:22.590000    CPU TIME running QUERIES                   55:22.990000
CPU TIME LOAD/ARCHIVE/RESTORE                  0.000000    CPU TIME LOAD/ARCHIVE/RESTORE                  0.000000

CONNECTS TO ACCELERATOR                         6075.00    DISK STORAGE AVAILABLE (MB)                 48000959.98
REQUESTS SENT TO ACCELERATOR                   11545.00      IN USE FOR ACCEL DB - ALL DB2 (MB)         9468592.04
  TIMED OUT                                        0.00      IN USE FOR ACCEL DB - THIS DB2(MB)         4704865.04
  FAILED                                           0.00
BYTES SENT TO ACCELERATOR                   40943575.00    MAXIMUM QUEUE LENGTH                              26.00
BYTES RECEIVED FROM ACCELERATOR            268174387.00    CURRENT QUEUE LENGTH                               0.00
MESSAGES SENT TO ACCELERATOR                   59425.00    AVG QUEUE WAIT ELAPSED TIME                    0.030067
MESSAGES RECEIVED FROM ACCEL                   59425.00    MAX QUEUE WAIT ELAPSED TIME                13:47.500427
BLOCKS SENT TO ACCELERATOR                         0.00
BLOCKS RECEIVED FROM ACCELERATOR                 905.00    WORKER NODES                                       7.00
ROWS SENT TO ACCELERATOR                           0.00    WORKER NODES DISK UTILIZATION (%)                 18.67
ROWS RECEIVED FROM ACCELERATOR                     0.00    WORKER NODES AVG CPU UTILIZATION (%)              32.41
                                                           COORDINATOR CPU UTILIZATION (%)                   12.42
```

Figure 8-8 DB2 Analytics Accelerator data for V4 from a Statistics Long report

In the report shown in Figure 8-8 on page 172, the numbers represent the following data:

- CONNECTS TO ACCELERATOR
  - The number of connects to DB2 Analytics Accelerator from this DB2 system
  - Field Name: Q8STCONN
- CURRENT QUEUE LENGTH
  - The current query queue length at DB2 Analytics Accelerator
  - Field Name: Q8STCQL

In Figure 8-9, we plot these key metrics as we moved through the three test scenarios mentioned previously (A, B, and C).

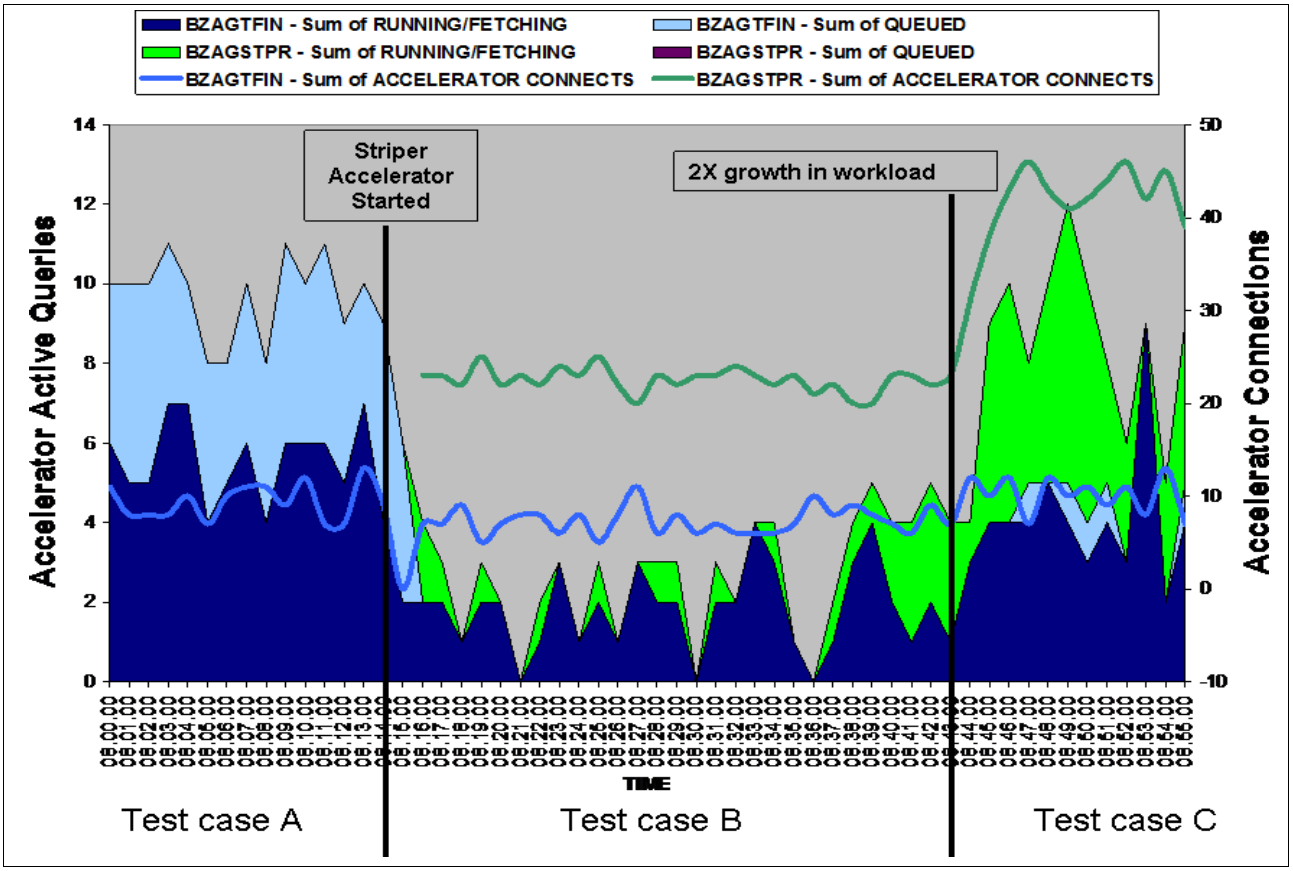


Figure 8-9 Accelerator usage and associated distribution of new connections

Observations and analysis

The following results were observed for the test cases:

TestCase A This is our base workload running with just the BZAGTFIN accelerator. Viewing the area plot against the left y-axis, we notice on average that there are approximately 12 - 14 active queries, of which approximately 4 queries are queued. Viewing the line plot (right y-axis), we notice on average about 10 connections per minute to the BZAGTFIN, the only DB2 Analytics Accelerator started.

TestCase B The BZAGSTPR accelerator is now added, and in a couple minutes we notice a significant drop in the average number of active queries on the BZAGTFIN, and only 1 - 2 on the BZAGSTPR. The queries are now getting in and out of the system quite quickly. Therefore, the average number of active queries is significantly lower. Additionally, we notice the we no longer have queries queued on the BZAGTFIN.

Viewing the number of connections line plot, as anticipated, we notice that the number of BZAGSTPR connections per minute is about 2 - 3 times that of BZAGTFIN.

TestCase C The number of users is now doubled. As expected, we notice a jump in the average number of active queries for both accelerators. With this heavier workload, we notice the BZA1STPR connections are now closer to 4 times the BZAGTFIN.

Comparing the average number of active queries for both accelerators, this distribution would appear to make good sense. We also notice some minor queuing starting again on the BZAGTFIN.

In summary, we were generally pleased with what we witnessed.

Workload balancing performance effect with a second accelerator

We now examine what occurred with regard to average query response times for the same three test scenarios described. Table 8-4 exhibits the performance results.

Table 8-4 Performance statistics of the workload balancing tests

Test scenario	Qualified accelerators	Users	Performance			
			Average response time (seconds)		Completions per minute	
			Q11A	Q1B	Q11A	Q1B
A	N1001	10	59	96	8	1.2
B	N1001 N2001	10	5	30	27	2.3
C	N1001 N2001	20	5	54	71	3.5

Observations and analysis

A significant, even more-than-anticipated drop in average response time was witnessed when adding the BZAGSTRP accelerator. We theorize that this was due to the ability of model N2001 (BZAGSTPR) to handle concurrency much better than model N1001 (BZAGTFIN).

We had a challenging time trying to understand how we could get a rate of 71 Q11A query completions per minute when we doubled the workload. Based on the fact that four out of every five connections now go to the PureData for Analytics N2001 accelerator, we theorized that this was probably due to the fact that most of our Q11A queries were now getting faster response times, leading to faster arrivals of the next Q11A query. This is likely due to the nature of our workload, and should not be taken forward as a normal expectation.

In summary, we were pleased with the distribution of the query workload. A good indicator was the fact that model N2001 had 2 - 3 times the number of connections than that of model N1001. This was consistent with other performance measurements conducted, which showed model N2001 providing 2 - 3 times X better throughput performance than model N1001.

8.5 High availability behavior with workload balancing

This section describes two aspects of HA associated with WLB.

8.5.1 High availability with query workload balancing experiments

We devised the following two experiments to better appreciate DB2 Analytics Accelerator HA by WLB:

- A *rolling accelerator upgrade migration* scenario
- An *accelerator failure* scenario

> **Important:** When running our first experiment, upon issuing a DB2 `STOP ACCEL` command, we encountered what appeared to be a hang relative to any additional accelerator commands we issued. We then applied authorized program analysis reports (APARs) PI09807 (PTF UI16997) and PI10774 (PTF UI15444), both having to do with accelerator STOP issues. These resolved the hang condition for our experiments.

For these experiments, we ran the “The residency workload balancing query workload” on page 171. As one would do with a rolling accelerator upgrade, we first do a controlled STOP of one of our accelerators. Then we START DB2 Analytics Accelerator again. Secondly, simulating an un-planned Accelerator failure, we force a STOP of one of our accelerators. The following list describes the specific steps performed and commands issued:

1. Start both accelerators to all members in the data sharing group.

   ```
   -BZA1 STA ACCEL(*) SCOPE(GROUP)
   ```

2. Start the WLB_LoadA query workload using the CLI workload driver, and let it run 15 minutes.
3. Perform a controlled STOP of one of DB2 Analytics Accelerators (as though preparing to upgrade).

   ```
   -BZA1 STOP ACCEL(BZAGTFIN) MODE(QUIESCE) SCOPE(GROUP)
   ```

4. After noticing that a DB2 Analytics Accelerator has been stopped, wait a few minutes and then restart it.

   ```
   -BZA1 STA ACCEL(BZAGTFIN) SCOPE(GROUP)
   ```

5. Simulate an accelerator failure by forcing DB2 Analytics Accelerator to stop immediately.

   ```
   -BZA1 STOP ACCEL(BZAGTFIN) MODE(FORCE) SCOPE(GROUP)
   ```

We next take you through our observations when going through these steps.

Stopping one Accelerator for a rolling upgrade

At time 10:09 we stopped the BZAGTFIN accelerator using the following command:

```
-BZA1 STOP ACCEL(BZAGTFIN) MODE(QUIESCE) SCOPE(GROUP)
```

Figure 8-10 on page 176 shows the ensuing messages displayed in the z/OS and DB2 logs. Notice that DB2 Analytics Accelerator is successfully stopped about one minute later (10:105:05), enabling all of the currently active BZAGTFIN queries to complete successfully.

Figure 8-10 shows the DB2 and z/OS log messages.

Referring to Figure 8-12 on page 178, we notice an associated increase in active queries and accelerator connections on the BZAGSTPR, and no query failures, because now all accelerated queries are being routed to BZAGSTPR. Additionally, notice that the number of currently running queries on DB2 Analytics Accelerator reduces as opposed to stopping completely, which we will soon witness with the next experiment.

```
 10:09:04.74 DINO     00000290  -BZA1 STOP ACCEL(BZAGTFIN) SCOPE(GROUP) MODE(QUIESCE)
 10:09:04.74 STC09635 00000080  DSNX810I  -BZA1 DSNX8CMD STOP ACCEL FOLLOWS -
 10:09:04.74 STC09635 00000080  DSNX863I  -BZA1 DSNX8STO STOP ACCELERATOR INITIATED FOR BZAGTFIN
 10:09:04.74 STC09635 00000090  DSN9035I  -BZA1 BEGIN OF DISPLAY FOR MEMBER: BZA2 650
                  650 00000090  ---------------------------------------------
                  650 00000090  DSNX863I  -BZA2 DSNX8STO STOP ACCELERATOR INITIATED FOR BZAGTFIN
                  650 00000090  --------END OF DISPLAY FOR MEMBER: BZA2    --------
 10:09:04.74 STC09635 00000090  DSN9022I  -BZA1 DSNX8CMD '-STOP ACCEL' NORMAL COMPLETION
.....
10:10:05.74 STC09635 00000080  DSNX860I  -BZA1 DSNX8EKG STOP ACCELERATOR SUCCESSFUL FOR BZAGTFIN
```

Figure 8-10 DB2 log messages for a controlled (QUIESCE) stop of Accelerator

In this experiment, we did not actually conduct a rolling accelerator migration, but for details about how to go about that, see 4.1.5, “Rolling migration in an HA and disaster recovery environment” on page 57.

Accelerator failure simulation (using STOP with FORCE)

At time 23:18:15, we simulated one of our accelerators failing by forcing it to stop immediately using the following command:

-BZA1 STOP ACCEL(BZAGTFIN) MODE(FORCE) SCOPE(GROUP)

Instantly, several BZAGTFIN active queries failed. In the z/OS and DB2 logs for each of the currently active DB2 Analytics Accelerator threads, we received a pair of DSNL027I and DSNL028I messages along with a DSNL511I message. Figure 8-11 on page 177 provides some of these messages, in addition to any DB2 Analytics Accelerator-related messages that we received.

Referencing Figure 8-12 on page 178, we can see an associated increase in active queries and DB2 Analytics Accelerator connections on the BZAGSTPR. Also notice the hard drop off of the number of active connections, unlike the smooth drop off in the previous scenario.

```
10:30:45.55 DINO     00000290  -BZA1 STOP ACCEL(BZAGTFIN) SCOPE(GROUP) MODE(FORCE)
10:30:45.55 STC09635 00000090  DSNL027I  -BZA1 SERVER DISTRIBUTED AGENT WITH 851
                 851 00000090            LUWID=07195096.GC45.140517031818=57700
                 851 00000090
                 851 00000090  THREAD-INFO=BFUSR5:DWH1:BFUSR11:Q11A:*:*:*:<135.25.80.150.3141.140517
                 851 00000090  031818>
                 851 00000090            RECEIVED ABEND=04E
                 851 00000090            FOR REASON=00E50013
....
....
10:30:45.55 STC09635 00000090  DSNL028I  -BZA1 07195096.GC45.140517031818=57700 856
                 856 00000090            ACCESSING DATA FOR
....
....
10:30:45.55 STC09635 00000090  DSNL511I  -BZA1 DSNLIENO TCP/IP CONVERSATION FAILED 865
                 865 00000090            TO LOCATION BZAGTFIN
                 865 00000090            IPADDR=::135.25.80.210 PORT=1400
                 865 00000090            SOCKET=CLOSE RETURN CODE=113 REASON CODE=0523011C
....
....
10:30:45.56 STC09635 00000080  DSNX810I  -BZA1 DSNX8CMD STOP ACCEL FOLLOWS -
10:30:45.56 STC09635 00000080  DSNX863I  -BZA1 DSNX8STO STOP ACCELERATOR INITIATED FOR BZAGTFIN
10:30:45.56 STC09635 00000090  DSN9035I  -BZA1 BEGIN OF DISPLAY FOR MEMBER: BZA2 868
                 868 00000090  --------------------------------------------
                 868 00000090  DSNX863I  -BZA2 DSNX8STO STOP ACCELERATOR INITIATED FOR BZAGTFIN
                 868 00000090  --------END OF DISPLAY FOR MEMBER: BZA2    --------
10:30:45.56 STC09635 00000090  DSN9022I  -BZA1 DSNX8CMD '-STOP ACCEL' NORMAL COMPLETION
10:30:45.80 STC09635 00000080  DSNX860I  -BZA1 DSNX8EKG STOP ACCELERATOR SUCCESSFUL FOR BZAGTFIN
```

Figure 8-11 DB2 log messages for forced (FORCE) stop of DB2 Analytics Accelerator

Figure 8-12 shows metrics for the two outage scenarios.

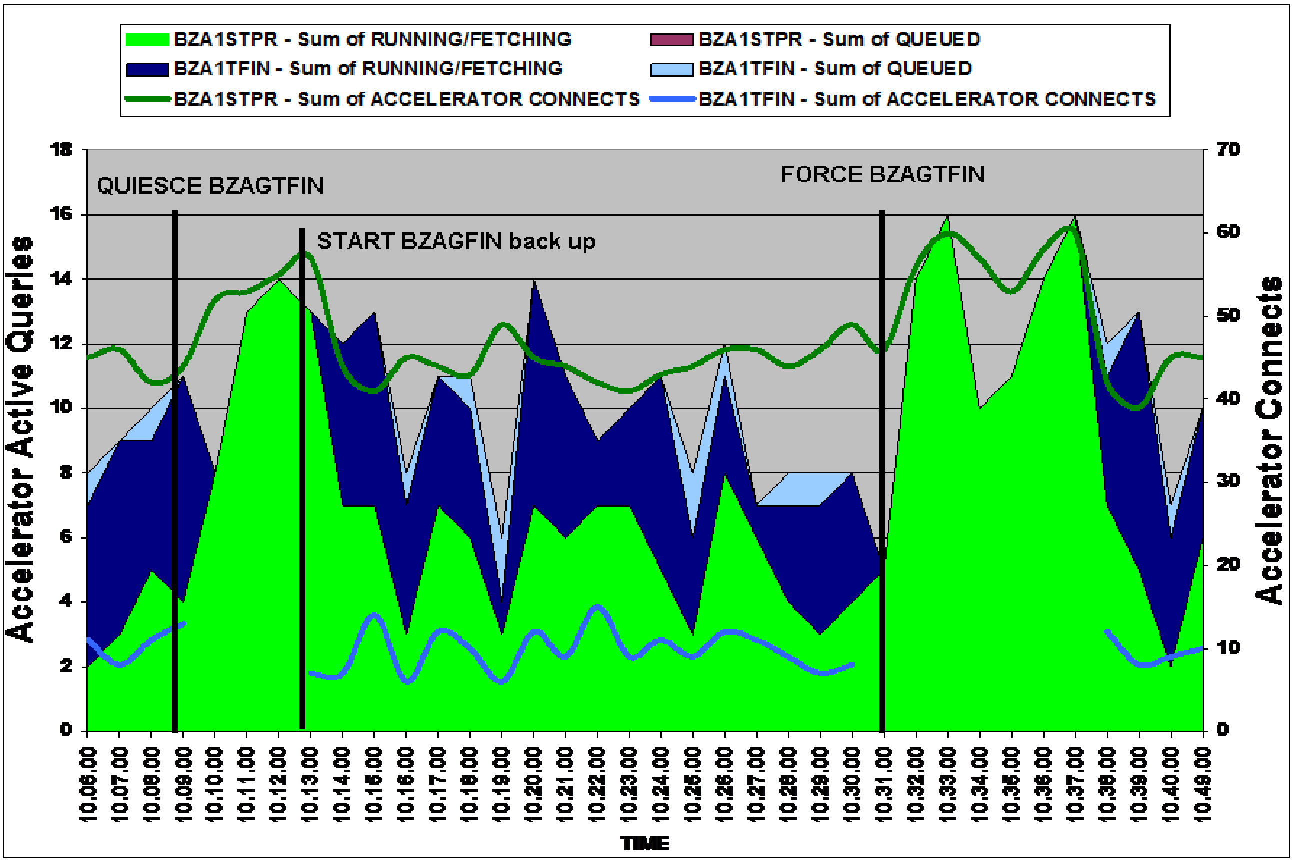


Figure 8-12 Monitoring key DB2 Analytics Accelerator metrics for a planned and an unplanned outage

Important: Having an additional qualified (for WLB) accelerator essentially makes query failover automatic, yielding a *recovery time* and *recovery point* of 0. Although this is valuable and perhaps sufficient for many client's recovery needs, this does not constitute *continuous availability*. As shown, the crashed accelerator's existing queries failed, and would need to be rerun.

In this experiment, we manually forced one of DB2 Analytics Accelerators to stop. The following section, 8.6, "Built-in HA DB2 Analytics Accelerator host server" describes a scenario in which an accelerator stop is *not* initiated by DB2, due to a different type of failure.

8.6 Built-in HA DB2 Analytics Accelerator host server

As discussed previously, there are several built-in HA features in the IBM PureData System for Analytics server. In this scenario, we conducted an experiment evaluating the query workload effect and recovery time associated with a failure with the active NPS host.

Each accelerator is equipped with two SMP servers, which is where the DB2 Analytics Accelerator coordinator node runs. At any point in time, the DB2 Analytics Accelerator coordinator node is running on one of the two SMP servers. The other SMP server acts as a hot standby.

8.6.1 DB2 Analytics Accelerator SMP host failover experiment

In this experiment, we examine how continuous query processing is affected when there is an SMP host failover to the hot standby SMP host. We initiate this activity by forcing the failover using a Netezza-documented host server migration command.

We used the same query workload WLB_loadB, described in “The residency workload balancing query workload” on page 171. The steps were simple:

1. With both accelerators started, start the query workload.
2. Initiate an SMP host server failover.

   A Netezza service representative issued the following command on one of the Netezza host servers:

   ```
   /nzlocal/scripts/heartbeat_admin.sh --migrate
   ```

   Note that we ran this experiment three times. In all three cases, when the SMP host server failover started, within 15 seconds, DB2 issued a stop accelerator command for that accelerator.
3. Manually re-start the failed accelerator.

   We had to issue the START twice, because it failed the first time, likely due to the host failover still being in process.

Figure 8-13 shows the messages from one of the syslogs in the sysplex. The other syslog contained similar messages.

```
10:55:45.71 STC09635 00000090  DSNL511I  -BZA1 DSNLIENO TCP/IP CONVERSATION FAILED 756
                 756 00000090            TO LOCATION BZA1STPR
                 756 00000090            IPADDR=::135.25.80.250 PORT=1400
                 756 00000090            SOCKET=READ RETURN CODE=0 REASON CODE=00000000
10:55:46.71 STC09635 00000090  DSNL511I  -BZA1 DSNLIENO TCP/IP CONVERSATION FAILED 757
                 757 00000090            TO LOCATION BZA1STPR
                 757 00000090            IPADDR=::135.25.80.250 PORT=1400
                 757 00000090            SOCKET=CONNECT RETURN CODE=1128 REASON CODE=76630291
10:55:46.71 STC09635 00000080  DSNX880I  -BZA1 DSNX8EKG DDF CONNECT FAILED WITH 758
                 758 00000080            RETURN CODE=12
                 758 00000080            SQLCODE  = -30081
                 758 00000080            SQLERRMT =
                 758 00000080  TCP/IP SOCKETS ::135.25.80.250 CONNECT 1128 76630291 0000
                 758 00000080            SQLERRP  = DSNLIENO
                 758 00000080            SQLERRD  = 00000009 00000000 00000000 FFFFFFFF 00000000
                 758 00000080  00000000
                 758 00000080            SQLWARN    0= ,1= ,2= ,3= ,4= ,5= ,6= ,7= ,8= ,9= ,A=
                 758 00000080            SQLSTATE = 08001
10:55:46.71 STC09635 00000080  DSNX863I  -BZA1 DSNX8EKG STOP ACCELERATOR INITIATED FOR BZAGSTPR
10:55:46.71 STC09635 00000080  DSNX860I  -BZA1 DSNX8EKG STOP ACCELERATOR SUCCESSFUL FOR BZAGSTPR
( When we noticed DB2 STOPPED DB2 Analytics Accelerator, we tried to manually re-start , but DB2
stopped it again.)
10:56:58.55 DINO     00000290  -BZA1 STA ACCEL(BZAGSTPR) SCOPE(GROUP)
10:56:58.56 STC09635 00000080  DSNX810I  -BZA1 DSNX8CMD START ACCEL FOLLOWS -
10:56:58.56 STC09635 00000080  DSNX820I  -BZA1 DSNX8STA START ACCELERATOR SUCCESSFUL FOR BZAGSTPR
10:56:58.56 STC09635 00000090  DSN9035I  -BZA1 BEGIN OF DISPLAY FOR MEMBER: BZA2 781
                 781 00000090  --------------------------------------------
                 781 00000090  DSNX820I  -BZA2 DSNX8STA START ACCELERATOR SUCCESSFUL FOR BZAGSTPR
                 781 00000090  --------END OF DISPLAY FOR MEMBER: BZA2    --------
10:56:58.56 STC09635 00000090  DSN9022I  -BZA1 DSNX8CMD '-START ACCEL' NORMAL COMPLETION
10:56:58.56 STC09635 00000080  DSNX880I  -BZA1 DSNX8EKG DDF CONNECT FAILED WITH 783
                 783 00000080            RETURN CODE=12
                 783 00000080            SQLCODE  = -30081
                 783 00000080            SQLERRMT =
                 783 00000080  TCP/IP SOCKETS ::135.25.80.250 CONNECT 1128 76630291 0000
                 783 00000080            SQLERRP  = DSNLIENO
                 783 00000080            SQLERRD  = 00000009 00000000 00000000 FFFFFFFF 00000000
                 783 00000080  00000000
                 783 00000080            SQLWARN    0= ,1= ,2= ,3= ,4= ,5= ,6= ,7= ,8= ,9= ,A=
                 783 00000080            SQLSTATE = 08001
10:56:58.56 STC09635 00000080  DSNX863I  -BZA1 DSNX8EKG STOP ACCELERATOR INITIATED FOR BZAGSTPR
10:56:58.56 STC09635 00000080  DSNX860I  -BZA1 DSNX8EKG STOP ACCELERATOR SUCCESSFUL FOR BZAGSTPR
( We waited approximately 15 seconds and then tried to start it again, this time successfully.)
10:57:17.76 DINO     00000290  -BZA1 STA ACCEL(BZAGSTPR) SCOPE(GROUP)
10:57:17.77 STC09635 00000080  DSNX810I  -BZA1 DSNX8CMD START ACCEL FOLLOWS -
10:57:17.77 STC09635 00000080  DSNX820I  -BZA1 DSNX8STA START ACCELERATOR SUCCESSFUL FOR BZAGSTPR
10:57:17.77 STC09635 00000090  DSN9035I  -BZA1 BEGIN OF DISPLAY FOR MEMBER: BZA2 792
                 792 00000090  --------------------------------------------
                 792 00000090  DSNX820I  -BZA2 DSNX8STA START ACCELERATOR SUCCESSFUL FOR BZAGSTPR
                 792 00000090  --------END OF DISPLAY FOR MEMBER: BZA2    --------
10:57:17.77 STC09635 00000090  DSN9022I  -BZA1 DSNX8CMD '-START ACCEL' NORMAL COMPLETION
```

Figure 8-13 Syslog messages after accelerator active NPS Host was lost, therefore accelerator heartbeat lost

At 10:59:13, we displayed the restarted accelerator. The BZAGSTPR accelerator was listed as STARTED and HEALTHY, but had a STATUS of INITIALIZING, as shown in Figure 8-14.

```
10:59:13.56 DINO     00000290  -BZA1 DIS ACCEL(BZAGSTPR),DETAIL
10:59:13.56 STC09349 00000090  CRE106I (ACC     ) DONE
10:59:13.56 STC09635 00000080  DSNX810I  -BZA1 DSNX8CMD DISPLAY ACCEL FOLLOWS -
10:59:13.56 STC09635 00000080  DSNX830I  -BZA1 DSNX8CDA 823
                 823 00000080  ACCELERATOR                      MEMB  STATUS  REQUESTS ACTV QUED MAXQ
                 823 00000080  -------------------------------- ---- -------- -------- ---- ---- ----
                 823 00000080  BZAGSTPR                         BZA1 STARTED         0    0    0    0
                 823 00000080  LOCATION=BZAGSTPR HEALTHY
                 823 00000080  DETAIL STATISTICS
                 823 00000080    LEVEL  = AQT04010
                 823 00000080    STATUS = INITIALIZING
.....
```

Figure 8-14 Accelerator INITIALIZING after active NPS Host failover and manual restart

At 11:02, approximately seven minutes after the host failure, we noticed DSNX881I messages indicating that the NPS system went from discovering to initializing. Then, at 11:04 we see more DSNX881I messages as the NPS system moves through additional states. Finally, at 11:07 we notice that DB2 Analytics Accelerator is back ONLINE.

Figure 8-15 shows the actual messages.

```
11:02:37.99 STC09635 00000080  DSNX881I  -BZA1 1 I 1579 (2014-05-15 15:18:13 UTC) 880
                 880 00000080  BZAGSTPR(135.25.80.250) NPS system zbap-n2001-h2 went from
                 880 00000080  discovering to initializing at 15-May-14, 15:18:13 UTC User
                 880 00000080  initiated. Event: eventType: sysStateChanged eventTimestamp:
                 880 00000080  15-May-14, 15:18:13 UTC eventArgs: prev
11:02:37.99 STC09635 00000080  DSNX881I  -BZA1 1 I 1579 (2014-05-15 15:18:13 UTC) 881
                 881 00000080  BZAGSTPR(135.25.80.250) iousState=discovering,
                 881 00000080  currentState=initializing, eventSource=user eventSource: User
                 881 00000080  initiated event
....
11:04:18.16 STC09635 00000080  DSNX881I  -BZA1 1 I 1582 (2014-05-15 15:20:08 UTC) 914
                 914 00000080  BZAGSTPR(135.25.80.250) NPS system zbap-n2001-h2 went from
                 914 00000080  initializing to initialized at 15-May-14, 15:20:08 UTC User
                 914 00000080  initiated. Event: eventType: sysStateChanged eventTimestamp:
                 914 00000080  15-May-14, 15:20:08 UTC eventArgs: prev
11:04:18.16 STC09635 00000080  DSNX881I  -BZA1 1 I 1582 (2014-05-15 15:20:08 UTC) 915
                 915 00000080  BZAGSTPR(135.25.80.250) iousState=initializing,
                 915 00000080  currentState=initialized, eventSource=user eventSource: User
                 915 00000080  initiated event
11:04:26.95 STC09350 00000281  ICH70001I INFOCDC  LAST ACCESS AT 11:03:49 ON THURSDAY, MAY 15, 2014
....
11:04:38.32 STC09635 00000080  DSNX881I  -BZA1 1 I 1583 (2014-05-15 15:20:11 UTC) 917
                 917 00000080  BZAGSTPR(135.25.80.250) NPS system zbap-n2001-h2 went from
                 917 00000080  initialized to preOnlining at 15-May-14, 15:20:11 UTC User initiated.
                 917 00000080  Event: eventType: sysStateChanged eventTimestamp: 15-May-14, 15:20:11
                 917 00000080  UTC eventArgs: previ
11:04:38.32 STC09635 00000080  DSNX881I  -BZA1 1 I 1583 (2014-05-15 15:20:11 UTC) 918
                 918 00000080  BZAGSTPR(135.25.80.250) ousState=initialized,
                 918 00000080  currentState=preOnlining, eventSource=user eventSource: User
                 918 00000080  initiated event
11:04:38.32 STC09635 00000080  DSNX881I  -BZA1 1 I 1584 (2014-05-15 15:20:15 UTC) 919
                 919 00000080  BZAGSTPR(135.25.80.250) NPS system zbap-n2001-h2 went from
                 919 00000080  preOnlining to resuming at 15-May-14, 15:20:15 UTC User initiated.
                 919 00000080  Event: eventType: sysStateChanged eventTimestamp: 15-May-14, 15:20:15
                 919 00000080  UTC eventArgs: previous
11:04:38.32 STC09635 00000080  DSNX881I  -BZA1 1 I 1584 (2014-05-15 15:20:15 UTC) 920
                 920 00000080  BZA1STPR(135.25.80.250) State=preOnlining, currentState=resuming,
                 920 00000080  eventSource=user eventSource: User initiated event
11:04:38.32 STC09635 00000080  DSNX881I  -BZA1 1 I 1585 (2014-05-15 15:20:17 UTC) 921
                 921 00000080  BZAGSTPR(135.25.80.250) NPS system zbap-n2001-h2 went from resuming
                 921 00000080  to online at 15-May-14, 15:20:17 UTC User initiated. Event:
                 921 00000080  eventType: sysStateChanged eventTimestamp: 15-May-14, 15:20:17 UTC
                 921 00000080  eventArgs: previousState
11:04:38.32 STC09635 00000080  DSNX881I  -BZA1 1 I 1585 (2014-05-15 15:20:17 UTC) 922
                 922 00000080  BZAGSTPR(135.25.80.250) =resuming, currentState=online,
                 922 00000080  eventSource=user eventSource: User initiated event
....
11:07:18.34 STC09635 00000080  DSNX871I  -BZA1 DSNX8EKG ACCELERATOR BZAGSTPR IS ONLINE
```

Figure 8-15 DSNX881I messages as accelerator and underlying NPS system recover

The following list summarizes the key time stamps:

10:55:46 Accelerator STOPPED.

10:57:17 Accelerator manually re-started successfully.

11:07:18 Accelerator is started and back ONLINE.

To verify that queries were again being routed to the recovered accelerator, we examined our one minute statistics trace data as we did in previous experiments. We witnessed at minute ending 11:08:00, we again have nonzero counts for accelerator connects, and successful query completions for the BZAGSTPR accelerator.

Summary

In summary, assuming that our manual restart of DB2 Analytics Accelerator was conducted within a few minutes after DB2 STOPPED DB2 Analytics Accelerator, our host failover *recovery time* of the accelerated query workload was approximately 11.5 minutes. The *recovery point time* is zero, because no data needed to be re-created.

8.7 Remote DB2 Analytics Accelerator experiments

In a DR scenario, the remote DB2 Analytics Accelerator is likely to be located at some distance, perhaps minimally 100 kilometers (KM) or 60 miles away. In this residency, we did not have the luxury of a robust DR infrastructure, but we thought it was important to obtain some sense of remote latency, and to verify that there was no unexpected behavior.

We conducted an experiment between the IBM Poughkeepsie Lab and the IBM Silicon Valley Lab (SVL), using the IBM internal public network. This network is shared by all IBM employees, and its performance is likely to vary minute to minute. Therefore, we don't have a high expectation for quality latency numbers. The following list describes the lab setups:

Local accelerator DB2 and DB2 Analytics Accelerator are both in Poughkeepsie, NY. The network for DB2 Analytics Accelerator is a dedicated private 10 gigabit (Gb) network.

Remote Accelerator DB2 is in Poughkeepsie, NY, and DB2 Analytics Accelerator is in San Jose, CA. The network is a shared IBM public network.

8.7.1 Experiments

Our objectives with this experiment were to demonstrate remote DB2 Analytics Accelerator latency, and to explain the functionality of the same.

To prepare for this experiment, we had to configure the SVL DB2 Analytics Accelerator in an unorthodox manner, temporarily setting DB2 Analytics Accelerator wall IP to a public IP on the IBM network. This allowed us to authenticate, pair, and add the SVL accelerator to our DB2 11 BZAG data sharing group.

We ran the following experiments:

- Accelerator load
- Query
- ACCEL_TEST_CONNECTION stored procedure

We re-emphasize that the remote accelerator tests were *not* run in a dedicated network DR environment, but provides some level of confidence in functionality, and examples of attainable latency in a non-dedicated network.

Accelerator load

We ran a load of a single partitioned table with 100 partitions that were unevenly distributed. AQT_MAX_UNLOAD_IN_PARALLEL was set to 16. Especially because we were running on a "public" network, we ran the remote load multiple times.

Table 8-5 contains the results of the accelerator load.

Table 8-5 Local versus remote accelerator load

Accelerator load of a five-million-row FACT table (500 megabytes, or MB)	Elapsed time (seconds)	Load rate (GB per hour)
Local N1001-010 (Poughkeepsie)	11	152.0
Remote N1001-010 (SVL) Run 1	948	1.8
Remote N1001-010 (SVL) Run 2	47	37.0

Functionally, the remote accelerator loads ran well. Due to the performance variability of the IBM public network, we witnessed some wide variations in elapsed time.

Query

For this test, we ran two different queries. Each one was run serially five times. Table 8-6 contains the results.

Table 8-6 Accelerator Query: Local versus Remote

Query	Answer set size		N1001-010 Average elapsed time (seconds)	
	Rows	Bytes	Local	Remote (SVL)
Q11A	50	27,000	2.0	3.0
Q1B	2,358	1,250,000	1.1	1.6

Despite the remote accelerator being on a non-dedicated network, average query elapsed time did not suffer much. We repeated these tests several times, and the remote accelerator elapsed times were quite erratic, likely due to the network. Therefore, we suggest that the general result with regard to these remote query workloads is that they functionally ran successfully, and in some cases still maintained elapsed times similar to local queries.

ACCEL_TEST_CONNECTION stored procedure

The DB2 Analytics Accelerator ACCEL_TEST_CONNECTION stored procedure is provided as a standard stored procedure with the DB2 Analytics Accelerator product. It enables you to determine the following functionality:

- Whether the mainframe computer can contact (ping) DB2 Analytics Accelerator over the network
- Whether the network path from DB2 to DB2 Analytics Accelerator has been properly configured
- Whether the IBM Distributed Relational Database Architecture™ (IBM DRDA®) connection between the DB2 subsystem and DB2 Analytics Accelerator works after completing the pairing process
- What the data transfer rate (load performance) is

We were mainly interested in using the stored procedure to obtain another indicator of the data transfer rate (or load performance potential) for the remote DB2 Analytics Accelerator over this non-dedicated network. Abstracting the *overall performance counter* data from the stored procedure output, we achieved the results depicted in Table 8-7 on page 185.

Table 8-7   ACCEL_TEST_CONNECTION transfer rate end-to-end

Overall performance counter	Local	Remote
Number of sender threads	10	10
Total number of bytes received	10,000,000,000	10,000,000,000
Total elapse time for receiving data in milliseconds (ms)	25,662	258,971
Transfer rate end-to-end	371.63	36.83
Transfer rate converted to GB per hour	**1,308**	**130**

Methodology for running the stored procedure

For those desiring to run this stored procedure, we provide information about how we ran it using DB2 Analytics Accelerator studio:

1. First, go to the **Application Objects** → **Stored Procedures** pane.
2. Select the DB2 Analytics Accelerator **ACCEL_TEST_CONNECTION** stored procedure.
3. Right-click it and select **RUN** (see Figure 8-16).

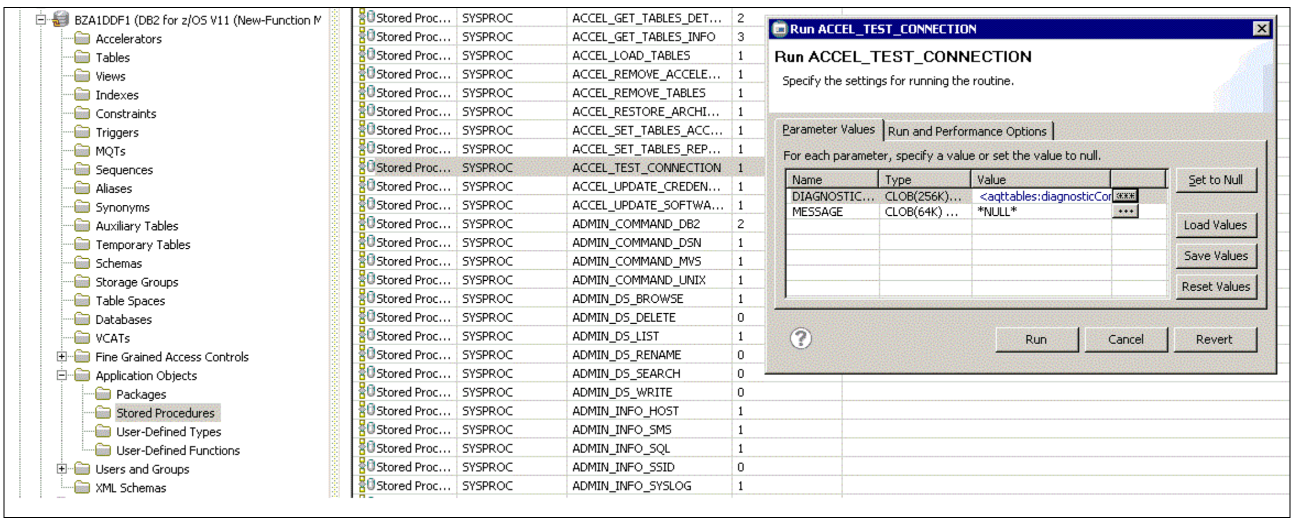


Figure 8-16   Running ACCELERATOR_TEST_CONNECTION stored procedure

4. Next, provide the necessary input parameters by clicking in the input box in the fourth column:
   a. For MESSAGE, we provided ‘NULL’.
   b. For DIAGNOSTICS, we provided the Extensible Markup Language (XML) input string shown in Figure 8-17.

```
<?xml version="1.0" encoding="UTF-8" ?><aqttables:diagnosticCommand
xmlns:aqttables="http://www.ibm.com/xmlns/prod/dwa/2011"
version="1.0"><networkSpeed accelerator="<SVLTFIN>"
totalNumberBytes="1000000000" dataBatchSize="320000"
parallelConnections="10" seed="12345"/></aqttables:diagnosticCommand>
```

Figure 8-17   ACCEL_STORED_PROCEDURE diagnostics specification

5. Next, click **RUN**, and wait for it to complete. When complete, the results can be found in the SQL RESULTS window, under the Parameters tab.
6. Select the DIAGNOSTICS row, and then click the Ellipses (...) box in the Value(OUT) column, as shown in Figure 8-18.

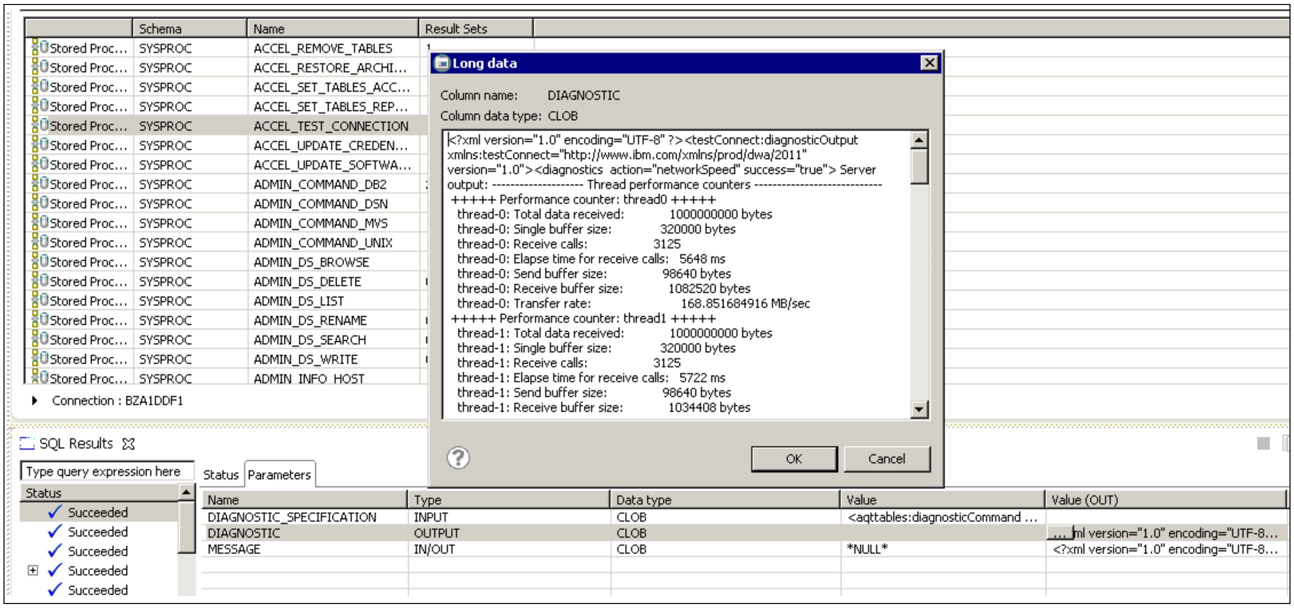


Figure 8-18 ACCEL_TEST_CONNECTION output

9

Monitoring enhancements

This chapter provides information about monitoring and capacity planning for IBM DB2 Analytics Accelerator for z/OS (DB2 Analytics Accelerator). It describes tools and methods for monitoring an accelerator. In addition, it explains indicators that might anticipate if an accelerator is reaching its capacity limit, or is in need of an upgrade.

Furthermore, this chapter provides information about methods and tools for assessing DB2 workloads regarding eligibility for acceleration, and about how to estimate benefits. It also presents some details about how the effect and benefits can actually be measured when a workload is moved from a mainframe to DB2 Analytics Accelerator.

This chapter contains information about the following topics:

- Performance monitoring and capacity planning
- Approaches to capacity planning
- Assessing existing DB2 workloads
- Measuring actual benefits of moving workloads to DB2 Analytics Accelerator

9.1 Performance monitoring and capacity planning

If you have an accelerator running, it can be important to monitor the system, particularly if DB2 Analytics Accelerator is running in a production environment. You monitor the system to evaluate the following conditions:

- If the system is getting close to its limits and needs to be upgraded
- If adding additional workload is still possible

You want to know when DB2 Analytics Accelerator has reached its capacity limit. Because DB2 Analytics Accelerator is an appliance, you do not have the option to add memory, central processing units (CPU), or more disk drives. The way to upgrade is to replace DB2 Analytics Accelerator with a bigger model, or to add more accelerators and use workload balancing (WLB).

A concept called *Growth on Demand* (GoD) is also available, which means that you buy a bigger DB2 Analytics Accelerator model than you currently need, and you only use and pay for part of its capacity. You can add capacity when you need it in an easy way. This concept is also valuable if you have a challenge to optimally size a DB2 Analytics Accelerator solution, because you anticipate growth for your data warehouse environment.

Again, because DB2 Analytics Accelerator is an appliance, it is also not possible to access DB2 Analytics Accelerator system directly. All monitoring data is available through DB2 only. A *heartbeat* process (signal) running on DB2 Analytics Accelerator provides monitoring data to DB2, and DB2 externalizes that data using instrumentation facility component identifier (IFCID) records.

The following list includes methods to monitor DB2 Analytics Accelerator, and are explained by way of examples in the following sections:

- DB2 Analytics Accelerator Studio
- DB2 commands
- DB2 monitoring tools:
  - IBM Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS (OMEGAMON PE)
  - IBM Tivoli OMEGAMON XE for DB2 Performance Monitor on z/OS (OMEGAMON PM)
- Query history table

9.1.1 Monitoring with DB2 Analytics Accelerator Studio

The IBM Data Studio plug-in DB2 Analytics Accelerator Studio, which is delivered with the product, provides monitoring information.

Figure 9-1 shows a portion of the DB2 Analytics Accelerator view in IBM Data Studio. In particular, it shows the information about monitoring DB2 Analytics Accelerator from a system view and from a query view.

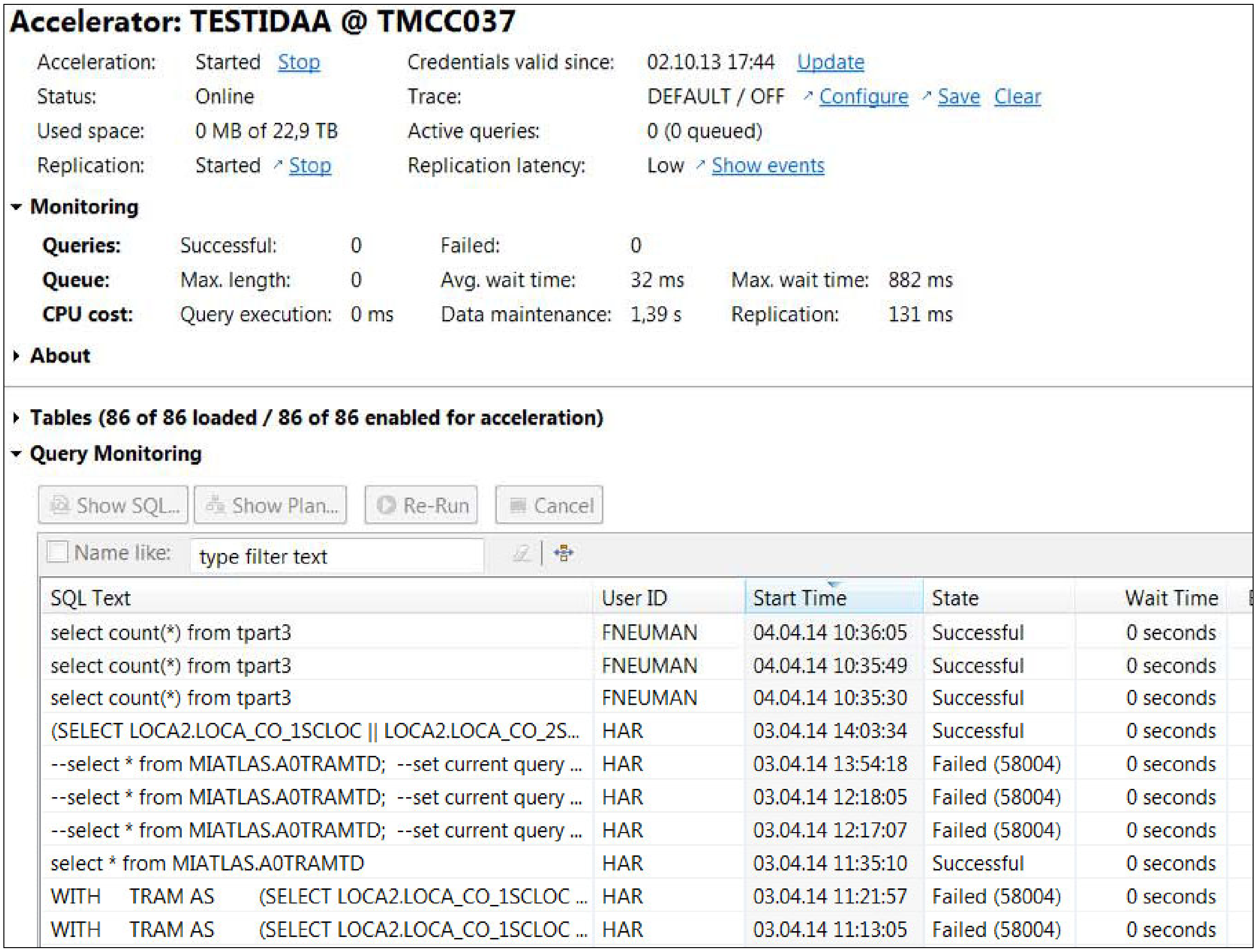


Figure 9-1 Monitoring queries with Accelerator Studio

The following list provides a description of some important counters:

- Used space (available space) in terabytes (TB): The amount of disk space that is taken up by the tables, in proportion to the available space on DB2 Analytics Accelerator.
- Number of active / queued queries (#): The number of accelerated queries that are currently being processed. In parentheses, you see the number of queued queries.
- Maximum queue length (#): The highest number of queries in the queue, up to this time, since the start of the currently viewed accelerator.
- Average wait time in milliseconds (ms): Average time that queries had to stay in the queue. The basis for the calculation is all queries that were processed during the last minute.
- Maximum wait time (ms): Longest time that a query stayed in the queue, that is, before it was processed since the start of the currently viewed accelerator.

A detailed explanation of these numbers can be found in *IBM DB2 Analytics Accelerator for z/OS Version 4.1.0 User's Guide*, SH12-7040.

9.1.2 Monitoring with the DB2 command -DIS ACCEL(*name*) DETAIL

Figure 9-2 shows the output of the DB2 command **-DIS ACCEL(*name*) DETAIL**, where *name* is the name of DB2 Analytics Accelerator. This command displays monitoring information, such as active or queued queries, wait times, and disk storage usage. They are numbers referring to the DB2 Analytics Accelerator that is connected to the subsystem from which the command was issued, and also numbers referring to *all* subsystems connected to that accelerator.

```
DSNX810I  -BZA1 DSNX8CMD DISPLAY ACCEL FOLLOWS -
DSNX830I  -BZA1 DSNX8CDA 561
ACCELERATOR                      MEMB  STATUS   REQUESTS ACTV QUED MAXQ
-------------------------------- ---- -------- -------- ---- ---- ----
BZA1STPR                         BZA1 STARTED      1928    0    0   17
LOCATION=BZA1STPR HEALTHY
DETAIL STATISTICS
  LEVEL  = AQT04012
  STATUS = ONLINE
  FAILED REQUESTS                                    =        0
  AVERAGE QUEUE WAIT                                 =       38 MS
  MAXIMUM QUEUE WAIT                                 =   690985 MS
  TOTAL NUMBER OF ACTIVE PROCESSORS                  =      224
  AVERAGE CPU UTILIZATION ON COORDINATOR NODES       =     5.00%
  AVERAGE CPU UTILIZATION ON WORKER NODES            =      .99%
  AVERAGE DISK IO UTILIZATION                        =      .00%
  NUMBER OF ACTIVE WORKER NODES                      =        7
  TOTAL DISK STORAGE                                 = 48000960 MB
  DISK STORAGE IN USE FOR THIS DB2 SYSTEM            =  4569951 MB
  DISK STORAGE IN USE FOR ALL DB2 SYSTEMS            =  9635154 MB
  TOTAL CPU FOR REQUESTS FOR THIS DB2 SYSTEM         =  9495580 MS
  TOTAL CPU FOR DATA MAINTENANCE FOR THIS DB2 SYSTEM = 126732860 MS
  TOTAL CPU FOR REPLICATION FOR THIS DB2 SYSTEM      =   172377 MS
DISPLAY ACCEL REPORT COMPLETE
```

Figure 9-2 -DIS ACCEL DETAIL output

The definition of the counters in Figure 9-2 are explained in detail in *DB2 11 for z/OS Messages*, GC19-4062. The following list gives descriptions of the most important counters:

ACTV
The current number of *active* accelerated requests for this DB2 system.

QUED
The current number of *queued* accelerated requests for this DB2 system.

MAXQ
The highest number of queued requests that have been reached since DB2 Analytics Accelerator was started.

AVERAGE QUEUE WAIT
The length of the average queue wait time in DB2 Analytics Accelerator, measured in ms.

MAXIMUM QUEUE WAIT
The length of the longest queue wait time in DB2 Analytics Accelerator, measured in ms.

TOTAL DISK STORAGE

The total amount of disk storage that is available in DB2 Analytics Accelerator, measured in megabytes (MB).

DISK STORAGE IN USE FOR ALL DB2 SYSTEMS

The amount of disk storage that is being used by all of the connected DB2 subsystems in DB2 Analytics Accelerator, measured in MB.

The following counters are new in DB2 Analytics Accelerator V4:

TOTAL CPU FOR DATA MAINTENANCE FOR THIS DB2 SYSTEM

The total processor time for data maintenance operations in DB2 Analytics Accelerator for this DB2 system.

TOTAL CPU FOR REPLICATION FOR THIS DB2 SYSTEM

The total processor time that is associated with the replication apply process in DB2 Analytics Accelerator for this DB2 system.

9.1.3 Monitoring with IBM Tivoli OMEGAMON Performance Expert

Figure 9-3 on page 192 shows the Statistics Report. The left side shows statistics for the subsystem only. The right side shows statistics for DB2 Analytics Accelerator in total (for all subsystems connected to this accelerator).

The data at the bottom relates to change data capture information.

The data in the middle on the right relates to capacity planning information.

Figure 9-3 shows the Statistics Report.

```
Q100              FOR SUBSYSTEM ONLY                QUANTITY   Q100              TOTAL ACCELERATOR                 QUANTITY
 ----------------------------------  --------------------      ----------------------------------  --------------------
 QUERIES SUCCESSFULLY EXECUTED                           1.00   QUERIES SUCCESSFULLY EXECUTED                          1.00
 QUERIES FAILED TO EXECUTE                               1.00   QUERIES FAILED TO EXECUTE                              1.00
 CURRENTLY EXECUTING QUERIES                             0.00   CURRENTLY EXECUTING QUERIES                            0.23
 MAXIMUM EXECUTING QUERIES                               1.00   MAXIMUM EXECUTING QUERIES                              1.00
 CPU TIME EXECUTING QUERIES                          1.290000   CPU TIME EXECUTING QUERIES                         1.290000
 CPU TIME LOAD/ARCHIVE/RESTORE                   15:42.600000   CPU TIME LOAD/ARCHIVE/RESTORE                  15:42.600000

 CONNECTS TO ACCELERATOR                                 4.00   ACCELERATOR SERVER START              09/05/13 13:36:48.19
 REQUESTS SENT TO ACCELERATOR                            6.00   ACCELERATOR STATUS CHANGE             09/09/13 11:47:05.22
   TIMED OUT                                             0.00
   FAILED                                                0.00   DISK STORAGE AVAILABLE (MB)                     48000959.97
 BYTES SENT TO ACCELERATOR                            7618.00     IN USE FOR ACCEL DB - ALL DB2 (MB)             1932487.60
 BYTES RECEIVED FROM ACCELERATOR                      2707.00     IN USE FOR ACCEL DB - THIS DB2(MB)               64322.60
 MESSAGES SENT TO ACCELERATOR                           33.00
 MESSAGES RECEIVED FROM ACCEL                           33.00   MAXIMUM QUEUE LENGTH                                   0.00
 BLOCKS SENT TO ACCELERATOR                              0.00   CURRENT QUEUE LENGTH                                   0.00
 BLOCKS RECEIVED FROM ACCELERATOR                        2.00   AVG QUEUE WAIT ELAPSED TIME                        0.021328
 ROWS SENT TO ACCELERATOR                                0.00   MAX QUEUE WAIT ELAPSED TIME                        0.945941
 ROWS RECEIVED FROM ACCELERATOR                         53.00
                                                                WORKER NODES                                           7.00
 TCP/IP SERVICES ELAPSED TIME                    28:18.061328   WORKER NODES DISK UTILIZATION (%)                      2.40
 ELAPSED TIME IN ACCELERATOR                         7.791182   WORKER NODES AVG CPU UTILIZATION (%)                  23.14
 WAIT TIME IN ACCELERATOR                            0.099476   COORDINATOR CPU UTILIZATION (%)                        8.71
                                                                PROCESSORS                                           224.00
                                                                DATA SLICES                                          240.00

 CPU TIME FOR REPLICATION                                 N/P   CPU TIME FOR REPLICATION                                N/P
 LOG RECORDS READ                                         N/P   LOG RECORDS READ                                        N/P
   LOG RECORDS FOR ACCEL TABLES                           N/P     LOG RECORDS FOR ACCEL TABLES                          N/P
 LOG RECORD BYTES PROCESSED                               N/P   LOG RECORD BYTES PROCESSED                              N/P
 INSERT ROWS FOR ACCEL TABLES                             N/P   INSERT ROWS FOR ACCEL TABLES                            N/P
 UPDATE ROWS FOR ACCEL TABLES                             N/P   UPDATE ROWS FOR ACCEL TABLES                            N/P
 DELETE ROWS FOR ACCEL TABLES                             N/P   DELETE ROWS FOR ACCEL TABLES                            N/P
 REPLICATION LATENCY IN SECONDS                           N/P
 REPLICATION STATUS CHANGE                                N/P
```

Figure 9-3 OMEGAMON PE statistics report

The following list provides a description of some important counters:

CURRENTLY running QUERIES

The number of currently running queries in DB2 Analytics Accelerator. This includes the queries from all of the DB2 systems connected to this accelerator.

MAXIMUM running QUERIES

The maximum number of queries running in DB2 Analytics Accelerator at any time since accelerator start. This includes the queries from all of the DB2 systems connected to this accelerator.

WAIT TIME IN ACCELERATOR

The accumulated wait time spent in DB2 Analytics Accelerator when running requests from the DB2 subsystem.

MAX QUEUE WAIT ELAPSED TIME

The maximum wait time of the DB2 Analytics Accelerator query queue.

MAXIMUM QUEUE LENGTH

The longest query queue length at DB2 Analytics Accelerator.

CURRENT QUEUE LENGTH

The current number of queued, accelerated requests.

DISK STORAGE AVAILABLE (MB)

The disk storage (MB) available at DB2 Analytics Accelerator.

This tool also has the capability to capture data and store it in the tool's Performance database, or unload it in `.csv` format to further process the data in spreadsheet applications. This is helpful to monitor growth rates and to analyze data trends.

The IBM Redbooks publication *Hybrid Analytics Solution using IBM DB2 Analytics Accelerator for z/OS V3.1*, SG24-8151, shows how to use the OMEGAMON PE Performance database, and how to use the OMEGAMON PE spreadsheet data generator.

The OMEGAMON PE product page is found on the following web page:

http://www.ibm.com/software/products/de/tivoliomegamonxefordb2performanceexpertonzos

The definitions of all metrics described previously are explained in detail in the *IBM Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS Version 5.2.0 Report Reference*, SH12-6991:

http://publibfp.dhe.ibm.com/epubs/pdf/ko2rr520.pdf

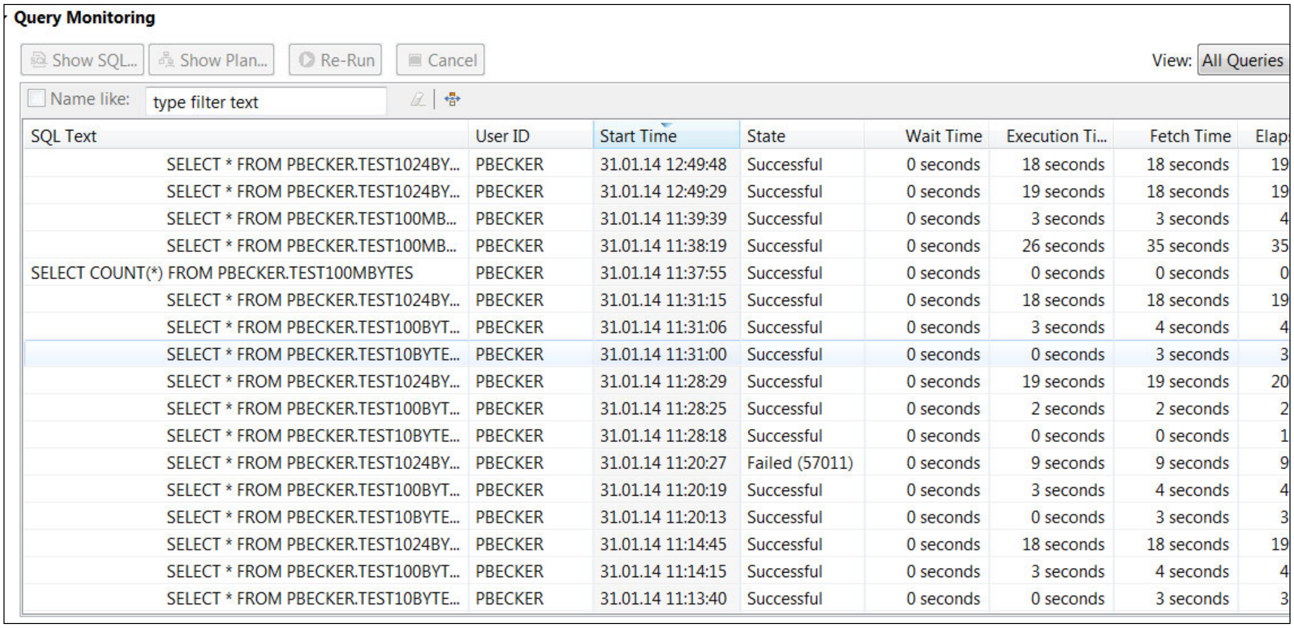
Updates to this document are described on the following web page:

http://www.ibm.com/support/docview.wss?uid=swg27041521

9.1.4 Monitoring using query history

DB2 Analytics Accelerator maintains a history of the most recent accelerated queries. DB2 Analytics Accelerator Studio can display the most recent queries that have been accelerated in the query monitoring section of the DB2 Analytics Accelerator View.

Figure 9-4 shows part of the query monitoring section of DB2 Analytics Accelerator View in DB2 Analytics Accelerator Studio.



SQL Text	User ID	Start Time	State	Wait Time	Execution Ti...	Fetch Time	Elap
SELECT * FROM PBECKER.TEST1024BY...	PBECKER	31.01.14 12:49:48	Successful	0 seconds	18 seconds	18 seconds	19
SELECT * FROM PBECKER.TEST1024BY...	PBECKER	31.01.14 12:49:29	Successful	0 seconds	19 seconds	18 seconds	19
SELECT * FROM PBECKER.TEST100MB...	PBECKER	31.01.14 11:39:39	Successful	0 seconds	3 seconds	3 seconds	4
SELECT * FROM PBECKER.TEST100MB...	PBECKER	31.01.14 11:38:19	Successful	0 seconds	26 seconds	35 seconds	35
SELECT COUNT(*) FROM PBECKER.TEST100MBYTES	PBECKER	31.01.14 11:37:55	Successful	0 seconds	0 seconds	0 seconds	0
SELECT * FROM PBECKER.TEST1024BY...	PBECKER	31.01.14 11:31:15	Successful	0 seconds	18 seconds	18 seconds	19
SELECT * FROM PBECKER.TEST100BYT...	PBECKER	31.01.14 11:31:06	Successful	0 seconds	3 seconds	4 seconds	4
SELECT * FROM PBECKER.TEST10BYTE...	PBECKER	31.01.14 11:31:00	Successful	0 seconds	0 seconds	3 seconds	3
SELECT * FROM PBECKER.TEST1024BY...	PBECKER	31.01.14 11:28:29	Successful	0 seconds	19 seconds	19 seconds	20
SELECT * FROM PBECKER.TEST100BYT...	PBECKER	31.01.14 11:28:25	Successful	0 seconds	2 seconds	2 seconds	2
SELECT * FROM PBECKER.TEST10BYTE...	PBECKER	31.01.14 11:28:18	Successful	0 seconds	0 seconds	0 seconds	1
SELECT * FROM PBECKER.TEST1024BY...	PBECKER	31.01.14 11:20:27	Failed (57011)	0 seconds	9 seconds	9 seconds	9
SELECT * FROM PBECKER.TEST100BYT...	PBECKER	31.01.14 11:20:19	Successful	0 seconds	3 seconds	4 seconds	4
SELECT * FROM PBECKER.TEST10BYTE...	PBECKER	31.01.14 11:20:13	Successful	0 seconds	0 seconds	3 seconds	3
SELECT * FROM PBECKER.TEST1024BY...	PBECKER	31.01.14 11:14:45	Successful	0 seconds	18 seconds	18 seconds	19
SELECT * FROM PBECKER.TEST100BYT...	PBECKER	31.01.14 11:14:15	Successful	0 seconds	3 seconds	4 seconds	4
SELECT * FROM PBECKER.TEST10BYTE...	PBECKER	31.01.14 11:13:40	Successful	0 seconds	0 seconds	3 seconds	3

Figure 9-4 Query history: Queries run on DB2 Analytics Accelerator

The document *How to store the query history in a DB2 for z/OS table* explains how you can copy the query history to a DB2 for z/OS table:

http://www.ibm.com/support/docview.wss?uid=swg27039739

The document contains the Data Definition Language (DDL) to create the table, and the ACCEL.GET_QUERY_HISTORY stored procedure that populates this table.

Here is a list of all of the available fields:

planID DB2 Analytics Accelerator plan ID that can be used to retrieve the full Structured Query Language (SQL) query text, and detailed plan information.

User The client-side user ID that submitted the query.

productID Identification of the product that routed the query, such as a Java Database Connectivity (JDBC) driver.

clientUser The name of the user on whose behalf the application that is using the connection is running.

workstation The value of the workstation name from the client information that is specified for the connection.

application The value of the application name from the client information that is specified for the connection.

locationName Location Name of the DB2 subsystem.

connName Name of the connection.

connType Type of the connection.

corrID Correlation ID.

authID Authorization ID.

planName Name of the DB2 plan.

accounting Optional client-specific accounting information that the client application provides.

subSystemID Name of the subsystem from which SQL was sent.

state The execution state of the query.

submitTimestamp DB2 Analytics Accelerator time stamp at which the query was submitted to DB2 Analytics Accelerator.

waitTimeSec The time required for preprocessing the query in DB2 Analytics Accelerator (before it can be sent to the IBM PureData for Analytics (formerly IBM Netezza) back-end database system), and any queue wait time from Netezza.

fetchTimeSec The time measured from receiving the first row from the Netezza back-end database system until the last row requested by DB2 and send to DB2. The fetch time usually overlaps with the execution time, because the first row is received from DB2 Analytics Accelerator before query execution in DB2 Analytics Accelerator can terminate.

The value is provided only if fetching result rows from the DB2 Analytics Accelerator back-end database system has already started.

CPUTimeSec CPU time required for running the query in the DB2 Analytics Accelerator back-end database system and on DB2 Analytics Accelerator. The value is provided only if CPU time information was already collected in DB2 Analytics Accelerator and from the DB2 Analytics Accelerator back-end database system.

elapsedTimeSec The total processing time of the query on DB2 Analytics Accelerator, including wait time.

priority The priority of the query execution on DB2 Analytics Accelerator. The attribute is only set if an explicit priority was associated with the query received from DB2.

resultRows Number of rows returned by the query. This value is an estimate if the query is not completed.

resultBytes Number of bytes returned by the query. This value is an estimate if the query is not completed.

errorDescription Description of the SQL error in case the query did not complete successfully.

Task DB2 Analytics Accelerator task that is running the query.

Sqltext The first 128 bytes of the SQL text of the query. Longer queries are truncated at character boundaries.

Having this data in a DB2 table opens the possibility to create customized reports to monitor DB2 Analytics Accelerator.

For instance, you can pick a subset of queries and analyze the growth rate of wait times and elapsed times of this set over time:

```
SELECT APPLICATION, DATE(SUBMITTIMESTAMP) DATE, SUM(WAITTIMESEC) WAIT,
SUM(ELAPSEDTIMESEC) ELAPSED
FROM ACCEL.QUERY_HISTORY_TABLE
WHERE APPLICATION = 'DB2JCC_APPLICATION'
GROUP BY APPLICATION, DATE(SUBMITTIMESTAMP);
```

This also provides the ability to report which proportion of DB2 Analytics Accelerator usage is attributed to which applications, and how much workload is coming from various sources, such as remote applications.

9.2 Approaches to capacity planning

Looking at the criteria on which an accelerator is initially sized is a good starting point to find out which counters are useful to monitor. Additionally, it can provide an indication of whether DB2 Analytics Accelerator is at its size limits.

The DB2 Analytics Accelerator system is sized based upon three criteria:

- The amount of data stored
- The complexity of the queries
- The arrival rate of the queries

Figure 9-5 shows an overview of the available models.

N2001 Models	005	010	020	040
Cabinets	1/2	1	2	4
S-Blades	4	7	14	28
Processing Units	64	112	224	448
Capacity (TB)	24	48	96	192
Effective Capacity (TB)*	96	192	384	768

N2002 Model	002	005	010	020	040
Cabinets	¼	1/2	1	2	4
S-Blades	2	4	7	14	28
Processing Units	32	64	112	224	448
Capacity (TB)	8	24	48	96	192
Effective Capacity (TB)*	32	96	192	384	768

N1001 Models	002	005	010	015	020	030	040	060	080	100
Cabinets	¼	½	1	1 ½	2	3	4	6	8	10
S-Blades	4	7	14	18	28	42	56	84	112	140
Processing Units	32	56	112	144	224	336	448	672	896	1120
Capacity (TB)	8	16	32	48	64	96	128	192	256	320
Effective Capacity (TB)*	32	64	128	192	256	384	512	768	1024	1280

*Figure 9-5 Accelerator models - *) Effective Capacity = User data space with compression (4x compression assumed)*

The numbers that are needed for an initial sizing are referring to the volume of data moved to DB2 Analytics Accelerator, and the characteristics of the workload that should be run on DB2 Analytics Accelerator:

- Size of uncompressed data you want to move (in TB)
- Annual growth rate of your application (as a percentage)
- Planned life of the application (in years)
- Peak processing usage (as a percentage)
- Peak storage usage (as a percentage)
- Estimated data compression ratio (a factor)
- Number of active users per hour (a number)
- Average number of queries per active users per hour (a number)
- Average data scanned per query (as a percentage)

Concluding from the previous information that data disk space and number of concurrent queries are limiting factors, and that they should be monitored continuously to answer the following questions related to capacity planning:

- Is the used disk space still sufficient below the available disk space? Compare the available space with the used space.
- How long will it take until the system runs out of space? Determine the growth rate of the used space.
- Is the number of concurrent queries below the concurrency level of the DB2 Analytics Accelerator system? Compare the number of running and queued queries.
- Is the elapsed time of the queries still acceptable? Consider the queue depth and the wait times.

The monitoring counters presented in 9.1, "Performance monitoring and capacity planning" on page 188 can be used to estimate by when a running DB2 Analytics Accelerator system is at its limits. Three approaches are covered regarding disk space usage, query workload, and accelerator wait times.

To plan capacity of a running system, it is essential to have a kind of threshold or limit as a decision point. If the limit is approached or exceeded, the system is in need of an upgrade. Defining this threshold often depends on system design or service levels.

9.2.1 Disk space usage

The disk space is dependent on the DB2 Analytics Accelerator model, as shown before. For instance, if you have a *Striper N2001 full rack*, you have an available disk space of 48 TB (uncompressed 192 TB assuming a compression factor of 4).

Example

Assume that the initial sizing for the system was based on the following values regarding space requirements:

- Size of uncompressed data: 50 TB (physical 12.5 TB with the assumption of a compression rate of 4)
- Annual growth rate: 20%
- Useful life of the system: 5 years
- Peak storage usage: 85%

Based on these assumptions, Figure 9-6 on page 198 shows a graph of the baseline regarding disk space growth. 31.104 TB is the peak storage usage. Therefore, the available storage size must be greater than (>) 36.590 TB. A Striper N2001 has 48 TB and would be in this case the best choice, because the next smaller model has only 24 TB storage size.

Figure 9-6 shows a graph of baseline disk space growth.

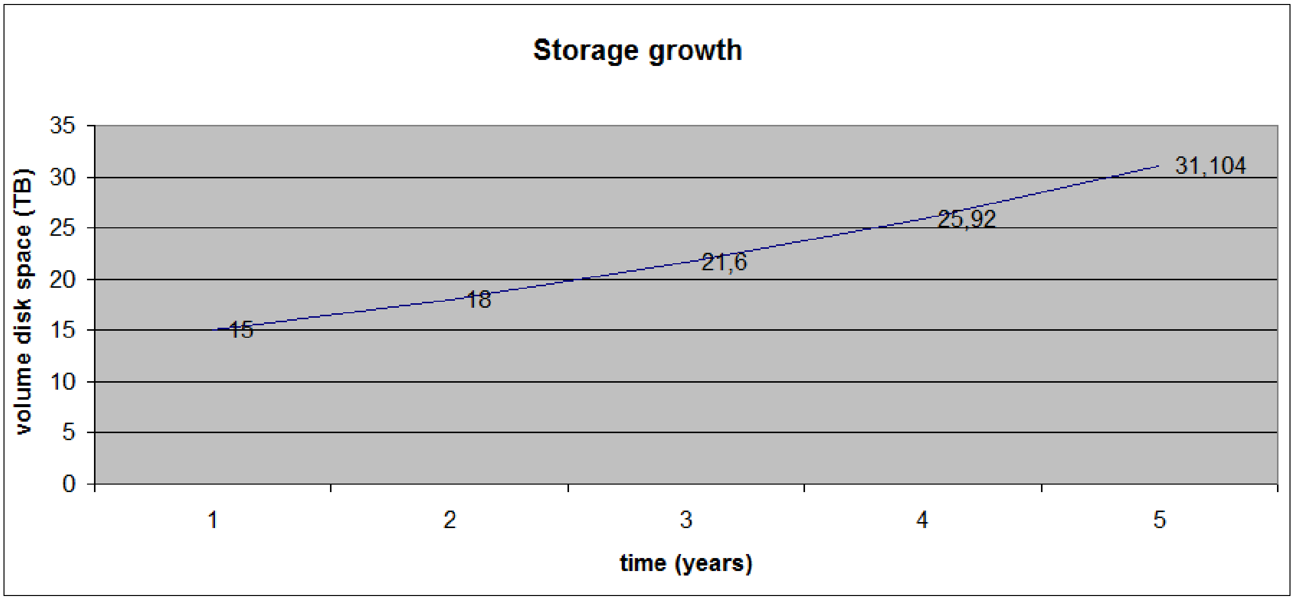


Figure 9-6 Storage growth

To see if the growth rate of the used storage is actually the same as estimated, you can monitor the space usage, for instance on a weekly basis, and calculate the actual growth rate. This can be done by monitoring the counters in Table 9-1.

Table 9-1 Methods and counters to estimate storage growth

Methods	**Counters**
DB2 Analytics Accelerator Studio	Used space / available space
DB2 command (**`-DIS ACCEL DETAIL`**)	Disk storage in use for all DB2 systems / Total disk storage
OMEGAMON PE	Disk storage available / in use for Accel DB - all DB2

Figure 9-7 shows the actual storage growth.

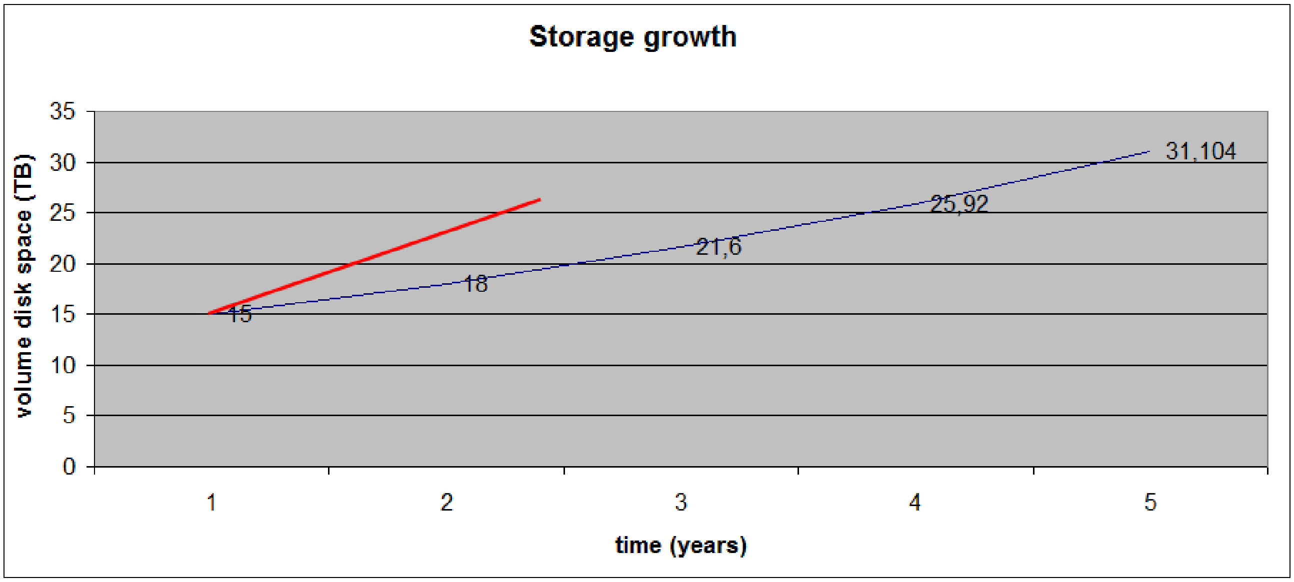


Figure 9-7 Actual storage growth

If it turns out that the actual growth rate is higher (red line in Figure 9-7), you might need to upgrade to a bigger DB2 Analytics Accelerator model earlier, perhaps after 3 years rather than the initial planned 5 years, or analyze which data can be removed from DB2 Analytics Accelerator.

9.2.2 Query workload

A maximum level of concurrency exists regarding queries on DB2 Analytics Accelerator. This is the upper limit of the number of queries that can be active on an accelerator. With DB2 Analytics Accelerator V3.1, this level was 100 queries. It does not mean that all 100 can be run in parallel. Up to 100 queries can be active on the system in one of the following states:

QUEUED Not yet processed in DB2 Analytics Accelerator server.

RUNNING Currently processed in DB2 Analytics Accelerator, however fetching has not started because no results are available yet.

FETCHING DB2 client is fetching rows. In case the query was spilled to disk, it might no longer be running in the DB2 Analytics Accelerator engine, but results might be fetched from disk on the host.

With DB2 Analytics Accelerator V4.1, this limit was changed. It is now dependent on the number of data slices in the system, and therefore dependent on the DB2 Analytics Accelerator model.

It can be calculated using the following formula:

```
Max(100; Min(240; # of data slices))
```

For some of the PureData for Analytics models, the limit is shown in Table 9-2 for N1001 (Twinfin) and Table 9-3 for N2001/2002 (Striper).

Table 9-2 Maximum level of concurrency for N1001

N1001	005	010	020	040
Cabinets	1/2	1	2	4
# of data slices	46	92	184	368
Limit	100	100	184	240

Table 9-3 Maximum level of concurrency for N2001

N2001/2002	005	010	020	040
Cabinets	1/2	1	2	4
# of data slices	120	240	480	960
Limit	120	240	240	240

If this limit is reached, queries sent to DB2 Analytics Accelerator are rejected and an error message is returned, or the query is run directly in DB2 in case the special register QUERY ACCELERATION is set to ENABLEWITHFAILBACK.

If this limit is continuously reached or exceeded, it is an indicator to upgrade to a bigger accelerator model, or to add another accelerator.

In the remainder of this section, we look at a method to monitor this concurrency limit.

We need to check the number of active queries on DB2 Analytics Accelerator. The methods and the counters to do so are shown in Table 9-4.

Table 9-4 Monitoring the concurrency limit

Methods	Counters
DB2 Analytics Accelerator Studio	Number of active / queued queries (#). The number of accelerated queries that are currently being processed. In parentheses, you see the number of queued queries.
DB2 command (`-DIS ACCEL DETAIL`)	ACTV The current number of active, accelerated requests for this DB2 system. QUED The current number of queued, accelerated requests for this DB2 system.
OMEGAMON PE	CURRENTLY running QUERIES The number of currently running queries in DB2 Analytics Accelerator. This includes the queries from all the DB2 systems connected to this Accelerator. CURRENT QUEUE LENGTH The current number of queued, accelerated requests.

The counters in Table 9-4 are updated every 20 seconds. Therefore, if you store them over a day, you have a good picture on the number of queries that are on the system.

As an example, Figure 9-8 shows the concurrent requests.

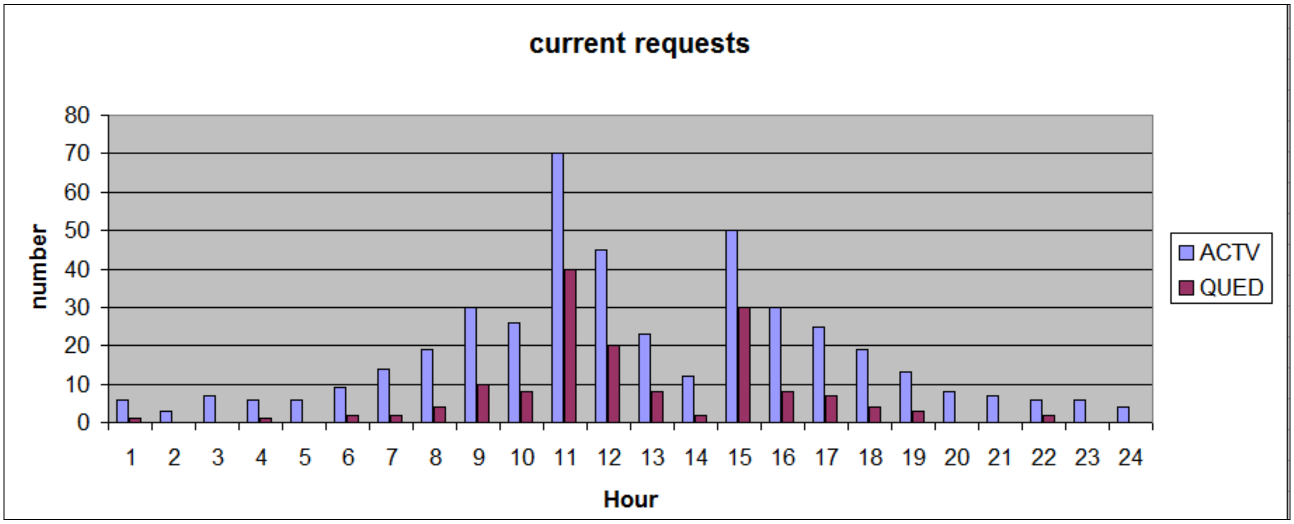


Figure 9-8 Number of active queries running on an accelerator during 24 hours

Assume an accelerator with a limit of 100 queries. In Figure 9-8, the maximum number of active requests is 70 (counter ACTV). At first glance this is good, because it is below 100. But that data is snapshot data, so it is not certain if there are peaks that are already above 100. Therefore, the question becomes, "Is there a kind of a threshold or upper limit that can be used as a decision point for upgrading to a new model?"

To find such a threshold, we can use concepts called peak-to-average ratio (PAR) and saturation design-point (SDP). These concepts are described in more detail in *OS/390 MVS Parallel Sysplex Capacity Planning*, SG24-4680. They are also mentioned in *Capacity Planning for Business Intelligence Applications: Approaches and Methodologies*, SG24-5689, where they are used in context of CPU usage.

Assume that the peak time is between 10 - 11 a.m. and 4 - 5 p.m. The maximum number of requests in those intervals is 70. The average during the day (for example, 7 a.m. to 7 p.m.) is 28.5 requests.

We can consider the limit of 100 requests as the design immanent limit, and can now calculate the threshold:

```
PAR = 70/28.5 = 2.5
Threshold = SDP/PAR = 100/2.5 = 40
```

Therefore, if the average between 7 a.m. and 7 p.m. reaches 40, it is likely that during peak times queries are rejected. In this case, it might be advisable to upgrade to a bigger accelerator model, or to add another accelerator. This threshold can be monitored with the counters mentioned previously by storing their values over the day, and by calculating the average.

Note: The MAXIMUM running QUERIES cannot be used to find the maximum at peak times, because this counter is the maximum number of queries running in DB2 Analytics Accelerator at any time since the accelerator started. This is a *high watermark*.

Figure 9-8 also shows the number of queued requests. This is the share of active queries that is currently queued. If this share is growing over time, in particular in peak times, it might be an indicator for increasing wait times and therefore elapsed times.

9.2.3 Wait times

It is likely that service level agreements (SLAs) are in place regarding response time limits of a system or an application. To be able to satisfy those limits, it is essential to monitor the system. Sometimes it is necessary to monitor the elapsed time on a statement level. A general way to get an indication of rising elapsed times is to monitor wait times on a system level.

Table 9-5 shows the methods and counters that can be used to monitor wait times.

Table 9-5 Monitoring wait times

Methods	Counters
DB2 Analytics Accelerator Studio	Average wait time (ms). Average time that queries had to stay in the queue. The basis for the calculation is all queries that were processed during the last minute.
DB2 command (**`-DIS ACCEL DETAIL`**)	AVERAGE QUEUE WAIT. The length of the average queue wait time in DB2 Analytics Accelerator, measured in ms.
OMEGAMON PE	WAIT TIME IN ACCELERATOR. The accumulated wait time spent in DB2 Analytics Accelerator when running requests from the DB2 subsystem.

The counters mentioned in Table 9-5 are updated every 20 seconds. Storing them in a spreadsheet tool is useful to realize trends. For instance, OMEGAMON PE has the capability to capture data and store it in the tool's Performance database, or unload them in `.csv` format to further process the data in spreadsheet applications.

Similar to the previous section is the question about which upper limit for wait times to use as a decision point for taking action, such as upgrading to a bigger accelerator model or adding another accelerator to benefit from workload balancing.

We can use the same approach as in the previous section: Calculation of a threshold using the concept of PAR and SDP.

The challenge here is the SDP, which does not exist as a system-designed measurement parameter. However, we can consider an SLA as a parameter of the same kind (for example, peak periods 10:00 - 12:00 a.m. and 14:00 - 16:00 p.m.).

In Figure 9-9, the maximum average wait time in the peak periods is 35 ms.

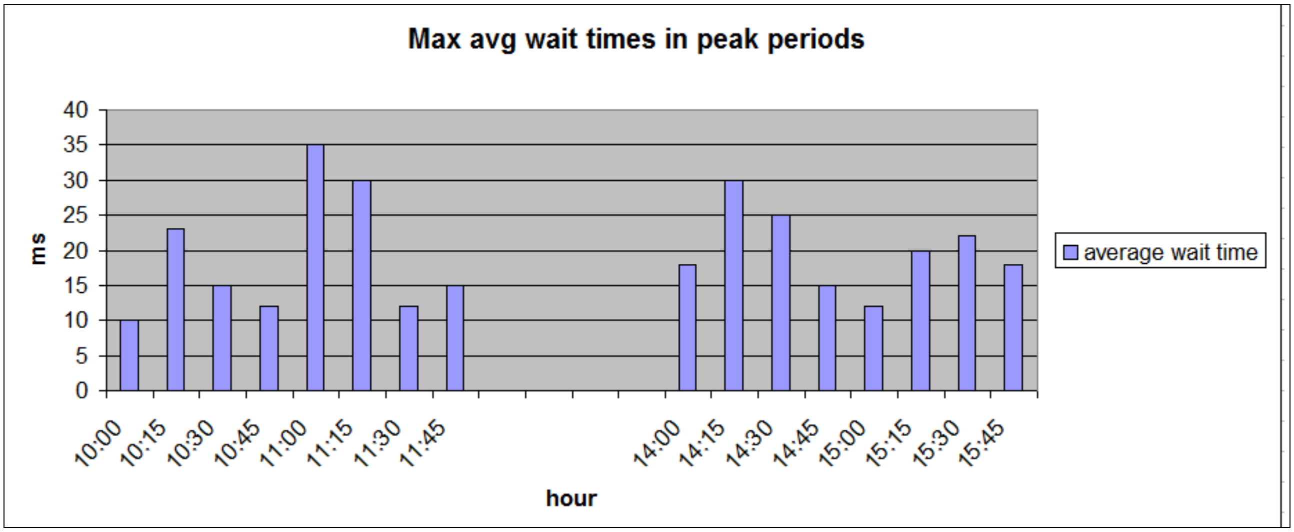


Figure 9-9 Maximum average wait time

Figure 9-10 shows the average wait times per hour during 24 hours.

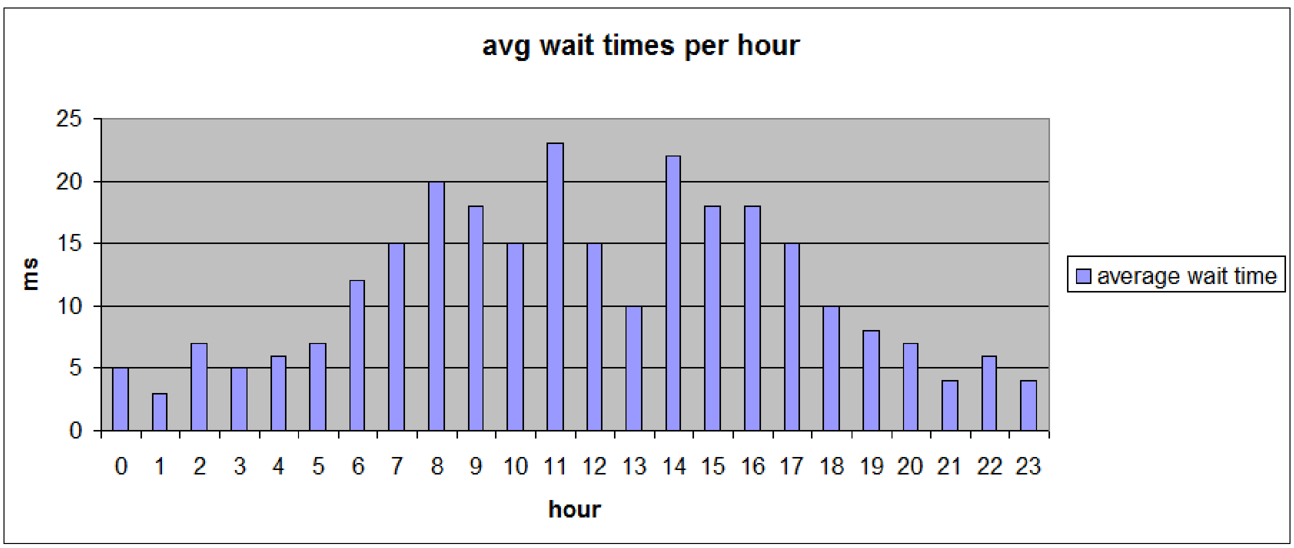


Figure 9-10 Average wait time per hour

The average during the day (7 a.m. - 7 p.m.) can be calculated. Using the numbers in Figure 9-10, it is 16 ms.

Assume that you can afford 100ms wait time regarding the SLA for response times. This value can then be used as the saturation point. The upper limit as a decision point can be determined using the following equations:

```
PAR = 35/16 = 2.19
Threshold = SDP/PAR = 100/2.19 = 46
```

If the average between 7 a.m. and 7 p.m. reaches 46, it is likely that during peak times the response time for queries is above the SLA. This threshold can be monitored with the counters mentioned previously by storing their values over the day, and by calculating the average.

9.3 Assessing existing DB2 workloads

Moving additional workload on an accelerator affects capacity planning. As a first step, it would be important to determine if the identified workload benefits from acceleration. Methods to assess workload regarding acceleration potential are outlined in this section.

9.3.1 Accelerator modeling

A method is available to assess and estimate the benefits of existing DB2 workloads running on an accelerator. This method is called *Accelerator modeling*, and does not require a DB2 Analytics Accelerator. It is based on analyzing DB2 trace data captured in System Management Facilities (SMF) records.

DB2 estimates the CPU cost and elapsed time for queries eligible for offload to DB2 Analytics Accelerator. The process works with SMF data record 101. The estimated cost is captured in fields contained in DB2 accounting record IFCID 3.

This method is described in the following list:

- Accelerator modeling is available in DB2 11 for z/OS. To enable the DB2 Analytics Accelerator *modeling* feature for your DB2 10 for z/OS subsystem, apply the following program temporary fixes (PTFs):
  - PTF UK97693 for PM90886 for dynamic SQL assessment, included in program update tape 1310 (PUT1310)
  - PTF UK98716 for PM95035 for static SQL assessment (included in PUT1311)
- Add the new system parameter ACCELMODEL. This parameter can be changed online and specifies that new fields in the accounting trace are filled, indicating potential query routing to an accelerator. Valid values are YES and NO.
- In case you are running a data sharing group, repeat the steps for all members of the group. Any member that does not have the PTFs installed and ACCELMODEL defined will not be able to capture proper data for an assessment.
- In case there is an accelerator already installed, you cannot run accelerator modeling when query acceleration is enabled. Make sure that query acceleration for your environment or application is disabled in one of the following ways:
  - Subsystem parameter QUERY_ACCELERATION and special register CURRENT QUERY ACCELERATION are not set (the same as in an environment without DB2 Analytics Accelerator configured).
  - Subsystem parameter QUERY_ACCELERATION is set to NONE, and you do not use special register CURRENT QUERY ACCELERATION.
  - Special register CURRENT QUERY ACCELERATION is set to NONE.
- Set the new subsystem parameter ACCELMODEL (online changeable) to YES.
- For dynamic SQL statement assessment, use the RUNSTATS utility for target table spaces, or restart DB2 to invalidate corresponding dynamic statement cache entries. ACCELMODEL needs to be set to YES during statement execution for accelerator modeling.
- For static SQL statement assessment, you must issue a BIND PACKAGE or REBIND PACKAGE statement for the DB2 packages you plan to include for this workload assessment. You might use option APREUSE(ERROR) with the REBIND PACKAGE statement to make sure that the current access plan remains unchanged.

ACCELMODEL=YES is required during BIND/REBIND only, not at static SQL statement execution time.

You can also use option EXPLAIN(ONLY) when REBINDing packages to obtain information from DSN_QUERYINFO_TABLE about statement eligibility for acceleration.

- Make sure that the accounting trace is started, and that at least classes 1 and 2 are active (use the **DISPLAY TRACE** command to check this). If needed, start the trace by issuing the following command:

  -START TRACE(ACCTG) CLASS(1,2) DEST(xxx)

- Run your workload.
- At the end of your data capture time frame, you can stop trace classes again and reset ACCELMODEL to NO.
- Extract (at least) SMF 101 records from your captured SMF data.

With OMEGAMON PM, you can see the effect on elapsed and CPU times, as shown in Example 9-1.

Example 9-1 Accounting report with ACCEL MODEL

```
                  LOCATION: SHADOWPRB1            OMEGAMON XE FOR DB2 PERFORMANCE EXPERT (V5R1M1)                     PAGE: 3-21
      GROUP: MSPRB1                          ACCOUNTING REPORT - LONG                             REQUESTED FROM: NOT SPECIFIED
     MEMBER: PRBA                                                                                            TO: NOT SPECIFIED
  SUBSYSTEM: PRBA                            ORDER: PLANNAME-PRIMAUTH                             INTERVAL FROM: 09/30/13 02:00:53.26
DB2 VERSION: V10                                  SCOPE: MEMBER                                              TO: 09/30/13 04:57:49.06

PLANNAME: DSNREXX  PRIMAUTH: MOOGMN

SQL DML   AVERAGE    TOTAL    SQL DCL           TOTAL    SQL DDL    CREATE   DROP  ALTER    DYNAMIC SQL STMT       AVERAGE     TOTAL
-------- -------- --------    -------------- --------    ---------- ------ ------ ------    -------------------   --------  --------
SELECT       0.00        0    LOCK TABLE            0    TABLE           0      0      0    REOPTIMIZATION            0.00         0
INSERT       0.00        0    GRANT                 0    CRT TTABLE      0    N/A    N/A    NOT FOUND IN CACHE        0.00         0
 ROWS        0.00        0    REVOKE                0    DCL TTABLE      0    N/A    N/A    FOUND IN CACHE            0.00         0
UPDATE       0.00        0    SET CURR.SQLID        0    AUX TABLE       0    N/A    N/A    IMPLICIT PREPARES         0.00         0
 ROWS        0.00        0    SET HOST VAR.         0    INDEX           0      0      0    PREPARES AVOIDED          0.00         0
MERGE        0.00        0    SET CUR.DEGREE        0    TABLESPACE      0      0      0    CACHE_LIMIT_EXCEEDED      0.00         0
DELETE       0.00        0    SET RULES             0    DATABASE        0      0      0    PREP_STMT_PURGED          0.00         0
 ROWS        0.00        0    SET CURR.PATH         0    STOGROUP        0      0      0    CSWL - STMTS PARSED       0.00         0
                              SET CURR.PREC.        0    SYNONYM         0      0    N/A    CSWL - LITS REPLACED      0.00         0
DESCRIBE     0.00        0    CONNECT TYPE 1        0    VIEW            0      0      0    CSWL - MATCHES FOUND      0.00         0
DESC.TBL     0.00        0    CONNECT TYPE 2        0    ALIAS           0      0    N/A    CSWL - DUPLS CREATED      0.00         0
PREPARE   5999.00    65989    SET CONNECTION        0    PACKAGE       N/A      0    N/A
OPEN      5999.00    65989    RELEASE               0    PROCEDURE       0      0      0
FETCH    99820.55  1098026    CALL                  0    FUNCTION        0      0      0
 ROWS    93821.82  1032040    ASSOC LOCATORS        0    TRIGGER         0      0    N/A
CLOSE     5998.73    65986    ALLOC CURSOR          0    DIST TYPE       0      0    N/A
                              HOLD LOCATOR          0    SEQUENCE        0      0      0
DML-ALL     117.8K  1295990   FREE LOCATOR          0    TRUST. CTX      0      0      0
                              DCL-ALL               0    ROLE            0      0    N/A
                                                         JAR           N/A    N/A      0
                                                         MASK/PERM       0      0      0

                                                         TOTAL           0      0      0
                                                         TRUNC TBL       0
                                                         RENAME TBL      0
                                                         RENAME IX       0
                                                         COMMENT ON      0
                                                         LABEL ON        0

MEASURED/ELIG TIMES  APPL (CL1)   DB2 (CL2)   RID LIST                  AVERAGE     TOTAL    DATA CAPTURE        AVERAGE     TOTAL
-------------------  ----------  ----------   ----------------------   --------  --------    -----------------  --------  --------
ELAPSED TIME         3:54.72157  3:30.03555   USED                         0.00         0    IFI CALLS MADE          N/P       N/P
 ELIGIBLE FOR ACCEL         N/A  3:11.32116                                                  RECORDS CAPTURED        N/P       N/P
                                              FAIL-NO STORAGE              0.00         0    LOG RECORDS READ        N/P       N/P
CP CPU TIME          1:25.32915  1:09.39248   FAIL-LIMIT EXCEEDED          0.00         0    ROWS RETURNED           N/P       N/P
 ELIGIBLE FOR SECP     0.000000         N/A                                                  RECORDS RETURNED        N/P       N/P
 ELIGIBLE FOR ACCEL         N/A   59.842250   INTERRUPTED-NO STORAGE       0.00         0    DATA DESC. RETURN       N/P       N/P
                                              INTERRUPTED-LIMIT EXC.       0.00         0    TABLES RETURNED         N/P       N/P
SE CPU TIME            0.000000    0.000000                                                  DESCRIBES               N/P       N/P
 ELIGIBLE FOR ACCEL         N/A    0.000000   OVERFLOWED-NO STORAGE        0.00         0
                                              OVERFLOWED-LIMIT EXC.        0.00         0
```

In Figure 9-11, we extracted the ACCEL MODEL values.

```
MEASURED/ELIG TIMES   APPL (CL1)   DB2 (CL2)
-------------------   ----------   ----------
ELAPSED TIME          3:54.72157   3:30.03555
 ELIGIBLE FOR ACCEL          N/A   3:11.32116

CP CPU TIME           1:25.32915   1:09.39248
 ELIGIBLE FOR SECP      0.000000          N/A
 ELIGIBLE FOR ACCEL          N/A    59.842250

SE CPU TIME             0.000000     0.000000
 ELIGIBLE FOR ACCEL          N/A     0.000000
```

Figure 9-11 ACCEL MODEL values

The reported fields of OMEGAMON PE accounting report with ACCEL MODEL are as follows[1]:

ELAPSED TIME Regular total class 1 and 2 elapsed times of the parent only.

ELIGIBLE FOR ACCEL

Elapsed time spent in DB2 for SQL eligible for acceleration. If the statements are run in parallel, the elapsed time relates to the parent task only.

CP CPU TIME Regular total class 1 and 2 CPU times, inclusive of all parallel threads eligible times for specialty engine processing.

ELIGIBLE FOR SECP

The part of CPU time spent on general-purpose processors for SQL eligible for specialty engine.

ELIGIBLE FOR ACCEL

The part of CPU time spent on general-purpose processors for SQL eligible for acceleration. If the statements are run in parallel, the CPU value includes the parent and all the subordinated parallel tasks.

SE CPU TIME Regular total class 1 and 2 specialty engine CPU times, inclusive of all parallel threads.

ELIGIBLE FOR ACCEL

The part of CPU time spent on specialty engine processors for SQL eligible for acceleration. If the statements are run in parallel, the CPU saving includes the parent and all of the subordinated parallel tasks.

In Figure 9-11, you can see how much CPU time is eligible to be offloaded from the mainframe, because the workload is processed on DB2 Analytics Accelerator. This is an estimate and must be considered as the potential maximum of CPU time that might be saved. There is DB2 processing associated with preparing and sending the statement to DB2 Analytics Accelerator and, particularly, processing associated with receiving the result set and passing it back to the application. The latter can be significant in very large result sets.

[1] DB2 calculates the eligible values for QUERY_ACCELERATION=ENABLE and not for QUERY_ACCELERATION=ELIGIBLE.

Figure 9-12 illustrates how the information flows between DB2 and DB2 Analytics Accelerator.

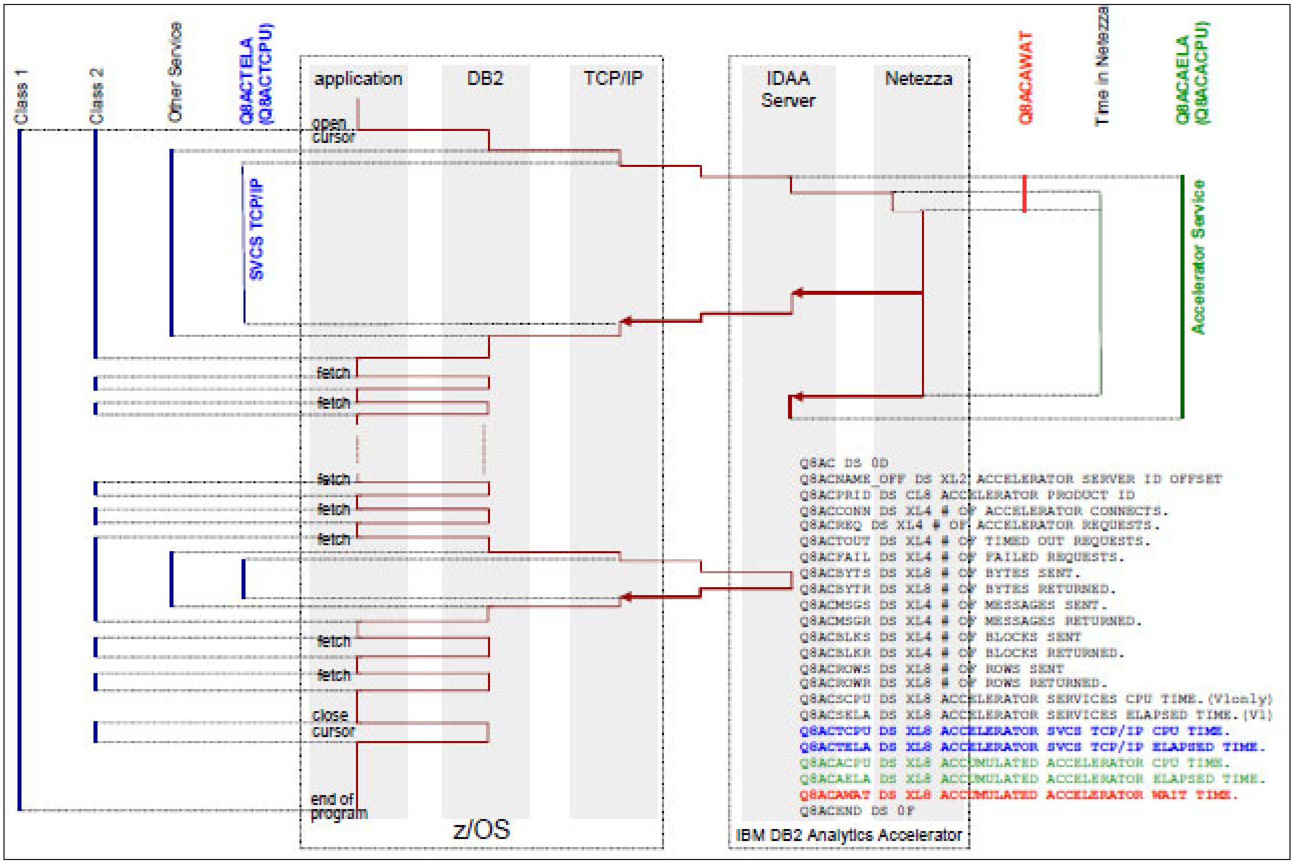


Figure 9-12 DB2 accounting trace fields and DB2 Analytics Accelerator

Q8ACWAT holds the accumulated accelerator wait time, which consists of a preprocessing time in DB2 Analytics Accelerator server and wait time in Netezza.

Q8ACACPU holds the accumulated accelerator CPU time, which accumulates during the elapsed time and in particular might be higher than the accumulated elapsed time due to parallel processing.

Again, the following link provides more information about OMEGAMON PE:

`http://www.ibm.com/software/products/en/tivoliomegamonxefordb2performancemonitoronzos/`

To evaluate if a statement is eligible for acceleration, you can run an EXPLAIN command. An EXPLAIN command for a statement that goes through accelerator modeling will have the following output:

- If the query is eligible for offload:

  `DSN_STATEMNT_TABLE.REASON = 'ACCELMODEL ELIGIBLE'`

- If the query is not eligible for offload:

  `DSN_STATEMNT_TABLE.REASON = 'ACCELMODEL NOT ELIGIBLE'`

9.3.2 Optim Query Workload Tuner

Another tool that can be used to assess workloads is IBM Optim Query Workload Tuner for z/OS. Its assessment is based on captured SQL statements.

The Workload Analytics Acceleration Advisor function provides information about estimated improvements for a DB2 workload if run on an accelerator. See the example in Figure 9-13.

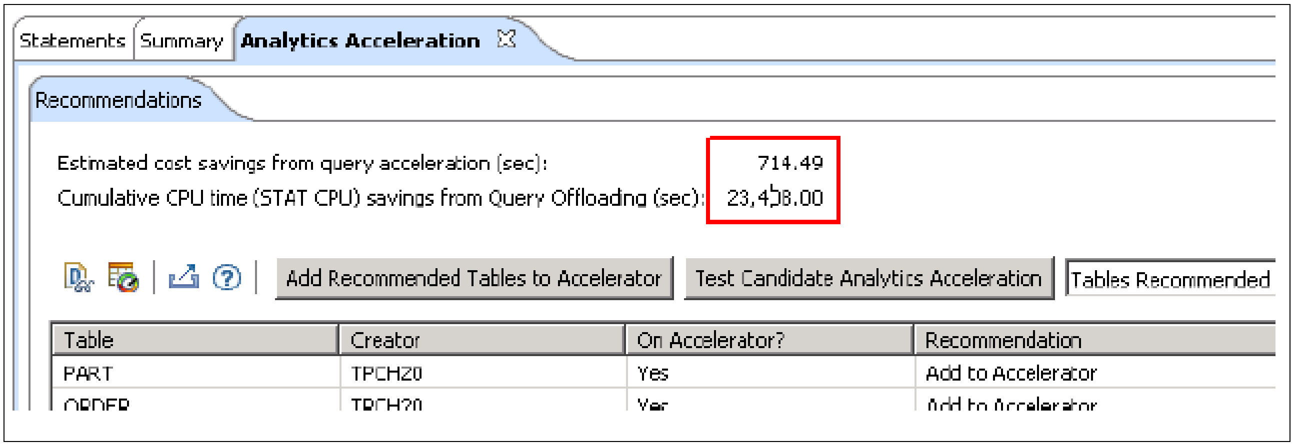


Figure 9-13 Estimated improvements

There are also suggestions available regarding which tables should be added to DB2 Analytics Accelerator so that the workload would benefit, as shown in Figure 9-14.

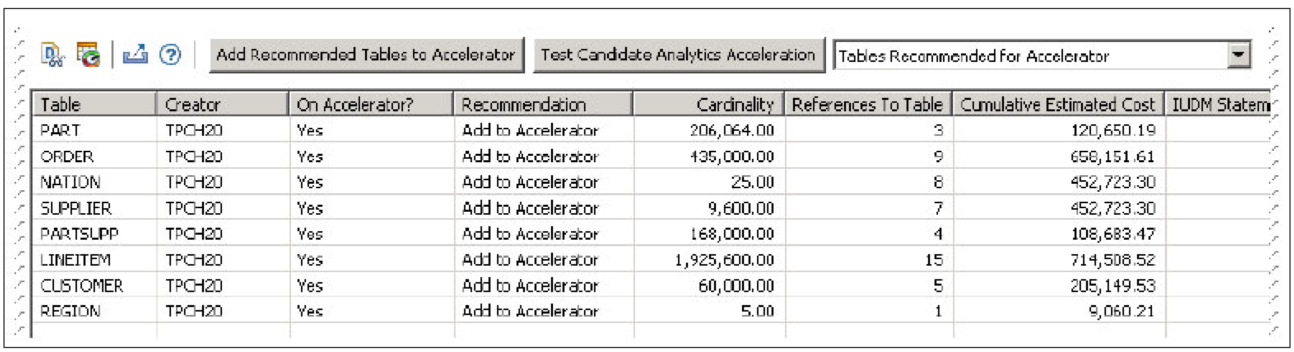


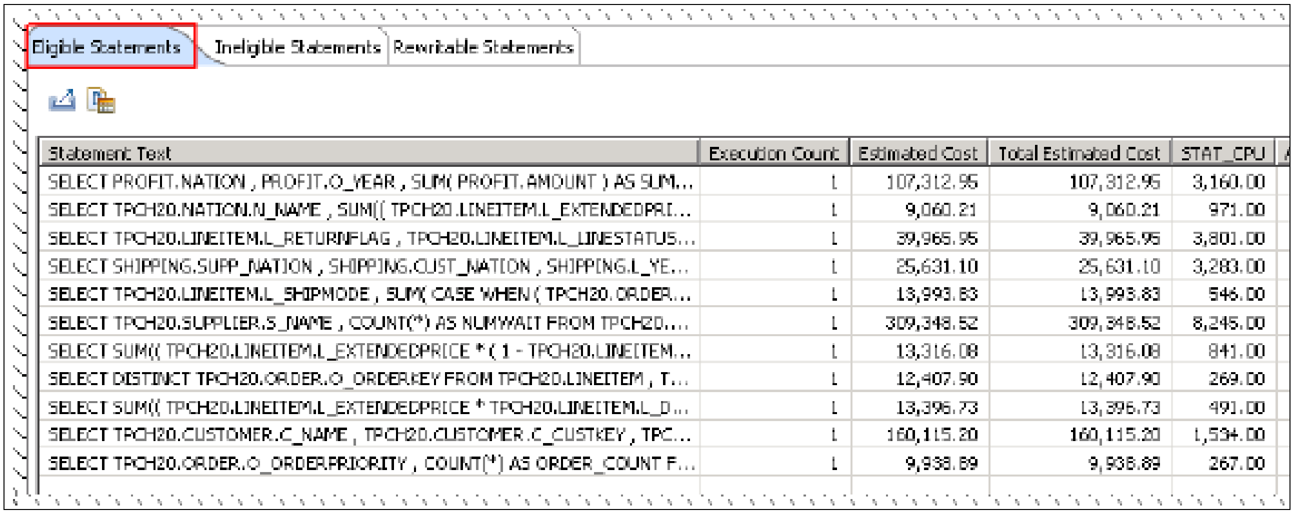
Table	Creator	On Accelerator?	Recommendation	Cardinality	References To Table	Cumulative Estimated Cost	IUDM Statem
PART	TPCH20	Yes	Add to Accelerator	206,064.00	3	120,650.19	
ORDER	TPCH20	Yes	Add to Accelerator	435,000.00	9	658,151.61	
NATION	TPCH20	Yes	Add to Accelerator	25.00	8	452,723.30	
SUPPLIER	TPCH20	Yes	Add to Accelerator	9,600.00	7	452,723.30	
PARTSUPP	TPCH20	Yes	Add to Accelerator	168,000.00	4	108,683.47	
LINEITEM	TPCH20	Yes	Add to Accelerator	1,925,600.00	15	714,508.52	
CUSTOMER	TPCH20	Yes	Add to Accelerator	60,000.00	5	205,149.53	
REGION	TPCH20	Yes	Add to Accelerator	5.00	1	9,060.21	

Figure 9-14 Suggested tables

Further suggestions refer to the SQL statements from the workload. The list of SQL statements is split into three parts:

- Eligible Statements
- Ineligible Statements
- Rewriteable Statements

Eligible Statements shows a list of SQL statements from the workload that will benefit from DB2 Analytics Accelerator, as shown in Figure 9-15.



Statement Text	Execution Count	Estimated Cost	Total Estimated Cost	STAT_CPU
SELECT PROFIT.NATION , PROFIT.O_YEAR , SUM(PROFIT.AMOUNT) AS SUM...	1	107,312.95	107,312.95	3,160.00
SELECT TPCH20.NATION.N_NAME , SUM((TPCH20.LINEITEM.L_EXTENDEDPRI...	1	9,060.21	9,060.21	971.00
SELECT TPCH20.LINEITEM.L_RETURNFLAG , TPCH20.LINEITEM.L_LINESTATUS...	1	39,965.95	39,965.95	3,801.00
SELECT SHIPPING.SUPP_NATION , SHIPPING.CUST_NATION , SHIPPING.L_YE...	1	25,631.10	25,631.10	3,283.00
SELECT TPCH20.LINEITEM.L_SHIPMODE , SUM(CASE WHEN (TPCH20.ORDER...	1	13,993.83	13,993.83	546.00
SELECT TPCH20.SUPPLIER.S_NAME , COUNT(*) AS NUMWAIT FROM TPCH20....	1	309,348.52	309,348.52	8,245.00
SELECT SUM((TPCH20.LINEITEM.L_EXTENDEDPRICE * (1 - TPCH20.LINEITEM...	1	13,316.08	13,316.08	841.00
SELECT DISTINCT TPCH20.ORDER.O_ORDERKEY FROM TPCH20.LINEITEM , T...	1	12,407.90	12,407.90	269.00
SELECT SUM((TPCH20.LINEITEM.L_EXTENDEDPRICE * TPCH20.LINEITEM.L_D...	1	13,396.73	13,396.73	491.00
SELECT TPCH20.CUSTOMER.C_NAME , TPCH20.CUSTOMER.C_CUSTKEY , TPC...	1	160,115.20	160,115.20	1,534.00
SELECT TPCH20.ORDER.O_ORDERPRIORITY , COUNT(*) AS ORDER_COUNT F...	1	9,938.89	9,938.89	267.00

Figure 9-15 Eligible statements

Additionally, DB2 Query Monitor for z/OS and OMEGAMON PE are both integrated with Optim Query Workload Tuner. Both tools, DB2 Query Monitor and OMEGAMON PE, capture workload on a SQL statement level. Both have the capability to transfer this workload to Optim Query Workload Tuner. For more information about these tools see the following links:

- Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS:

  `http://www.ibm.com/software/products/en/tivoliomegamonxefordb2performanceexpertonzos/`

- DB2 Query Monitor for z/OS:

  `http://www.ibm.com/software/products/de/db2-query-monitor-for-zos/`

- Optim Query Workload Tuner:

  `http://www.ibm.com/software/products/en/optiquerworktunefordb2forzos`

9.3.3 Manual assessment of query workloads

With DB2 Analytics Accelerator Studio, it is possible to add a virtual accelerator. It can be used to EXPLAIN queries, but not to run queries. The EXPLAIN provides the information if a query is eligible for acceleration (that queries can benefit from an accelerator). If this check is positive, you might want to know the magnitude of the elapsed time of this query when run on an accelerator in comparison with what might have been measured on DB2.

This can be calculated and depends on the following parameters:

- Number of data slices of the DB2 Analytics Accelerator model
- Scan speed of the DB2 Analytics Accelerator model
- Volume of data scanned
- Assumption of data compression rate

Example

In the example in Figure 9-16, the DB2 Analytics Accelerator Model is a full rack Striper N2001 with 240 data slices and a scan speed of 130 MB per sec.

	A	B	C	D
1	Input parameters			
2	Model		Striper N2002	
3	Model Size (number of racks)		1	
4	Number of Data Slices		240	
5	per Slice scan Speed		130	MB/sec
6				
7				
8	Result			
9	**Table Name**	**Compressed Data Volume in GB**	**Compressed Data Volume per Data Slice in MB**	**Time in sec**
10	table1	50	213,33	1,64
11	table2	20	85,33	0,66
12	table3	10	42,67	0,33

Figure 9-16 Elapsed time calculation

The elapsed time for three queries was calculated. Each of them is accessing a single table. The calculation is based on the compressed size per data slice.

The result is a rough magnitude of the response time. The actual elapsed time is higher, because the internal processing time for broadcasts, redistributions, or sorts, and the time to return the data to DB2 was not considered. In particular, the size of the result set that is returned to DB2 can affect the elapsed time significantly.

Also note that for complex joins, where lots of redistributions happen internally, the numbers might be different. For queries relying on scan rates only (such as table scans), these numbers apply.

9.4 Measuring actual benefits of moving workloads to DB2 Analytics Accelerator

If DB2 Analytics Accelerator runs in a production environment, it would be interesting how the estimated benefits could actually be measured. Therefore, you need to compare captured data:

- Capture SMF data running a DB2 workload without an active accelerator.
- Capture SMF data running the same workload with an active accelerator.

For instance, this can be done for dynamic queries by setting the special register:

- `SET CURRENT QUERY ACCELERATION NONE`

  The workload is run on DB2.

- `SET CURRENT QUERY ACCELERATION ELIGIBLE`

  The workload is run on DB2 Analytics Accelerator.

Then, compare the captured data.

9.4.1 Measuring with OMEGAMON Performance Expert

With OMEGAMON PE, it is possible to measure the benefits at DB2 plan level with accounting reports. You can compare CPU Class 2 times with and without an accelerator. With an active accelerator, the CPU Class 2 times can be found in the DB2 Analytics Accelerator section of the accounting report, as shown in Figure 9-17.

```
PAGE: 1-24
         REQUESTED FROM: NOT SPECIFIED
                     TO: NOT SPECIFIED
          INTERVAL FROM: 02/13/12 15:30:30.48
                     TO: 02/13/12 16:38:44.53
 PLANNAME: db2jcc_a
  ACCELERATOR       AVERAGE        TOTAL
 ------------  ------------  ------------
 ELAPSED TIME
 SVCS TCP/IP    15:50.410018  15:50.410018
 ACCUM ACCEL    15:50.247750  15:50.247750
 CPU TIME
 SVCS TCP/IP        0.000430      0.000430
 ACCUM ACCEL        5.823999      5.823999
 WAIT TIME
 ACCUM ACCEL        0.004668      0.004668

 DB2 THREAD
 CLASS 1
 ELAPSED                 N/A  15:50.523562
 CP CPU                  N/A      0.004088
 SE CPU                  N/A      0.000000
 CLASS 2
 ELAPSED                 N/A           N/P
 CP CPU                  N/A           N/P
 SE CPU                  N/A           N/P
```

Figure 9-17 OMEGAMON PE accounting report

If you run the workload using accelerator modeling, the CPU Class 2 times can be found in the Accelerator Modeling section of the accounting report. See Figure 9-18. This is useful if you want to compare the estimated savings with the actually measured savings.

```
MEASURED/ELIG TIMES  APPL (CL1)   DB2 (CL2)
------------------  ----------   ----------
ELAPSED TIME          4.830139    4.740227
 ELIGIBLE FOR ACCEL        N/A    4.442327

CP CPU TIME           6.337894    6.336111
 ELIGIBLE FOR SECP    4.990042         N/A
 ELIGIBLE FOR ACCEL        N/A    6.329119

SE CPU TIME           0.000000    0.000000
 ELIGIBLE FOR ACCEL        N/A    0.000000
```

Figure 9-18 Accounting report - Accelerator Modeling

As already mentioned, with this monitoring tool it is also possible to save that data in `.csv` format and compare the CPU times in spreadsheet applications.

9.4.2 Measuring using query history

Another method to measure the benefits is to use the information in the query history table. This is a method based on SQL statements running on DB2 Analytics Accelerator. 9.1.4, “Monitoring using query history” on page 193, describes how to create the query history table, and how to run the stored procedure that populates this table.

After you have the query history, follow these steps:

1. Identify a representative workload (SQL statements), and extract it from the query history table.
2. Then rerun this workload on DB2 (without acceleration) and measure the CPU cost in DB2 (with OMEGAMON PM).

Now the information is available about how much CPU the workload uses on DB2 Analytics Accelerator and on DB2.

It is possible to calculate a multiplier from CPU cost on DB2 Analytics Accelerator to CPU cost on DB2. This multiplier can be used to compute the cost savings. For every query that runs on DB2 Analytics Accelerator during mainframe peak times, compute the CPU savings on the mainframe. This can also be used to determine the million service units (MSU) reduction on the mainframe.

With this multiplier, it is also possible to create reports on the savings over time. For instance, how much percentage of total available mainframe CPU would this workload use when it is run on DB2, and not on DB2 Analytics Accelerator, which is currently the case.

9.4.3 DB2 Query Monitor

With DB2 Query Monitor, it is possible to compare workloads regarding CPU or elapsed time. It has a *Compare* function that can be used to compare the DB2 CPU use with and without an accelerator. It is even possible to do this comparison on a SQL statement level.

You can compare workload on an interval basis.

For more information about DB2 Query Monitor for z/OS, see the following website:

http://www.ibm.com/software/products/en/db2-query-monitor-for-zos

10

DB2 Analytics Accelerator Loader

In this chapter, we focus on the newest accelerator tool, the IBM DB2 Analytics Accelerator Loader for z/OS V1.1. Learn about the use cases DB2 Analytics Accelerator Loader was designed to address, and how it works, to better understand how to use its features and functions. This chapter provides information about the following topics:

- DB2 Analytics Accelerator Loader overview
- Customizing Accelerator Loader
- DB2 Analytics Accelerator Loader intercept and started task
- Group Consistent Load
- Dual load
- Current limitations and considerations
- First steps to troubleshooting

10.1 DB2 Analytics Accelerator Loader overview

DB2 Analytics Accelerator Loader for z/OS V1.1 (5639-OLA) enables you to load operational DB2 for z/OS data into IBM DB2 Analytics Accelerator for z/OS. This minimizes the application effect when loading related tables into DB2 Analytics Accelerator. It also gives you the flexibility to specify a historical point-in-time to which you want a consistent set of data to be loaded into DB2 Analytics Accelerator to meet your business needs. DB2 Analytics Accelerator Loader uses existing recovery assets to establish this consistent point-in-time.

You can also load non-DB2 data to DB2 Analytics Accelerator, including non-mainframe. Sources can include data from other IBM DB2 for z/OS systems, Virtual Storage Access Method (VSAM), IBM IMS, and non-mainframe data to facilitate enterprise business analytics on IBM System z.

After you have extracted data from another source into a format that is compatible with IBM DB2 LOAD utility, DB2 Analytics Accelerator Loader can then load that data into DB2 Analytics Accelerator, and optionally into DB2 for z/OS. Loading data into both DB2 Analytics Accelerator and into DB2 for z/OS is performed in parallel, which can improve availability and reduce system resource use, such as central processing unit (CPU) usage.

If you choose to load non-DB2 data directly into DB2 Analytics Accelerator and not into DB2 for z/OS, you can eliminate the time and CPU use of loading this data into DB2. Use of this feature still requires some administrative tasks to be performed.

These tasks include creating the DB2 tables on System z, adding the table to DB2 Analytics Accelerator, and enabling it for acceleration. Additional tasks include routing application queries to DB2 Analytics Accelerator to ensure that all queries for that table are routed to DB2 Analytics Accelerator rather than the table on DB2 for z/OS. A client considering using this feature should use caution and engage IBM for guidance.

DB2 Analytics Accelerator Loader V1.1 uses the speed and efficiency of IBM FlashCopy® by optionally enabling you to create a new FlashCopy consistent image copy of the data being loaded from DB2 for z/OS into DB2 Analytics Accelerator. DB2 Analytics Accelerator Loader uses this consistent copy as input to load the data into DB2 Analytics Accelerator, which eliminates the need to lock the production tables for update during the load.

10.2 Customizing Accelerator Loader

DB2 Analytics Accelerator Loader installs using standard System Modification Program Extended (SMP/E) for z/OS. The product options are then customized through the IBM Tools Customizer for z/OS (TCz). However, mandatory pre-customization tasks must be completed before the TCz-generated jobs can be run.

Detailed insight on customizing DB2 Analytics Accelerator Loader product options through TCz is covered in the *IBM DB2 Analytics Accelerator Loader for z/OS User's Guide*, SC19-4165, and is not discussed here. Instead, the customization process focuses on the pre- and post-customization tasks that are required to facilitate a smooth installation.

10.2.1 Pre-customization steps

Running the TCz batch jobs is only one part of the customization process. These pre-customization tasks are as important as defining the product options through TCz. However, because these require planning, each is discussed in more detail. The items listed in this subsection must be addressed before the TCz-generated customized jobs can be run, and before the product can be used. Define the product options:

1. Select a dedicated storage area network (SAN) Volume Controller (SVC) number.

   The started task uses an SVC number while it is running. Therefore, an unused SVC number must be obtained from your system programmers for use with a DB2 Analytics Accelerator Loader started task. The started task dynamically allocates the SVC number when it is started, and releases the SVC number when it is stopped.

   However, the started task must still be assigned an SVC number that is not in use, and can be dynamically allocated each time that the started task is brought up. This SVC number is defined in the Product Parameters panel in TCz, and is generated into the Options Module SVC_NUMBER parameter.

2. Grant the started task user ID proper authority.

   A user ID must be defined to the started task so that it can provide services to the user on behalf of DB2 Analytics Accelerator Loader as it processes a batch job. This user ID must be granted both the proper DB2 authorization and IBM Resource Access Control Facility (IBM RACF®) authorization.

   The DB2 authorization must be one of the three following authorizations: SYSADM, SYSCTRL, or SYSOPR with MONITOR1. If SYSOPR with MONITOR1 is chosen as the authorization to grant the started task ID, additional SELECT authority must be granted to several DB2 catalog tables. See the chapter about preparing to customize DB2 Analytics Accelerator Loader in the *IBM DB2 Analytics Accelerator Loader for z/OS User's Guide,* SC19-4165, for more information about each of these authorizations.

   This started task user ID is used as the bind owner when binding DB2 Analytics Accelerator Loader packages and plans, and should not be changed in the TCz-generated BIND job. The started task user ID must also be granted RACF READ access to its own product data sets.

3. Authorized program facility (APF) authorize the product libraries.

   Before the TCz-generated customized batch jobs can be run, DB2 Analytics Accelerator Loader load library must be APF-authorized. If you installed FEC common code into its own target libraries and not into the same libraries as DB2 Analytics Accelerator Loader code, the FEC load library must be APF-authorized as well. Authorizing the product libraries ensures that the program code runs properly as each module is called.

4. Copy load module member DSNUTILF.

   In order for DB2 Analytics Accelerator Loader to be started for the external load feature, DB2 Analytics Accelerator Loader load library must be part of the STEPLIB concatenation for any DB2 LOAD utility batch jobs that you want to load data to DB2 Analytics Accelerator.

   Optionally, if you do not want to change the STEPLIB concatenation of many DB2 LOAD utility batch jobs, module DSNUTILF can be copied from the product's load library to an APF-authorized DB2 library in the STEPLIB concatenation. This method ensures that no changes have to be made to the existing DB2 LOAD utility JCL, and enables DB2 Analytics Accelerator Loader to automatically intercept only those DB2 LOAD utility batch jobs that contain DB2 Analytics Accelerator Loader extended syntax.

5. Edit the IBM Workload Manager (WLM) address space.

   DB2 Analytics Accelerator Loader load library must be concatenated into the WLM address space in which the stored procedure SYSPROC.DSNUTILU is run. This ensures DB2 Analytics Accelerator Loader is properly started when the stored procedure ACCEL_LOAD_TABLES starts DSNUTILU. Because both DB2 Analytics Accelerator Loader and the DB2 Utilities Enhancement Tool intercept invocations of the DB2 LOAD utility, ensure that DB2 Analytics Accelerator Loader load library is concatenated in the WLM address space before the DB2 Utilities Enhancement Tool load library if present.

10.2.2 Customize the product options using TCz

To customize DB2 Analytics Accelerator Loader, you must use TCz. The customization process using TCz is described in the *IBM DB2 Analytics Accelerator Loader for z/OS User's Guide*, SC19-4165.

The TCz process generates customized job control language (JCL) that creates the required components for DB2 Analytics Accelerator Loader to function. These components include but are not limited to the batch JCL jobs that perform the following functions:

- Create the underlying repository
- Bind product packages and plans
- Create the required started task
- Create and populate the DB2 Control File

Batch jobs are appropriately generated for each DB2 subsystem on which you are customizing DB2 Analytics Accelerator Loader. They must be run in sequential order to complete the customization on each DB2 subsystem.

After the customized jobs have been run, there is one last post-customization step required to complete the installation.

10.2.3 Post-customization step

After the TCz-generated HLOSTCJ job has been run to create the started task, it must then be copied from your hlq.SHLOSAMP library into your system PROCLIB. This enables the started task to be started with a system command.

If you plan to monitor invocations of the DB2 LOAD utility across multiple DB2 subsystems that are at different DB2 versions, ensure that the STEPLIB concatenation in the started task specifies the earliest of these DB2 versions as the DSNLOAD library. Otherwise, connection problems might occur when the started task attempts to write information to its logging and auditing tables while monitoring DB2 subsystems other than the subsystems designated as the primary.

10.2.4 Data sharing considerations

If you plan to deploy DB2 Analytics Accelerator Loader in a DB2 data sharing environment, consider the following concepts. Note that all customizations for the components of DB2 Analytics Accelerator Loader are done through TCz. The following components comprise DB2 Analytics Accelerator Loader, and must be customized through the TCz:

Started task	Communicates with the client, intercepts DB2 LOAD utility jobs.
Options Module	Contains product processing options.
Policy	Specifies which subsystems to monitor.
DB2 Control File	Contains information specific to DB2 subsystems.

A DB2 data-sharing group is composed of one or more DB2 subsystems that are located on the same z/OS image, or on different z/OS images. The member subsystems share a common DB2 catalog, and can directly access and change the same data while maintaining data integrity.

The DB2 Analytics Accelerator Loader *started task* can monitor multiple members of a data sharing group that exist on the same logical partition (LPAR). The started task cannot monitor subsystems on other LPARs from where the started task is running. Therefore, if there are members of a data sharing group on another LPAR, a second started task must be configured to run on the second LPAR.

Each LPAR must have its own started task, and each started task must have its own options module and policy. In a data sharing environment, all members of the data-sharing group share a common set of accelerator repository tables on the primary subsystem. In a non-data sharing environment, the primary subsystem contains all of the repository tables, but each additional subsystem that you want to monitor must contain DB2 Analytics Accelerator Loader work tables, which are a subset of the repository tables.

To generate the appropriate JCL for the primary subsystem, select the **This is the primary subsystem** option on the DB2 Parameters panel, as shown in Figure 10-1.

```
TCUSTMZR                         DB2 Parameters                       10:16:23
Command ===>                                                     Scroll ===> CSR

Enter values for all of the DB2 parameters. Press End to save and exit.

Commands: SAVE - Save parameter values

 Product to Customize
   Product metadata library . : HLO.WRK0110.SHLODENU    > LPAR  . . : RS22
   Product name . . . . . . . : DB2 Analytics Accelera  > Version . : 1.1.0
   Configuration ID: HLO      > Description: DB2 Analytics Accelerator Load  >

                                                                      More:     +

 DB2 subsystem ID  . . . . . . . . . . . . : DA1A
 Group attach name . . . . . . . . . . . . :
 This is the primary subsystem . . . . . . . YES (YES, NO)

 General DB2 Information
  Mode . . . . . . . . . . . . . . . . . . . NFM (NFM, CM)
  Level number . . . . . . . . . . . . . . . 101 (101, 111)

 DB2 Libraries
  Load library . . . . . . . . . . . . . . . DA1A.SDSNEXIT       >    More...
  Run library  . . . . . . . . . . . . . . . DSN.VA10.RUNLIB.LOA  >    Add...
  Exit library . . . . . . . . . . . . . . . DA1A.SDSNEXIT       >    Add...
```

Figure 10-1 Option to generate the appropriate repository DDL

This ensures the correct Data Definition Language (DDL) member is generated for the primary DB2 subsystem, and creates all repository tables in the same subsystem identifier (SSID). When customizing a non-data sharing environment, each SSID that you are customizing has only the work table DDL created for them.

Then, on the Product Parameters panel, enter the name you chose for the primary SSID in the **The primary DB2 subsystem ID** field, as shown in Figure 10-2.

```
 TCUSTMZR                  Product Parameters                          10:18:27
  Command ===>                                                      Scroll ===> CSR

  Complete the following tasks to customize the products. The required tasks
  and steps are preselected. Ensure that all parameters are specified for each
  selected step within a task. Press End to save and exit.

  Commands: SAVE - Save parameter values
  Line Commands: / - Select

   Product to Customize
     Product metadata library . : HLO.WRK0110.SHLODENU    > LPAR  . . : RS22
     Product name . . . . . . . : DB2 Analytics Accelera  > Version . : 1.1.0
     Configuration ID: HLO      > Description: DB2 Analytics Accelerator Load
>

  Product customization library .: CSJENN.ALL3.$RS22$.HLO110
                                                                     More:   -
+
    Started task identifier  . . . . . . . . . HLO1

    / Create PROC, PLCY, and other SAMPLIB members
      The primary DB2 subsystem ID . . . . . . ABCD
      The started task user ID . . . . . . . . HLOSTC
      SYSOUT class . . . . . . . . . . . . . . *
      STC audit active . . . . . . . . . . . . YES (YES, NO)
      Maximum STC audit age  . . . . . . . . . 45
      Connect to all DB2 subsystems  . . . . . YES (YES, NO)
```

Figure 10-2 Generate the appropriate options module parameter

This ensures that the correct subsystem value is populated into the *options module* DB2_SSID field to annotate which subsystem contains the full set of repository tables. Only one options module batch job is required per started task. If you need to create more than one started task, each started task must have its own options module, which defines a primary subsystem that contains the full set of repository tables.

The third component is the *policy*. Each member subsystem must be specified in the DSNUTILB intercept policy for each started task by using the <DB2SYSTEM> element. One policy is generated through TCz per started task. If you need a second policy generated for a second started task, you can simply copy the first policy, and edit the subsystem names that you want to monitor.

In this sample policy, shown in Figure 10-3, group DBSG contains 4 members, DB1D, DB2D, DB3D, and DB4D.

```
<DSNUTILB_INTERCEPT>
 <POLICY>

  <DB2SYSTEM SSID="DBSG" ACTION="LOAD_ACCELERATOR">
  </DB2SYSTEM>

  <DB2SYSTEM SSID="DB1D" ACTION="LOAD_ACCELERATOR">
  </DB2SYSTEM>

  <DB2SYSTEM SSID="DB2D" ACTION="LOAD_ACCELERATOR">
  </DB2SYSTEM>

  <DB2SYSTEM SSID="DB3D" ACTION="LOAD_ACCELERATOR">
  </DB2SYSTEM>

  <DB2SYSTEM SSID="DB4D" ACTION="LOAD_ACCELERATOR">
  </DB2SYSTEM>

 </POLICY>
</DSNUTILB_INTERCEPT>
```

Figure 10-3 Each member of a data sharing group must be defined to the policy

Using a policy with each group and member defined ensures that if a DB2 LOAD utility is run on either the group name or one of the member names, then DB2 Analytics Accelerator Loader intercepts it.

The last component is the *DB2 control file*. Only one control file is required per installation of DB2 Analytics Accelerator Loader. However, it must have each member of the DB2 data sharing group defined to it, listing the DB2 DSNZPARM member with the DSNEXIT library and bootstrap data sets (BSDS) for each member of the data sharing group.

This definition can be done through TCz, using the Product Parameters **Update the control file** panel option, as shown in Figure 10-4.

```
TCUSTMZR                 Product Parameters                           10:58:45
 Command ===>                                                    Scroll ===> CSR

 Complete the following tasks to customize the products. The required tasks
 and steps are preselected. Ensure that all parameters are specified for each
 selected step within a task. Press End to save and exit.

 Commands: SAVE - Save parameter values
 Line Commands: / - Select

  Product to Customize
    Product metadata library . : HLO.WRK0110.SHLODENU    > LPAR  . . : RS22
    Product name . . . . . . . : DB2 Analytics Accelera  > Version . : 1.1.0
    Configuration ID: HLO      > Description: DB2 Analytics Accelerator Load  >

 Product customization library .: CSJENN.ALL3.$RS22$.HLO110
                                                                   More:   - +
 / Create the control file, update it, or both

   / Creates a new control file
     Volume serial number for control file  .

   / Update the control file

 / Add Accelerator Loader to the DB2 Admin Launchpad
```

Figure 10-4 Update the control file with DB2 information for each member of the group

10.3 DB2 Analytics Accelerator Loader intercept and started task

An *intercept* is used when loading data to DB2 Analytics Accelerator through the Dual Load function. If you take a close look at DB2 Analytics Accelerator Loader JCL for the Dual Load function, you see JCL that looks much like the standard DB2 LOAD Utility JCL. Specifically, you see an EXEC PGM=DSNUTILB.

DB2 Analytics Accelerator Loader intercepts the standard DB2 LOAD utility job, (DB2 utilities running with the DSNUTILB program), and determines whether the LOAD utility is to be intercepted to load tables to DB2 Analytics Accelerator and optionally DB2 at the same time. A `//HLODUMMY DD DUMMY` must be present in the Dual Load JCL in order for DB2 Analytics Accelerator Loader to properly intercept the DB2 LOAD Utility job.

When this DD is present, DB2 Analytics Accelerator Loader intercepts the DSNUTILB execution and performs its special processing. You can think of DB2 Analytics Accelerator Loader as enhancing and extending the DB2 LOAD Utility functionality. The intercept gets control near the start of the DSNUTILB program execution.

If the `//HLODUMMY DD` is not present, the intercept does not pass control to DB2 Analytics Accelerator Loader and only the DB2 LOAD utility runs. The result is a syntax error from the DB2 LOAD utility on any specific Accelerator Loader keywords. The syntax error is shown in Figure 10-5.

```
DSNU005I  134 15:53:36.16 DSNUGPRS -     LOAD DATA
DSNU005I  134 15:53:36.16 DSNUGPRS -         IDAA_ONLY ON QDS5ACC1
DSNU049I  134 15:53:36.16 DSNUGPRS - INVALID OPERAND 'IDAA_ONLY' FOR KEYWORD 'LOAD'
DSNU005I  134 15:53:36.16 DSNUGPRS -         INDDN ISYSREC
DSNU005I  134 15:53:36.16 DSNUGPRS -         REPLACE
DSNU005I  134 15:53:36.16 DSNUGPRS -         KEEPDICTIONARY
DSNU005I  134 15:53:36.16 DSNUGPRS -         LOG NO
DSNU005I  134 15:53:36.16 DSNUGPRS -         ENFORCE NO
```

Figure 10-5 Results of the intercept not being active for DB2 Analytics Accelerator Loader

The next action performed by the intercept is to connect to the DB2 Analytics Accelerator Loader started task. This started task is set up and configured when the product is customized.

The started task is required for all DB2 Analytics Accelerator Loader functionality, both Dual Load and Consistent Load. If the started task is not running, the intercept stops processing, and the DB2 LOAD Utility itself continues processing. This results in the same syntax error in Figure 10-5.

The started task policy must be configured to monitor the DB2 system that the batch load job is operating against. This should have been done during the customization process of DB2 Analytics Accelerator Loader. A policy contains a record for each DB2 system that has been set up to run with DB2 Analytics Accelerator Loader. Figure 10-6 shows a policy where three different DB2 SSIDs are defined for use with DB2 Analytics Accelerator Loader.

```
EDIT       RSQA.HLO110.QAA5.OPTPLY(QALLPLCH) - 01.02      Columns 00001 00072
Command ===>                                                 Scroll ===> CSR
000030 <!--                                                                -->
000031 <DSNUTILB_INTERCEPT>
000032  <POLICY>
000033
000034   <DB2SYSTEM SSID="BAA1" ACTION="LOAD_ACCELERATOR">
000035   </DB2SYSTEM>
000036   <DB2SYSTEM SSID="QA1A" ACTION="LOAD_ACCELERATOR">
000037   </DB2SYSTEM>
000038   <DB2SYSTEM SSID="QA1Z" ACTION="LOAD_ACCELERATOR">
000039   </DB2SYSTEM>
000046
000047  </POLICY>
000048 </DSNUTILB_INTERCEPT>
****** **************************** Bottom of Data ****************************
```

Figure 10-6 DB2 Analytics Accelerator Loader intercept policy

If the DB2 subsystem of the table being loaded is not in the policy, the intercept does not run and a write to operator (WTO) is seen in the DB2 Analytics Accelerator Loader output stating that the DSNUTILF exit is not active. The following WTO is issued:

```
HLOU5903W/ABPU5903W DSNUTILF EXIT IS INOPERATIVE FOR SSID: QAA5, RSN: 0006
```

In this case, you should use TCz to configure the DB2 subsystem for use with the DB2 Analytics Accelerator Loader.

Every DB2 LOAD job that is intercepted successfully by DB2 Analytics Accelerator Loader is detailed in the started task job log, and recorded in its repository DB2 tables. Figure 10-7 shows an example of the started task job log for a Dual Load run. The complete output is not shown in the interest of space.

```
HLOS0101I 092 16:48:35.69 TCB: 009D1228 Session created. SESS:13D125C8-000006CB-U-LOD5MD2 -JOB04452-0035-RES5 A
HLOS0101I 092 16:49:05.81 TCB: 009D1228 Session created. SESS:13D12510-000006CC-U-BZA1WLMU-STC04077-0063-IBM B
HLOS0101I 092 16:49:05.82 TCB: 009D1228 Session created. SESS:13D139C0-000006CD-U-BZA1WLMU-STC04080-0066-IBM C
HLOS0103I 092 16:49:05.85 TCB: 009D1228 Session terminated. SESS:13D12510-000006CC-U-BZA1WLMU-STC04077-0063-IBM D
HLOS0103I 092 16:49:05.85 TCB: 009D1228 Session terminated. SESS:13D139C0-000006CD-U-BZA1WLMU-STC04080-0066-IBM E
HLOS0101I 092 16:49:07.03 TCB: 009D1228 Session created. SESS:13D12510-000006CE-U-BZA1WLMU-STC04073-005F-IBM
HLOS0101I 092 16:49:07.03 TCB: 009D1228 Session created. SESS:13D139C0-000006CF-U-BZA1WLMU-STC04081-0067-IBM
HLOS0101I 092 16:49:07.03 TCB: 009D1228 Session created. SESS:13D14B40-000006D0-U-BZA1WLMU-STC04085-006B-IBM
HLOS0101I 092 16:49:07.06 TCB: 009D1228 Session created. SESS:13D13908-000006D1-U-BZA1WLMU-STC04082-0068-IBM
HLOS0103I 092 16:49:07.07 TCB: 009D1228 Session terminated. SESS:13D12510-000006CE-U-BZA1WLMU-STC04073-005F-IBM
HLOS0103I 092 16:49:07.07 TCB: 009D1228 Session terminated. SESS:13D14B40-000006D0-U-BZA1WLMU-STC04085-006B-IBM
HLOS0103I 092 16:49:07.08 TCB: 009D1228 Session terminated. SESS:13D139C0-000006CF-U-BZA1WLMU-STC04081-0067-IBM
......
HLOS0103I 092 17:05:54.26 TCB: 009D1228 Session terminated. SESS:13D12510-0000072E-U-BZA1WLMU-STC04081-0067-IBM
HLOS0101I 092 17:05:56.96 TCB: 009D1228 Session created. SESS:13D12510-0000072F-U-BZA1WLMU-STC04081-0067-IBM
HLOS0103I 092 17:05:57.01 TCB: 009D1228 Session terminated. SESS:13D12510-0000072F-U-BZA1WLMU-STC04081-0067-IBM
HLOS0204I 092 17:10:57.01 TCB: 009CDE88 Disconnection from DB2 was successful. SSID=BZA1
HLOS0103I 092 17:18:40.91 TCB: 009D1228 Session terminated. SESS:13D125C8-000006CB-U-LOD5MD2 -JOB04452-0035-RES5 F
```

Figure 10-7 Sample output in the started task job log from a DUAL load run

Legend:

A	Session created for the start of the job (JOB04452).
B	Session created for WLM DSNUTILU (STC04077).
C	Session created for WLM DSNUTILU (STC04080).
D	Session terminated for WLM DSNUTILU (STC04077).
E	Session terminated for WLM DSNUTILU (STC04080).
F	Session terminated for job (JOB04452).

The log records indicate when DSNUTILB or DSNUTILU are intercepted. Error information can sometimes be logged in the started task as well.

The started task is used for these purposes:

- Facilitates communication with DB2 for DB2 Analytics Accelerator Loader
- Controls which DB2 subsystems are configured for the intercept
- Writes log and trace information used for problem determination
- Handles restart processing when multiple DB2 utilities are included in the same SYSIN

The started task and intercept are also used to establish data delivery mechanisms to DB2 Analytics Accelerator and is required for both Dual Load and Consistent Load functions.

Figure 10-8 shows an example of the started task output.

```
HLOS0000I DB2 Analytics Accelerator Loader   Version 0110, FMID=HHL0110, COMPONENT ID=5639-OLA
HLOS0001I Started task initialization is in progress
HLOS0002I Started task initialization is complete
HLOS0000I 134 23:56:39.47 DB2 Analytics Accelerator Loader   Version 0110, FMID=HHL0110, COMPONENT ID=5639-OLA
HLOS0001I 134 23:56:39.47 Started task initialization is in progress
HLOG8008I 134 23:56:39.47 System=RS25    ,Job=HLOQALLS,Job Id=S0244828,Step=HLOQALLS,Program=HLOSINIT,User=HLOSTC  .
HLOG8010I 134 23:56:39.47 CPU=002817-M15-IBM. z/OS     02.01.
HLOS0014I 134 23:56:39.47 SVC installation is complete. SVC number=255
HLOS0019I 134 23:56:39.47 COMX: 2A8BCB68, COMI: 2F408018, SVC EPA: 80C05CA8, MNTLEVEL: PI13308
HLOS0080I 134 23:56:39.47 DB2 Analytics Accelerator Loader   initialization parameters:
HLOS0082I 134 23:56:39.47 HLOID                              = HLO5
HLOS0082I 134 23:56:39.47 HLOOPTS Dataset Name               = RSQA.HLO110.QAA5.OPTPLY
HLOS0082I 134 23:56:39.47 HLOOPTS Member                     = QALLOPTS
HLOS0082I 134 23:56:39.47 HLOPLCY Dataset Name               = RSQA.HLO110.QAA5.OPTPLY
HLOS0082I 134 23:56:39.47 HLOPLCY Member                     = QALLPLCS
HLOS0081I 134 23:56:39.47 Accel Load Tasks                   = 8
HLOS0082I 134 23:56:39.47 Audit Active                       = YES
HLOS0081I 134 23:56:39.47 Audit Rows Maximum Age              = 0
HLOS0082I 134 23:56:39.47 Log Active                          = YES
HLOS0081I 134 23:56:39.47 Log Rows Maximum Age                = 0
HLOS0081I 134 23:56:39.47 SVC Number                          = 255
HLOS0082I 134 23:56:39.47 Trace Active                        = YES
HLOS0081I 134 23:56:39.47 Trace Size                          = 1
HLOS0082I 134 23:56:39.47 Wildcard - Multi-character          = %
HLOS0082I 134 23:56:39.47 Wildcard - Single-character         = _
HLOS0082I 134 23:56:39.47 Workfile DATACLAS Name              = NONE
HLOS0082I 134 23:56:39.47 Workfile MGMTCLAS Name              = NONE
HLOS0082I 134 23:56:39.47 Workfile STORCLAS Name              = NONE
HLOS0082I 134 23:56:39.48 Workfile UNIT Name                  = SYSALLDA
```

Figure 10-8 The started task output

The information is useful in determining the program temporary fix (PTF) level, z/OS level, and other options in effect for the installation.

The installation parameters are all set by TCz options during customization. The following list defines some of the most important install parameters:

AUDIT_ACTIVE Controls whether the auditing table is written to each Accelerator Loader batch job. The audit table is HLOAUDIT.

LOG_ACTIVE Controls whether the log table is written to for historical logging of Accelerator Loader jobs. The log table is HLOLOG.

TRACE_ACTIVE Controls whether tracing information is written to the started task output files.

ACCEL_LOAD_TASKS Controls the default number of parallel tasks for a Dual Load batch job if multiple input files are specified.

In summary, the started task controls which subsystems DB2 Analytics Accelerator Loader can be run against and, after setup, only needs to be examined if there is an issue loading a table.

10.4 Group Consistent Load

Group Consistent Load is a term used to describe the ability of DB2 Analytics Accelerator Loader to load data from one or more operational DB2 tables into DB2 Analytics Accelerator with data consistency to the same point in time. The tables can be related by referential integrity (RI), or simply unrelated tables that need to be loaded to the same point in time. The data can be loaded to a user-specified time for historical analysis, or it can be loaded to the current point in time.

In both cases, this can be done while the operational tables are being updated with Structured Query Language (SQL), thereby eliminating unavailability of operational data. DB2 Analytics Accelerator Loader uses recovery resources and processes to load the data into DB2 Analytics Accelerator.

Unlike a DB2 Analytics Accelerator process that uses the ACCEL_LOAD_TABLES stored procedure, DB2 Analytics Accelerator Loader does not unload the data from the operational tables, and does not therefore lock the operational tables:

1. Users specify the point in time to load.
2. DB2 Analytics Accelerator Loader selects the appropriate image copies before the specified point in time. Images copies can be full, incremental, in-line, or flash copies.
3. DB2 Analytics Accelerator Loader reads the DB2 logs from the image copy point to the specified point in time.
4. The log records are sorted in page and time sequence.
5. Image copies are read and merged.
6. The sorted log records are applied to the relevant page.
7. Table rows are extracted from each page. Rows are decompressed if required.
8. Table rows are written to DB2 Analytics Accelerator using the pipe initiated by the ACCEL_LOAD_TABLES stored procedure.

10.4.1 Not logged tables

Tables that are in a table space that has the NOT LOGGED logging attribute cannot be used for Group Consistent Load. DB2 Analytics Accelerator Loader issues a message when Group Consistent Load is attempted using a table space where the logging attributed is NOT LOGGED:

```
HLO1010E  The following space is not set to LOG for a required log range
```

Message number HLO1014I identifies the database, table space, and partition number.

Group Consistent Load uses DB2 recovery resources (image copies and logs) to load DB2 Analytics Accelerator. Without the log records, it would not be possible to create a consistent point for loading DB2 Analytics Accelerator.

If a table space has the NOT LOGGED logging attribute, and you want to load DB2 Analytics Accelerator using Group Consistent Load, the table space must be altered to change the logging attribute to LOGGED. It is then possible to load DB2 Analytics Accelerator using Group Consistent Load, if an image copy of the table space is taken after the alter statement is run.

10.4.2 Using Group Consistent Load profiles

This section shows how the profile support in the IBM Interactive System Productivity Facility (ISPF) interface can help with the following scenarios:

- Referential integrity-related tables
- Loading to the current point in time
- Loading to a previous point in time

Referential integrity-related tables

To add RI-related tables, follow these steps:

1. On the Consistent Load Options panel, enter the **TABLES** primary command, as shown in Figure 10-9.

```
LOADER                                       Consistent Load Options
Command ===> TABLES

Commands: TABLES - Edit tables list  ACCELERATOR - Select Accelerator

Creator  . . . : TSSXS       Name  . . . . : RI RELATED
Share option . . UPDATE      Description . .                               >

Utility processing options:
  Accelerator name . . . . . . . . . QDS5ACC1
  Load time  . . . . . . . . . . . . CURRENT   (CURRENT, SPECIFIED, or QUIESCE)
  RBA or LRSN end point  . . . . . .                     (hexadecimal value)
  Timestamp end point  . . . . . . .
                                   (YYYY-MM-DD-hh.mm.ss.nnnnnn)
    Time zone of timestamp . . . . . LOCAL (LOCAL or GMT)
  Quiesce end point  . . . . . . . .     (1-999)
  Continue on errors . . . . . . . . NO  (Yes/No)

FlashCopy options:
  Use FlashCopy  . . . . . . . . . . NO  (Yes/No) Only valid for current
  Use FlashCopy DSN template . . . : NO  (Yes/No) Update NO  (Yes/No)

Log read and log apply options:
  SYSCOPY scan operating mode  . . . ZPARM   (LOCAL, RECOVER, ZPARM, or USER)
    SYSCOPY selection preference . . LPLBRPRBFC (LPLBRPRBFC in any order)
  Log reader copy preference . . . . R1R2A1A2   (R1R2A1A2 in any order)
```

Figure 10-9 Consistent Load Options panel

2. On the DB2 Table List panel enter the primary command **ADD** as shown in Figure 10-10.

```
LOADER                  DB2 Table List                     2014/05/12  04:25:22
Command ===> ADD                                              Scroll ===> PAGE

Commands:      ADD - Add table(s)
Line commands: D - Delete table  RIS - Select RI tables
               RIA - Add all RI tables

Creator  . . . : TSSXS       Name  . . . . : RI RELATED
Share option . : UPDATE      Description . :


Cmd Table Name                      Part  Creator       Database Tablespace
--- --------------------------------> ----- ------------> -------- ----------
****************************** Bottom of data ******************************
```

Figure 10-10   DB2 Table List panel is initially empty

3. Enter your table search criteria in the panel, as shown in Figure 10-11. Note that the input is case-sensitive.

```
Table creator like . . BCTEST                              >
Table name like  . . . GLWTEMP                             >

Please note: Table creator and Table name are case sensitive

Press ENTER to process
Press F12 to cancel
```

Figure 10-11   Table search criteria

4. The tables matching your specified criteria are then displayed, as shown in Figure 10-12.

```
LOADER                  Add DB2 Tables                     2014/05/12  04:26:25
Command ===>                                                  Scroll ===> PAGE

Commands:      ALL - Select all tables  DEFAULT - Default sort
Line commands: S - Select a table

Table creator like . . BCTEST*                                > DB2 SSID: QAA5
Table name like  . . . GLWTEMP*                               >
                                                               Row 1 of 5
Cmd Table Name                      Part  Creator       Database Tablespace
--- --------------------------------> ----- ------------> -------- ----------
    GLWTEMP                         ALL   BCTEST        BCTEST   GLWSEMP
    GLWTEMP                         1     BCTEST        BCTEST   GLWSEMP
    GLWTEMP                         2     BCTEST        BCTEST   GLWSEMP
    GLWTEMP                         3     BCTEST        BCTEST   GLWSEMP
    GLWTEMP                         4     BCTEST        BCTEST   GLWSEMP
****************************** Bottom of data ******************************
```

Figure 10-12   Adding tables to the batch job

If you select a table at the ALL part level, currently, for DB2 Analytics Accelerator Loader, this is equivalent to specifying DSNUM 0 (the whole table space), and you require an image copy of the entire table space. DB2 Analytics Accelerator Loader still loads DB2 Analytics Accelerator by partition, but is looking for an image copy of the whole table space to load from. Therefore, you cannot use a FlashCopy image copy, because that type of image copy only works at the partition level.

Generally you should select the partitions that you want to load, because that gives you the greater flexibility of using FlashCopy image copies. However, this might present a challenge if the table has a significant number of partitions.

5. The solution to this is to use the **ALL** primary command which places a “S” at the start of each line, and then clear the ALL part, as shown in Figure 10-13.

```
LOADER                Add DB2 Tables                     2014/05/12  04:26:25
Command ===>                                                   Scroll ===> PAGE

Commands:      ALL - Select all tables  DEFAULT - Default sort
Line commands: S - Select a table

Table creator like . . BCTEST*                                 > DB2 SSID: QAA5
Table name like  . . . GLWTEMP*                                >
                                                                Row 1 of 5
Cmd Table Name                        Part  Creator       Database Tablespace
--- --------------------------------> ----- ------------> -------- ----------
    GLWTEMP                           ALL   BCTEST        BCTEST   GLWSEMP
S   GLWTEMP                           1     BCTEST        BCTEST   GLWSEMP
S   GLWTEMP                           2     BCTEST        BCTEST   GLWSEMP
S   GLWTEMP                           3     BCTEST        BCTEST   GLWSEMP
S   GLWTEMP                           4     BCTEST        BCTEST   GLWSEMP
****************************** Bottom of data ******************************
```

Figure 10-13 Tables can be selected at the partition level

6. After pressing Enter to make your selection, press and PF3 to return to the DB2 Table List panel. Figure 10-14 shows the Table List panel with just the selected partitions displayed.

```
LOADER                DB2 Table List                     2014/05/12  04:28:11
Command ===>                                                 Scroll ===> PAGE

Commands:      ADD - Add table(s)
Line commands: D - Delete table  RIS - Select RI tables
               RIA - Add all RI tables

Creator  . . . : TSSXS       Name  . . . . : RI RELATED
Share option . : UPDATE      Description . :

                                                              Row 1 of 4
Cmd Table Name                      Part  Creator      Database Tablespace
--- ------------------------------> ----- ------------> -------- ----------
    GLWTEMP                         1     BCTEST       BCTEST   GLWSEMP
    GLWTEMP                         2     BCTEST       BCTEST   GLWSEMP
    GLWTEMP                         3     BCTEST       BCTEST   GLWSEMP
    GLWTEMP                         4     BCTEST       BCTEST   GLWSEMP
****************************** Bottom of data ********************************
```

Figure 10-14 DB2 Table List displaying the chosen partitions

7. To add RI-related tables you have two options:
   - Use the **RIS** line command to select from a list of the tables that are related by RI to the table on this panel.
   - Use the **RIA** line command to automatically select all of the tables that are related by RI to the table on this panel.

   Using the **RIS** line command displays the Referentially Dependent Table selection panel shown in Figure 10-15.

```
LOADER                DB2 Table List                     2014/05/12  04:28:11
Command ===>                                                 Scroll ===> PAGE

Commands:      ADD - Add table(s)
Line commands: D - Delete table  RIS - Select RI tables
               RIA - Add all RI tables

Creator  . . . : TSSXS       Name  . . . . : RI RELATED
Share option . : UPDATE      Description . :

                                                              Row 1 of 4
Cmd Table Name                      Part  Creator      Database Tablespace
--- ------------------------------> ----- ------------> -------- ----------
RIS GLWTEMP                         1     BCTEST       BCTEST   GLWSEMP
    GLWTEMP                         2     BCTEST       BCTEST   GLWSEMP
    GLWTEMP                         3     BCTEST       BCTEST   GLWSEMP
    GLWTEMP                         4     BCTEST       BCTEST   GLWSEMP
****************************** Bottom of data ******************************
```

Figure 10-15 Table list using RIS command

Figure 10-16 shows the tables that are referentially dependent. You can select individual tables, or all tables, on this panel.

```
LOADER        Referentially Dependent Tables          2014/05/12  04:31:01
Command ===>                                                Scroll ===> PAGE

Commands:      ALL - Select all tables  DEFAULT - Default sort
Line commands: S - Select/unselect a table
                                                             Row 1 of 4
Cmd Table Name                       Part  Creator       Database Tablespace
--- --------------------------------> ----- ------------> -------- ----------
    GLWTDPT                          N/A   BCTEST        BCTEST   GLWSDPT
    GLWTEPA                          N/A   BCTEST        BCTEST   GLWSEPA
    GLWTPJA                          N/A   BCTEST        BCTEST   GLWSPJA
    GLWTPRJ                          N/A   BCTEST        BCTEST   GLWSPRJ
******************************* Bottom of data *******************************
```

Figure 10-16 List of referentially related tables

8. After pressing ENTER to make your selection, press PF3 to return to the DB2 Table List panel. The selected RI-related tables are added to the list, as shown in Figure 10-17.

   The **RIA** line command automatically adds all RI-related tables to the list without having to select them.

```
LOADER                 DB2 Table List                    2014/05/12  04:33:42
Command ===>                                                Scroll ===> PAGE

Commands:      ADD - Add table(s)
Line commands: D - Delete table  RIS - Select RI tables
               RIA - Add all RI tables

Creator  . . . : TSSXS       Name  . . . . : RI RELATED
Share option . : UPDATE      Description . :

                                                             Row 1 of 8
Cmd Table Name                       Part  Creator       Database Tablespace
--- --------------------------------> ----- ------------> -------- ----------
    GLWTEMP                          1     BCTEST        BCTEST   GLWSEMP
    GLWTEMP                          2     BCTEST        BCTEST   GLWSEMP
    GLWTEMP                          3     BCTEST        BCTEST   GLWSEMP
    GLWTEMP                          4     BCTEST        BCTEST   GLWSEMP
    GLWTDPT                          N/A   BCTEST        BCTEST   GLWSDPT
    GLWTEPA                          N/A   BCTEST        BCTEST   GLWSEPA
    GLWTPJA                          N/A   BCTEST        BCTEST   GLWSPJA
    GLWTPRJ                          N/A   BCTEST        BCTEST   GLWSPRJ
******************************* Bottom of data ********************************
```

Figure 10-17 Table list showing all related tables to the original table

PF3 returns to the Consistent Load Options panel.

Loading to the current point in time

Now that the tables have been specified for Group Consistent Load, it is necessary to choose the *load time*. You have three options to specify the load time:

- CURRENT
- SPECIFIED
- QUIESCE

To load tables to DB2 Analytics Accelerator to the current point in time, specify a load time of CURRENT. DB2 Analytics Accelerator Loader finds the appropriate image copies, and then reads the DB2 log and merges any log records that are relevant to the tables being loaded to DB2 Analytics Accelerator. When the Group Consistent Load process finishes, the relative byte address/log record sequence number (RBA/LRSN) consistency point is visible in the job output. This is specified in relation to the database location time, not the local time.

For example, the following output shows the RBA/LRSN and run time to be different because the database location time has been specified as GMT:

```
Consistent RBA/LRSN = X'00CD17D10A0395821400' (2014-05-02-15.53.20.580952)
RUN DATE 2014/05/02 RUN TIME 11:53:13
```

You can minimize the log reading when you want to load data to DB2 Analytics Accelerator to the current point (with consistency) by instructing DB2 Analytics Accelerator Loader to take a FlashCopy image copy at the start of the Group Consistent Load process. When the FlashCopy image copy is taken, it is taken with consistency, so the FLASHCOPY CONSISTENT parameter is generated into the control cards.

The FLASHCOPY CONSISTENT parameter means that FlashCopy technology is used to copy the object and that any uncommitted work included in the copy is backed out of the copy to make the copy consistent.

The use of FLASHCOPY CONSISTENT requires additional time and system resources during utility processing, because the COPY utility must read the logs and make changes to the image copy.

Figure 10-18 shows the options to control the use of a FlashCopy image copy:

```
FlashCopy options:
  Use FlashCopy  . . . . . . . . . . NO  (Yes/No) Only valid for current
  Use FlashCopy DSN template . . . : NO  (Yes/No) Update NO  (Yes/No)
```

Figure 10-18 FlashCopy options

You must first set Use FlashCopy to Yes and then press Enter, after which you are able to specify the use of a FlashCopy template.

If you specify NO for the Use FlashCopy DSN template field, the default template that is specified in the DSNZPARM is used. Care must be taken to ensure that it includes the symbolic variable for a partition number as part of the template.

If you specify YES for the Use FlashCopy DSN template field, you can specify your own template, and you can ensure that it is correctly defined for the purpose. However, the template is not shared across Accelerator Loader profiles.

Remember that FlashCopy image copies can only be taken at the partition level (if the table space is partitioned).

Figure 10-19 shows an example of the FlashCopy DSN Template panel in which the template name is specified as TEMPL1. The data set name generated by the template is "TSSXS.&DB..&SN..P&PA(3,3)..&UQ." and is made up of a fixed HLQ (TSSXS), database name, space name, the last three numbers of the partition number (using substring), and a unique string of characters.

```
LOADER             Flashcopy DSN Template                 2014/05/12  04:34:50
Command ===>

Commands: SHOW - Show DSN

Template name . . TEMPL1

Enter codes to create a data set name specification:
  Qualifier code . .     Free form literal . .
  Current data set name qualifier string:
  TSSXS.&DB..&SN..P&PA(3,3)..&UQ.

Valid data set name specification codes are
  1. Database         10. Time (HHMMSS)          19. Unique
  2. Space name       11. Hours (HH)             20. SSID
  3. Partition        12. Minutes (MM)           21. User ID
  4. DSNUM            13. Seconds (SS)           22. Job name
  5. Date (YYYYDDD)   14. Local/Recovery (L/R)   23. Step name
  6. Year (YYYY)      15. Primary/Backup (P/B)   24. Utility ID
  7. Month (MM)       16. Copy type (Full/Incr)  25. Utility name
  8. Day (DD)         17. Listdef                26. Use freeform literal
  9. Julian day (DDD) 18. Sequence               27. Substring qualifier
```

Figure 10-19   FlashCopy data set template panel

When you press PF3 you are returned to the Consistent Load Options panel. Other options relate to log read and log apply can be reviewed or changed on this panel, as shown in Figure 10-20.

```
Log read and log apply options:
  SYSCOPY scan operating mode  . . . ZPARM   (LOCAL, RECOVER, ZPARM, or USER)
    SYSCOPY selection preference . . LPLBRPRBFC (LPLBRPRBFC in any order)
  Log reader copy preference . . . . R1R2A1A2   (R1R2A1A2 in any order)
  Buffers in 31 bit storage  . . . . Yes (Yes/No)
  Number of PARALLEL log read  . . . 0   (0-16 tasks)
  Number of PARALLEL log apply . . . 4   (1-10 tasks)
```

Figure 10-20   Log read and log apply options for parallel processing

The following list describes the log read and log apply options:

SYSCOPY scan operating mode

This option relates to the way in which SYSCOPY rows are considered when finding a starting point for processing. The default ZPARM option is sufficient for most scenarios. Using this option, DB2 Analytics Accelerator Loader detects the operating mode that DB2 is running with, and automatically inserts the corresponding control card. The specified value for “SYSCOPY selection preference” is ignored for this option.

Using the LOCAL or RECOVER options instructs DB2 Analytics Accelerator Loader to read the SYSCOPY rows looking for LP/LB (LOCAL) or RP/RB (RECOVER). The specified value for “SYSCOPY selection preference” is ignored for this option. The USER option enables you to specify the “SYSCOPY selection preference.”

SYSCOPY selection preference

The image copy types to attempt to use when scanning SYSCOPY for a starting point.

Log reader copy preference

Enables you to specify the order in which the active and archive logs are searched for relevant records.

The default value R1R2A1A2 scans the archive logs first, and uses archive logs when the same range exists in an archive and active log. This is a safer option, because it avoids an open error in DB2 if an active log is opened for input for DB2 Analytics Accelerator Loader, and DB2 attempts to open the same log for output.

Buffers in 31 bit storage

This value should be set to Yes, because the value No is provided for compatibility reasons and uses 24-bit storage for buffers. Note that Yes is not the default value at the time of this publication.

Number of PARALLEL log read

Relates to the number of parallel log reading tasks. A value of 0 (zero) means that only one task is run per data sharing member. A value of 1 means that there is only one read task running at a time, so for a 2-member data sharing group only one member’s log is processed at a time (no log read parallelism). Generally a value of 0 (the default) should be used so as not to restrict parallelism.

Number of PARALLEL log apply

The number of parallel log apply and load tasks. Specifying a number greater than one (> 1) enables DB2 Analytics Accelerator Loader to distribute the objects across the specified number of tasks to load the partitions in parallel.

Note that only *partitions* of a partitioned table (if greater than one partition) are distributed across the specified number of tasks to run in parallel. Currently, there is a restriction in the ACCEL_LOAD_TABLES stored procedure that prevents more than one table being loaded in parallel, so tables are loaded serially (one at a time).

If you are loading a partitioned table, you should specify a value greater than one to maximize the parallelism, but for non-partitioned tables the value has no effect.

Loading to a previous point in time

To load tables to DB2 Analytics Accelerator to a previous point in time, specify a load time of SPECIFIED or QUIESCE. As with loading tables to the CURRENT point in time, DB2 Analytics Accelerator Loader finds the appropriate image copies, and then reads the DB2 log and merges any logs records that are relevant to the tables being loaded to DB2 Analytics Accelerator to the chosen load time.

Figure 10-21 shows the utility processing options. If you choose the option titled SPECIFIED, a log endpoint or a time stamp endpoint must also be specified. If you choose QUIESCE, a quiesce endpoint must be specified.

```
Utility processing options:
  Accelerator name . . . . . . . . . QDS5ACC1
  Load time  . . . . . . . . . . . . SPECIFIED (CURRENT, SPECIFIED, or QUIESCE)
  RBA or LRSN end point  . . . . . .                     (hexadecimal value)
  Timestamp end point  . . . . . . .
                                    (YYYY-MM-DD-hh.mm.ss.nnnnnn)
    Time zone of timestamp . . . . . LOCAL (LOCAL or GMT)
  Quiesce end point  . . . . . . . .     (1-999)
  Continue on errors . . . . . . . . NO  (Yes/No)
```

Figure 10-21 Utility processing options

For a load time of SPECIFIED, for any end-point type, including time stamps, the RBA chosen by DB2 Analytics Accelerator Loader is determined by rolling the RBA back until there is no in-flight unit of work. For a group of objects, this is effectively a quiesce point.

This process avoids DB2 Analytics Accelerator Loader having to verify the specified log point by attempting a read of that log record in the actual log, and possibly incurring a tape mount, data set allocation, or an extra input/output (I/O).

For a load time of QUIESCE, DB2 Analytics Accelerator Loader takes the RBA from SYSIBM.SYSCOPY. This is known as a *validated load point*. However, if you were to copy the RBA for the quiesce point found in SYSIBM.SYSCOPY and use (paste) it for the SPECIFIED (RBA endpoint), DB2 Analytics Accelerator Loader would not consider that as a validated load point.

You can specify which quiesce you want to use as an endpoint. If you specify “1” as the quiesce endpoint, the most recent one found in SYSIBM.SYSCOPY is taken. If you specify “2” as the quiesce endpoint, the quiesce found before (earlier than) the most recent one is used.

10.5 Dual load

Dual load is a term used to describe the ability of DB2 Analytics Accelerator Loader to load both the DB2 table and DB2 Analytics Accelerator at the same time (IDAA_DUAL). It is also possible to only load DB2 Analytics Accelerator, leaving the DB2 table empty (IDAA_ONLY). Dual Load is sometimes referred to as *External Load*, because the data to be loaded originates from an external source, such as another DB2 subsystem from another LPAR, a non-relational database on z/OS, or another data store from a non-z/OS environment.

It is the responsibility of the user to supply DB2 Analytics Accelerator Loader with a proper source input data set to load into a DB2 Analytics Accelerator accelerated table, or into a DB2 for z/OS table. The input data can come from any source *if* the input data is in a format that can be loaded by the standard DB2 LOAD utility, and it is on z/OS.

The user must also supply field specifications (load control cards) to map the positions and lengths of fields of the input data set records to the columns in the target DB2 or DB2 Analytics Accelerator table. There are many potential sources of data available to load into DB2 Analytics Accelerator, but the following list includes some of the more popular options:

- IMS
- VSAM
- Oracle
- Data from other DB2 subsystems

There are various data extraction methods available for users when pulling data from a non-DB2 source to load into the DB2 Analytics Accelerator. For example, an extract, transform, and load (ETL) tool, such as IBM InfoSphere DataStage® could be used. See Appendix A, “Preparing input data for IBM DB2 Analytics Accelerator Loader” on page 255.

If you are using an existing IBM DB2 LOAD job, care must be taken to examine the LOAD syntax and column specification before adding the required DB2 Analytics Accelerator Loader control statements. There are several restrictions and limitations that are detailed in section 10.6, “Current limitations and considerations” on page 247.

Here is an outline for the dual data load process:

1. DB2 Load JCL needs to be modified to add //HLODUMMY DD DUMMY. This instructs the DB2 Analytics Accelerator Loader started task to intercept the load job.

   One of the two new parameters needs to be added to the Load Utility parameters. They direct the load to the correct Accelerator and determines if it is a load to both DB2 and DB2 Analytics Accelerator or just to DB2 Analytics Accelerator:

   – `IDAA_DUAL on` <*accelerator_name*>
   – `IDAA_ONLY on` <*accelerator_name*>

2. DB2 Analytics Accelerator Loader reads the SYSIN.

   If partition parallelism is wanted, there needs to be one SYSREC/input data set defined per partition being loaded.

3. DB2 Analytics Accelerator Loader loads the data after converting the input data to DB2 internal format using a IBM System z Integrated Information Processor (zIIP) if available. A zIIP engine is not required:

   a. Changes the LOAD Utility statement to include the FORMAT INTERNAL.

   b. DB2 Analytics Accelerator Loader loads either DB2 and DB2 Analytics Accelerator tables in parallel, or only DB2 Analytics Accelerator tables.

      Load REPLACE is always used.

      Load RESUME is not currently supported.

10.5.1 Loading DB2 Analytics Accelerator only

The process of using DB2 Analytics Accelerator Loader to load only the DB2 Analytics Accelerator is called *external data load*. When using DB2 Analytics Accelerator Loader feature to load data only into DB2 Analytics Accelerator, DB2 Analytics Accelerator Loader performs the following steps:

1. Reads the input source data file once.
2. Converts external format data to DB2 internal format data for faster loading. This conversion is zIIP-enabled when zIIP processors are available.
3. Bypasses the load into DB2 for z/OS, and simply loads the DB2 Analytics Accelerator. The DB2 table on z/OS is emptied (Load REPLACE is run with zero rows).

This process can include the following benefits:

- A significant reduction in elapsed time and CPU use.
- Reduced CPU and storage resource usage on z/OS because the table on DB2 is not loaded, resulting in a smaller footprint required for the DB2 table definition.

When using the DB2 Analytics Accelerator Loader external load feature to load non-DB2 or non-z/OS data to the DB2 Analytics Accelerator only, any data in the DB2 for z/OS table is deleted. The data in the table is deleted by running a LOAD REPLACE and loading zero rows. If the data in the DB2 table must be retained for any reason, the Dual Load feature or the Group Consistent Load feature should be used instead.

Take into account the following considerations when loading non-DB2 data into only DB2 Analytics Accelerator using the external load feature of DB2 Analytics Accelerator Loader:

- The data cannot be updated by DB2, and can only be used for query purposes.
- All queries run against the table being loaded must qualify to run on the DB2 Analytics Accelerator. This requires a change to the SQL to ensure that the queries are run against the table on DB2 Analytics Accelerator, because the table in DB2 for z/OS is empty. Additionally, the corresponding tables on DB2 for z/OS are not protected against any INSERTs or LOAD utilities.
- All DB2 security is enforced when using DB2 Analytics Accelerator Loader. That is to say, all data access is still performed through DB2 for z/OS. Therefore, no security is compromised, and no additional security definitions are required.

The DB2 optimizer still drives data access for tables on DB2 Analytics Accelerator. The SET CURRENT QUERY ACCELERATION=ALL statement must be used so that all application's queries are routed to the table on the DB2 Analytics Accelerator to avoid inconsistent results. If SET CURRENT QUERY ACCELERATION=ALL is not specified, the optimizer might send the query to the table on DB2 for z/OS, and not the table on DB2 Analytics Accelerator, causing zero rows or unexpected data to be returned.

10.5.2 Using existing DB2 LOAD Utility JCL

It is common for customers to have existing IBM DB2 LOAD (DSNUTILB) JCL that loads data into a DB2 table. You can use existing JCL to load both DB2 for z/OS table (IDAA_DUAL), and to load a DB2 Analytics Accelerator table (IDAA_ONLY) by making a few modifications to the existing LOAD utility JCL.

Care must be taken to examine the load syntax and column specification before adding the required DB2 Analytics Accelerator Loader control statements. Current limitations and restrictions are listed 10.6, "Current limitations and considerations" on page 247.

The following modifications must be made to the existing DB2 LOAD utility JCL:

1. Add the following DD statement to the batch JCL. This instructs DB2 Analytics Accelerator Loader started task to interrogate the load job:

   ```
   //HLODUMMY DD DUMMY
   ```

2. Add one of the following parameters to the LOAD utility syntax:

   – `IDAA_DUAL on <accelerator_name>`
   – `IDAA_ONLY on <accelerator_name>`

   In this case, *<accelerator_name>* is the name of the DB2 Analytics Accelerator that you want to load.

3. DB2 Analytics Accelerator Loader's load library needs to be added to the LOAD utility steplib concatenation, or to the system LINKLIST.

   Alternatively, module DSNUTILF can be copied to a DB2 data set that is already in the batch job concatenation.

4. The ACCEL_LOAD_TASKS *n* parameter can optionally be added to the LOAD utility syntax to start processing partitions of a partitioned object in parallel. If partition parallelism is wanted, there needs to be one SYSREC/Input data set defined per partition being loaded.

Before execution, a few verifications should be made:

- Verify that the table that is being loaded has been added to the specified accelerator.
- Verify that the DB2 Analytics Accelerator Loader started task or batch job is active.

After verifications are completed and the minor modifications are completed, the load JCL can be submitted for execution.

10.5.3 Using Dual Load profiles

Profiles enable you to reuse a set of options and objects that you want to use to load tables without having to re-enter the options and objects again. Profiles can be shared between users, but they are unique to each subsystem. A profile can be updated by only one person at a time. After a profile has been created, its options can be edited at a later time as needed. This section describes using DB2 Analytics Accelerator Loader ISPF interface to create profiles for the purpose of generating batch utility JCL.

Creating a profile to load into DB2 and DB2 Analytics Accelerator

To create a profile to load into both DB2 and DB2 Analytics Accelerator, follow these steps:

1. On the initial IBM DB2 Analytics Accelerator Loader for z/OS panel, choose option 1 to go to DB2 Analytics Accelerator Loader profiles panel in Figure 10-22.

```
IBM DB2 Analytics Accelerator Loader for z/OS  2014/05/16  09:06:22
Option ===> 1

0  Settings                    User parameters        User ID . . : TSAXC
1  Accelerator Loader profiles  Manage profiles        System ID . : RS25
X  Exit                                               Appl ID . . : HLO
                                                      Version . . : 1.1

                                                      DB2 SSID  . . QAA5
```

Figure 10-22 Accelerator Loader main menu

2. On the Profile Display panel, place the cursor on the `CREATE` command or type CREATE on the command line to create a profile as shown in Figure 10-23.

```
LOADER                  Profile Display                    2014/05/16  09:28:45
 Command ===> CREATE                                            Scroll ===> PAGE

 Commands:      CREATE
 Line commands: B - Build  D - Delete  E - Edit  R - Rename  V - View  C - Copy

 Profile like . . RED BOOK DUAL*                                  DB2 SSID . : QAA5
 Creator like . . TSAXC
 Profile type . . ALL       (DUAL, CONSISTENT, or ALL)
                                                                               >
                                              Share
 Cmd Name                 Creator  Type       Option    Description
 --- -------------------> -------- ---------- --------- --------------------->
```

Figure 10-23 Profile Display panel

3. On the Profile Option panel, a profile name and description of the profile must be added and a profile type of DUAL needs to be specified for a Dual Load profile, as shown in Figure 10-24.

```
 Profile Options

  Enter new Accelerator Loader profile options:
    Creator . . . . : TSAXC
    Profile name  . . RED BOOK DUAL
    Profile type  . . DUAL      (DUAL, CONSISTENT)
    Description . . . RED BOOK DUAL LOAD PROFILE
    Share option  . . UPDATE    (UPDATE, VIEW ONLY, or NO ACCESS)

  Press ENTER to process
  Press F12 to cancel
```

Figure 10-24 Create Profile option panel

4. On the Load from External Options panel, enter the primary command **TABLE** or place the cursor on the option TABLE and hit enter, as shown in Figure 10-25.

```
LOADER           Load from External Options              2014/05/16  09:36:13
 Command ===> TABLE

 Commands: TABLE - Select DB2 table  ACCELERATOR - Select accelerator
           COLINFO - Edit table column info

 Creator  . . . : TSAXC       Name  . . . . : RED BOOK DUAL
 Share option . . UPDATE      Description . . RED BOOK DUAL LOAD PROFILE
>

 Schema . . . . :
 Table name . . :
 Partition  . . :
                                                               More:     +
 Target options:
   Load target . . . . . . . B (A - Accelerator, B - Both accelerator and DB2)
   Accelerator name  . . . . QDS5ACC1
```

Figure 10-25 Load from External Options panel

5. On the DB2 Table Selection panel, the table creator and table name values can be wild carded to return a list of eligible tables. Individual partitions can be selected, all partitions can be individually selected, or PART ALL can be selected to select all partitions of an object.

   In this example, partition 11 through partition 20 were selected, as shown in Figure 10-26.

```
LOADER              DB2 Table Selection                  2014/05/16  09:38:47
 Command ===>                                                 Scroll ===> PAGE

 Commands:      DEFAULT - Default sort
 Line commands: S - Select a table

 Table creator like . . TSAXC*                                > DB2 SSID: QAA5
 Table name like  . . . RANDOM_TABLE*                         >
                                                             Row 10 of 21  -+
 Cmd Table Name                         Part  Creator       Database Tablespace
 --- ---------------------------------> ----- ------------> -------- ----------
     RANDOM_TABLE                       ALL   TSAXC         DBRAND   TSRAND
     RANDOM_TABLE                       10    TSAXC         DBRAND   TSRAND
 S   RANDOM_TABLE                       11    TSAXC         DBRAND   TSRAND
 S   RANDOM_TABLE                       12    TSAXC         DBRAND   TSRAND
 S   RANDOM_TABLE                       13    TSAXC         DBRAND   TSRAND
 S   RANDOM_TABLE                       14    TSAXC         DBRAND   TSRAND
 S   RANDOM_TABLE                       15    TSAXC         DBRAND   TSRAND
 S   RANDOM_TABLE                       16    TSAXC         DBRAND   TSRAND
 S   RANDOM_TABLE                       17    TSAXC         DBRAND   TSRAND
 S   RANDOM_TABLE                       18    TSAXC         DBRAND   TSRAND
 S   RANDOM_TABLE                       19    TSAXC         DBRAND   TSRAND
 S   RANDOM_TABLE                       20    TSAXC         DBRAND   TSRAND
```

Figure 10-26 DB2 Table Selection panel

6. On the Load from External Option panel, select the appropriate options. The Load Target option enables you to load both DB2 and DB2 Analytics Accelerator (B) or just DB2 Analytics Accelerator (A). In this example, select option B to load both DB2 and DB2 Analytics Accelerator.

   An important option under the DB2 Load Options section is called Parallel Load and Load tasks:

   – If Parallel Load is set to Yes, an input default subsystem name (DSN) template is needed, as shown in Figure 10-27, and then it must be named to have an individual SYSREC per partition being loaded.

   – If Parallel Load is set to No, a single SYSREC should be entered in the Input Data Set Name field.

```
LOADER           Load from External Options            2014/05/16  09:44:02
 Command ===>

 Commands: TABLE - Select DB2 table  ACCELERATOR - Select accelerator
           COLINFO - Edit table column info

 Creator  . . . : TSAXC        Name  . . . . : RED BOOK DUAL
 Share option . . UPDATE       Description . . RED BOOK DUAL LOAD PROFILE
>

 Schema . . . . : TSAXC
 Table name . . : RANDOM_TABLE
 Partition  . . : 11:20
                                                                More:     +
 Target options:
   Load target . . . . . . . B (A - Accelerator, B - Both accelerator and DB2)
   Accelerator name  . . . . QDS5ACC1

 Required load options:
   Input data set name  . . . :
   Input member . . . . . . . :          (if data set is partitioned)
   Input DSN template . . . . : &US..IDSD.&DB..&SN..P&PA.    Update NO
(Yes/No)
   Table column info DSN  . . . 'TSAXC.PUNCH.DBRAND.TSRAND.P00001'
   Table column info member . .          (if data set is partitioned)
```

Figure 10-27 Load from External Options panel displaying a SYSREC template

Figure 10-28 shows Parallel load set to Yes.

```
LOADER          Load from External Options              2014/05/16  09:45:14
 Command ===>

 Commands: TABLE - Select DB2 table  ACCELERATOR - Select accelerator
           COLINFO - Edit table column info

 Creator  . . . : TSAXC       Name  . . . . : RED BOOK DUAL
 Share option . . UPDATE      Description . . RED BOOK DUAL LOAD PROFILE
>

 Schema . . . . : TSAXC
 Table name . . : RANDOM_TABLE
 Partition  . . : 11:20
                                                                More:   - +
 DB2 load options:
   Parallel load  . . YES (Yes/No)
   Load tasks . . . . 4   (1-20)
   Utility ID . . . . HMIAL
   KEEPDICTIONARY . . YES (Yes/No)
   ENFORCE  . . . . . YES (Yes/No)
   LOG  . . . . . . . YES           (YES, NO, or NOCOPYPEND)
   NUMRECS  . . . . .               (Integer or blank)
   SORTDEVT . . . . . SYSALLDA      (Device type or blank)
   SORTNUM  . . . . . 4             (2-255 or blank)
```

Figure 10-28 Load from External Options panel

7. The tables and partitions have been selected for loading, and on the Profile Display panel the JCL can be built by placing a B on the command line next to the profile name, as shown in Figure 10-29.

```
 LOADER                 Profile Display                  2014/05/16  09:48:19
 Command ===>                                                 Scroll ===> PAGE

 Commands:      CREATE
 Line commands: B - Build  D - Delete  E - Edit  R - Rename  V - View  C - Copy

 Profile like . . RED BOOK DUAL L*                          DB2 SSID . : QAA5
 Creator like . . TSAXC
 Profile type . . ALL       (DUAL, CONSISTENT, or ALL)
                                                             Row 1 of 1    >
                                             Share
 Cmd Name                  Creator  Type       Option    Description
 --- ------------------->  -------- ---------- --------- -------------------->
 B   RED BOOK DUAL LOAD P TSAXC    Dual       UPDATE    RED BOOK DUAL LOAD PRO
```

Figure 10-29 Profile Display panel

In the generated JCL shown in Figure 10-30 and Figure 10-31, the required DD statement //HLODUMMY DD is placed in the JCL, in addition to DB2 Analytics Accelerator Loader parameters.

```
//REDBOOKD JOB ,'TSAXC',CLASS=A,MSGCLASS=X,NOTIFY=TSAXC,
//              REGION=256M
//JOBLIB   DD DISP=SHR,DSN=QA1A.SDSNEXIT
//         DD DISP=SHR,DSN=HLO.QA0110.SHLOLOAD
//         DD DISP=SHR,DSN=DSN.VA10.SDSNLOAD
//*
//HLOD0100 EXEC PGM=DSNUTILB,
//         REGION=0000M,
//         PARM=('QAA5','HMIAL')
//STEPLIB  DD DISP=SHR,DSN=RSQA.HLO110.IBMTAPE.SHLOLOAD
//         DD DISP=SHR,DSN=QDS5.SDSNEXIT
//         DD DISP=SHR,DSN=DSN.VA10.SDSNLOAD
//HLODUMMY DD DUMMY
//SYSPRINT DD SYSOUT=*
//UTPRINT  DD SYSOUT=*
```

Figure 10-30 Accelerator Loader external load JCL

```
 LOAD DATA
       IDAA_DUAL ON QDS5ACC1
       ACCEL_LOAD_TASKS 4
       KEEPDICTIONARY
       LOG NO
       ENFORCE NO
       WORKDDN(ISYSUT1,ISORTOUT)
       INTO TABLE
           "TSAXC"."RANDOM_TABLE"
           PART 11
           REPLACE
           INDDN DEMODUAL
           (
               PART_ID      POSITION(01)   INTEGER EXTERNAL (8),
               PART_KEY     POSITION(01)   INTEGER EXTERNAL (8),
           )
       INTO TABLE
           "TSAXC"."RANDOM_TABLE"
           PART 12
           REPLACE
           INDDN DEMODUAL
           (
               PART_ID      POSITION(01)   INTEGER EXTERNAL (8),
               PART_KEY     POSITION(01)   INTEGER EXTERNAL (8),
           )
```

Figure 10-31 Accelerator Loader external load SYSIN

Creating a profile to load only DB2 Analytics Accelerator

To create a profile to load only Db2 Analytics Accelerator, follow these steps:

1. On the initial IBM DB2 Analytics Accelerator Loader for z/OS panel, choose option 1 to go to DB2 Analytics Accelerator Loader profiles panel shown in Figure 10-32.

```
IBM DB2 Analytics Accelerator Loader for z/OS  2014/05/16  09:06:22
Option ===> 1

0  Settings                     User parameters        User ID . . : TSAXC
1  Accelerator Loader profiles  Manage profiles        System ID . : RS25
X  Exit                                                Appl ID . . : HLO
                                                       Version . . : 1.1

                                                       DB2 SSID  . . QAA5
```

Figure 10-32 Accelerator Loader main menu

2. On the Profile Display panel place the cursor on the **CREATE** command or type CREATE on the command line to create a profile, as shown in Figure 10-33.

```
LOADER                 Profile Display                    2014/05/16  09:28:45
 Command ===> CREATE                                             Scroll ===> PAGE

 Commands:      CREATE
 Line commands: B - Build  D - Delete  E - Edit  R - Rename  V - View  C - Copy

 Profile like . . RED BOOK DUAL*                               DB2 SSID . : QAA5
 Creator like . . TSAXC
 Profile type . . ALL       (DUAL, CONSISTENT, or ALL)
                                                                               >
                                             Share
 Cmd Name                 Creator  Type      Option    Description
 --- ------------------> -------- ---------- --------- --------------------->
```

Figure 10-33 Profile Display panel

3. On the Profile Option panel, a profile name and description of the profile must be added, and a profile type of DUAL needs to be added for a DUAL Load Profile. See Figure 10-34.

```
 Profile Options

  Enter new Accelerator Loader profile options:
    Creator . . . . : TSAXC
    Profile name  . . RED BOOK DUAL ACCELERATOR
    Profile type  . . DUAL       (DUAL, CONSISTENT)
    Description . . . RED BOOK DUAL LOAD PROFILE
    Share option  . . UPDATE    (UPDATE, VIEW ONLY, or NO ACCESS)

  Press ENTER to process
  Press F12 to cancel
```

Figure 10-34 Create Profile Options panel

4. On the Load from External Options panel, enter the primary **TABLE** command, or place the cursor on TABLE and hit enter, as shown in Figure 10-35.

```
LOADER           Load from External Options              2014/05/16  09:36:13
 Command ===> TABLE

 Commands: TABLE - Select DB2 table  ACCELERATOR - Select accelerator
           COLINFO - Edit table column info

 Creator  . . . : TSAXC       Name  . . . . : RED BOOK DUAL
 Share option . . UPDATE      Description . . RED BOOK DUAL LOAD PROFILE
>

 Schema . . . . :
 Table name . . :
 Partition  . . :
                                                                More:     +
 Target options:
   Load target . . . . . . . A (A - Accelerator, B - Both accelerator and DB2)
   Accelerator name  . . . . QDS5ACC1
```

Figure 10-35 Load from External Options panel

5. On the DB2 Table Selection panel, you can use wild cards in table creator and table name to return a smaller list of eligible tables. Individual partitions can be selected, all partitions can be individually selected, or PART ALL can be selected to select all partitions of an object.

   In this example, partition 11 through partition 20 were selected, as shown in Figure 10-36.

```
LOADER               DB2 Table Selection                 2014/05/16  09:38:47
 Command ===>                                                 Scroll ===> PAGE

 Commands:      DEFAULT - Default sort
 Line commands: S - Select a table

 Table creator like . . TSAXC*                                > DB2 SSID: QAA5
 Table name like  . . . RANDOM_TABLE*                         >
                                                             Row 10 of 21  -+
 Cmd Table Name                        Part  Creator       Database Tablespace
 --- --------------------------------> ----- ------------> -------- ----------
     RANDOM_TABLE                      ALL   TSAXC         DBRAND   TSRAND
     RANDOM_TABLE                      10    TSAXC         DBRAND   TSRAND
 S   RANDOM_TABLE                      11    TSAXC         DBRAND   TSRAND
 S   RANDOM_TABLE                      12    TSAXC         DBRAND   TSRAND
 S   RANDOM_TABLE                      13    TSAXC         DBRAND   TSRAND
 S   RANDOM_TABLE                      14    TSAXC         DBRAND   TSRAND
 S   RANDOM_TABLE                      15    TSAXC         DBRAND   TSRAND
 S   RANDOM_TABLE                      16    TSAXC         DBRAND   TSRAND
 S   RANDOM_TABLE                      17    TSAXC         DBRAND   TSRAND
 S   RANDOM_TABLE                      18    TSAXC         DBRAND   TSRAND
 S   RANDOM_TABLE                      19    TSAXC         DBRAND   TSRAND
 S   RANDOM_TABLE                      20    TSAXC         DBRAND   TSRAND
```

Figure 10-36 DB2 Table Selection panel

6. On the Load from External Option panel, select the appropriate options. The Load Target option is the option to load both DB2 and DB2 Analytics Accelerator (B) or just DB2 Analytics Accelerator (A). In this example, select A to load just DB2 Analytics Accelerator.

   Important options are under the DB2 Load Options section, called Parallel load and Load tasks:

   – If Parallel load is set to Yes, an input DSN template needs to be named to have an individual SYSREC per partition being loaded, as shown in Figure 10-37 and Figure 10-38 on page 245.

   – If Parallel load is set to No, a single SYSREC should be entered in the Input Data Set Name field.

```
LOADER           Load from External Options             2014/05/16  09:44:02
 Command ===>

 Commands: TABLE - Select DB2 table  ACCELERATOR - Select accelerator
           COLINFO - Edit table column info

 Creator  . . . : TSAXC       Name  . . . . : RED BOOK DUAL
 Share option . . UPDATE      Description . . RED BOOK DUAL LOAD PROFILE
>

 Schema . . . . : TSAXC
 Table name . . : RANDOM_TABLE
 Partition  . . : 11:20
                                                               More:     +
 Target options:
   Load target . . . . . . . A (A - Accelerator, B - Both accelerator and DB2)
   Accelerator name  . . . . QDS5ACC1

 Required load options:
   Input data set name  . . . :
   Input member . . . . . . . :          (if data set is partitioned)
   Input DSN template . . . . : &US..IDSD.&DB..&SN..P&PA.    Update NO
(Yes/No)
   Table column info DSN  . . . 'TSAXC.PUNCH.DBRAND.TSRAND.P00001'
   Table column info member . .          (if data set is partitioned)
```

Figure 10-37 Load from External Options panel displaying a SYSREC template

```
LOADER           Load from External Options             2014/05/16  09:45:14
 Command ===>

 Commands: TABLE - Select DB2 table  ACCELERATOR - Select accelerator
           COLINFO - Edit table column info

 Creator  . . . : TSAXC        Name  . . . . : RED BOOK DUAL
 Share option . . UPDATE       Description . . RED BOOK DUAL LOAD PROFILE
>

 Schema . . . . : TSAXC
 Table name . . : RANDOM_TABLE
 Partition  . . : 11:20
                                                                More:   - +
 DB2 load options:
   Parallel load  . . YES (Yes/No)
   Load tasks . . . . 4   (1-20)
   Utility ID . . . . HMIAL
   KEEPDICTIONARY . . YES (Yes/No)
   ENFORCE  . . . . . YES (Yes/No)
   LOG  . . . . . . . YES          (YES, NO, or NOCOPYPEND)
   NUMRECS  . . . . .              (Integer or blank)
   SORTDEVT . . . . . SYSALLDA     (Device type or blank)
   SORTNUM  . . . . . 4            (2-255 or blank)
```

Figure 10-38 Accelerator Loader Load from External Options panel

7. The tables and partitions have been selected for loading. On the Profile Display panel, the JCL can be built by placing a B on the command line next to the profile name, as shown in Figure 10-39.

```
 LOADER                 Profile Display                  2014/05/16  09:48:19
 Command ===>                                                Scroll ===> PAGE

 Commands:      CREATE
 Line commands: B - Build  D - Delete  E - Edit  R - Rename  V - View  C - Copy

 Profile like . . RED BOOK DUAL L*                             DB2 SSID . : QAA5
 Creator like . . TSAXC
 Profile type . . ALL       (DUAL, CONSISTENT, or ALL)
                                                               Row 1 of 1    >
                                              Share
 Cmd Name                   Creator  Type       Option    Description
 --- -------------------> -------- ---------- --------- --------------------->
 B   RED BOOK DUAL LOAD P TSAXC    Dual       UPDATE    RED BOOK DUAL LOAD PRO
```

Figure 10-39 Profile Display panel

8. In the generated JCL shown in Figure 10-40 and Figure 10-41, the required DD statement //HLODUMMY DD is placed in the JCL, in addition to DB2 Analytics Accelerator Loader parameters.

```
//REDBOOKD JOB ,'TSAXC',CLASS=A,MSGCLASS=X,NOTIFY=TSAXC,
//             REGION=256M
//JOBLIB   DD DISP=SHR,DSN=QA1A.SDSNEXIT
//         DD DISP=SHR,DSN=HLO.QA0110.SHLOLOAD
//         DD DISP=SHR,DSN=DSN.VA10.SDSNLOAD
//*
//HLOD0100 EXEC PGM=DSNUTILB,
//         REGION=0000M,
//         PARM=('QAA5','HMIAL')
//STEPLIB  DD DISP=SHR,DSN=RSQA.HLO110.IBMTAPE.SHLOLOAD
//         DD DISP=SHR,DSN=QDS5.SDSNEXIT
//         DD DISP=SHR,DSN=DSN.VA10.SDSNLOAD
//HLODUMMY DD DUMMY
//SYSPRINT DD SYSOUT=*
//UTPRINT  DD SYSOUT=*
```

Figure 10-40 Accelerator only load JCL sample

```
 LOAD DATA
       IDAA_ONLY ON QDS5ACC1
       ACCEL_LOAD_TASKS 4
       KEEPDICTIONARY
       LOG NO
       ENFORCE NO
       WORKDDN(ISYSUT1,ISORTOUT)
       INTO TABLE
           "TSAXC"."RANDOM_TABLE"
           PART 11
           REPLACE
           INDDN DEMODUAL
           (
               PART_ID      POSITION(01)   INTEGER EXTERNAL (8),
               PART_KEY     POSITION(01)   INTEGER EXTERNAL (8),
           )
       INTO TABLE
           "TSAXC"."RANDOM_TABLE"
           PART 12
           REPLACE
           INDDN DEMODUAL
           (
               PART_ID      POSITION(01)   INTEGER EXTERNAL (8),
               PART_KEY     POSITION(01)   INTEGER EXTERNAL (8),
           )
```

Figure 10-41 Accelerator Loader Accelerator only load JCL sample SYSIN

This provides an efficient mechanism to extend existing batch jobs to load data into DB2 Analytics Accelerator too. Rather than adding a step to the JCL, the IDAA_DUAL option and the HLODUMMY DD statement provide a convenient way to change job skeletons only.

10.6 Current limitations and considerations

Being a new product, there are some current restrictions and limitations to DB2 Analytics Accelerator Loader. This section details some of those restrictions, and spells out some potential workarounds until support for some restricted functionality is added.

10.6.1 Change data capture

DB2 Analytics Accelerator Loader doesn't currently support loading tables that are in replication mode. Tables in replication mode are being replicated from DB2 into DB2 Analytics Accelerator. This is true for both the Consistent Load and Dual Load functions. If you want to use DB2 Analytics Accelerator Loader for a table in replication mode, you first need to remove it from replication mode.

10.6.2 Dual load-specific limitations

In general, existing DB2 LOAD Utility JCL can be easily converted to run a Dual Load and provide the capability to load both DB2 and DB2 Analytics Accelerator in parallel, or load DB2 Analytics Accelerator only. Therefore, consider that you currently have DB2 LOAD Utility JCL that is being used to populate a DB2 data warehouse or an operational data store on DB2.

The LOAD Utility JCL can be converted to run DB2 Analytics Accelerator Loader fairly easily by simply adding the //HLODUMMY DD statement. Also add a DB2 Analytics Accelerator Loader control card to direct it to load DB2 Analytics Accelerator in parallel with DB2. There are some restrictions that should be considered before converting existing DB2 LOAD Utility JCL to Analytics Accelerator Loader JCL. See "Input data format (SYSREC)" and "Other DB2 LOAD Utility syntax restrictions" on page 248.

Input data format (SYSREC)

The only type of input currently supported by DB2 Analytics Accelerator Loader is FORMAT UNLOAD data. This is sometimes referred to as *FORMAT EXTERNAL*. Specifically, DB2 Analytics Accelerator Loader does not accept data in the following formats:

FORMAT INTERNAL The data must be in an external format. The newer UNLOAD FORMAT INTERNAL is not supported in DB2 Analytics Accelerator Loader at this time. This doesn't necessarily mean that numbers (integers, decimal, and so on) need to be represented as a character, but a true FORMAT INTERNAL unload file cannot be used in DB2 Analytics Accelerator Loader today.

If FORMAT INTERNAL is being used, a good path forward would be to switch to a FORMAT EXTERNAL unload, and have the Unload utility generate a field specification. The field specification can be used to build JCL for DB2 Analytics Accelerator Loader.

FORMAT DELIMITED This format can sometimes be used when exporting data from non-System z platform databases.

FORMAT SQL/DS The DB2 LOAD Utility can be passed a SQL statement, and load the DB2 table from the results of the SQL statement.

DB2 Analytics Accelerator Loader only supports Extended Binary Coded Decimal Interchange Code (EBCDIC) input to the load process, so the input data SYSREC file must be in EBCDIC encoding. The table must also be in EBCIDC encoding as well. The following syntax is therefore not supported:

ASCII This specifies the input file is in American Standard Code for Information Interchange (ASCII) format. If the target table is in EBCDIC format, the input file must be converted to EBCDIC format before running DB2 Analytics Accelerator Loader.

UNICODE This specifies the input file is in Unicode format. If the target table is in EBCDIC format, the input file must be converted to EBCDIC format before running DB2 Analytics Accelerator Loader.

CCSID This is used to specify the specific encoding format of the input file to the LOAD Utility. It might be possible to convert the data to EBCDIC before running DB2 Analytics Accelerator Loader.

Other keywords that control the specific format of the input file have yet to be added to DB2 Analytics Accelerator Loader:

CONTINUEIF This keyword signifies that the input file can wrap to more than one line. In many cases, the tool being used to build the input SYSREC file has options to not wrap the records to more than one line. The LRECL output file can be made bigger in many cases to make this change.

STRIP This keyword directs the DB2 LOAD Utility to remove leading zeros, or leading or trailing blanks from input fields.

TRUNCATE When the input file data is characters and the input field is too long for the target column, this tells the DB2 LOAD Utility to truncate the input field from the right so that it fits in the target column.

Other DB2 LOAD Utility syntax restrictions

The following keyword functions are not available in DB2 Analytics Accelerator Loader. These don’t necessarily deal with the input data, but are restrictions for other reasons:

RESUME This is more of a DB2 Analytics Accelerator restriction. Data can only be loaded and replaced at the partition level. If there is a great amount of data, it might be possible to create a partition by range key, and only replace the data for the last partition.

SHRLEVEL CHANGE This is an Accelerator Loader restriction. If you require data to be loaded with share level change to DB2, you might consider using the Consistent Load capability of DB2 Analytics Accelerator Loader. This enables you to load the data to DB2 with SHRLEVEL CHANGE, and then load it to DB2 Analytics Accelerator without locking the DB2 table in any way.

10.7 First steps to troubleshooting

The purpose of this section is to provide you with some help to diagnose and correct (or clean up) failed DB2 Analytics Accelerator Loader jobs, for either Dual Load or Group Consistent Load.

10.7.1 Dual load or external load failures

This section describes the information available that can help you clean up a failed DB2 Analytics Accelerator Loader job.

DB2 Analytics Accelerator Loader has several tables that are used for auditing, logging and job tracking purposes. The HLOUSTAT table space is used by DB2 Analytics Accelerator Loader to track utility jobs used for Dual or External loads. To track utility progress, rows are inserted into the table when the utility begins, updated during execution, and deleted when the utility job finishes. For DB2 Analytics Accelerator Loader jobs that have completed successfully, no rows are left in the table.

Step 1: Locating the utility job status table

To locate this table, you have to look at the DB2 Analytics Accelerator Loader options module to determine the DB2 subsystem where the table is.

The option member contains the initialization options for the DB2 Analytics Accelerator Loader started task. The contents of the options module are emitted in the appropriate DB2 Analytics Accelerator Loader started task output.

The DB2 subsystem defined in the DB2_SSID variable is the "primary DB2 subsystem ID" where logging and auditing takes place.

Note that one DB2 Analytics Accelerator Loader started task can monitor multiple DB2 subsystems. In some cases, it might have been decided to run multiple DB2 Analytics Accelerator Loader started tasks to handle the workload more efficiently.

In Figure 10-42, the primary DB2 subsystem ID (DB2_SYSTEM) has been identified as QA1A. Also note the DB2 Analytics Accelerator Loader ID (HLOID). You need these two pieces of information later to clean up any failed external load jobs.

```
<OPTIONS
 HLOID="HLO5"
 AUDIT_ACTIVE="YES"
 AUDIT_MAX_AGE="0"
 DB2_CONNECT_TO_ALL_SUBSYSTEMS="YES"
 DB2_CONNECTION_IDLE_TIMEOUT="300"
 DB2_PLAN_NAME="HLO11SMP"
 DB2_SSID="QA1A"
```

Figure 10-42 Example options

Step 2: Cleaning up an external load job

If an external load job does not complete successfully, you might see messages indicating that DB2 Analytics Accelerator Loader failed with a syntax error, or something similar. If you are running a Dual Load batch job, where both DB2 and DB2 Analytics Accelerator are being loaded, normally this means that the LOAD Utility has stopped in DB2. Also, there is an entry in the HLOUSTAT status table.

To clean up the status in DB2 you might just run a **`-TERM UTILITY`** DB2 command, but this still leaves an entry in the HLOUSTAT status table. If you run the same DB2 Analytics Accelerator Loader job again, it fails with message HLOU5312E, which indicates that there is already a DB2 Analytics Accelerator Loader job running with the same utility ID. Instead, you should run the HLOMAINT batch job to clean up the DB2 Analytics Accelerator Loader repository table. A sample job can be found in the hlq.SHLOSAMP data set, member HLOMAINT.

The HLOMAINT program has several input parameters that uniquely identify the load utility job to clean up. These input parameters should be input in the order described:

- DB2 Analytics Accelerator Loader intercept ID
- The utility function to be performed:
  - TERM_UTILITY
  - MARK_COMPLETE
  - STEP_ADVANCE
  - FORCE_RESTART
- The DB2 subsystem ID of the subsystem on which the utility is stopped
- The utility ID to be stopped

From the information in “Step 1: Locating the utility job status table” on page 249, you can determine the DB2 Analytics Accelerator Loader intercept ID. In addition to that information, you need to identify the utility function that you want to run (in this example, TERM UTILITY), the DB2 subsystem where the utility was run, and the utility ID.

You can determine that information from the utility job itself, or from the HLOUSTAT table. The DSNUTILB program PARM input value contains the DB2 subsystem ID (or the DSNUPROC statement). Figure 10-43 shows an example.

```
//HLOD0100 EXEC PGM=DSNUTILB,
//         REGION=0000M,
//         PARM=('QAA5','EMP')
```

Figure 10-43 Parameters for starting DSNUTILB

If the job has a utility ID, that is also shown on the PARM input value (EMP) as shown in Figure 10-43. If the job was not given a utility ID, it is composed of the USERID assigned to DB2 Analytics Accelerator Loader job and the JOBNAME in the format <*userid*>.<*jobname*>.

You can also query the HLOUSTAT table if you know some of the information, or you want to check that you are using the correct input values for the HLOMAINT program.

Figure 10-44 shows a query and output that is used to retrieve information from the HLOUSTAT table that ran with today's date (CURRENT DATE).

```
SELECT HLOID,DB2SSID,UTILID,JOBNAME,USERID,INSERTED
 FROM  "HLO"."HLOUSTAT"
 WHERE DATE(INSERTED) = CURRENT DATE


HLOID DB2SSID UTILID           JOBNAME  USERID   INSERTED
----- ------- ---------------- -------- -------- --------------------------
HLO5  QA1A    LOAD.HLO         HLLD063  CSTROS   2014-05-12-05.21.09.704618
HLO5  QA1A    HLO0592.LOADIDAA HLO0592  CSTROS   2014-05-12-06.00.30.061772
HLO5  QA1A    HLLD063.LOADIDAA HLLD063  CSTROS   2014-05-12-05.30.19.960179
```

Figure 10-44 Sample query and output

To clean up the first Accelerator Loader job, use the HLOMAINT program, as shown in Figure 10-45.

```
//MAINT EXEC PGM=HLOMAINT,
//     PARM='HLO5,TERM_UTILITY,QA1A,LOAD.HLO'
```

Figure 10-45 Parameters for HLOMAINT program

The TERM_UTILITY function issues the **-TERM UTILITY** DB2 command for the specified utility ID, and deletes the associated entry from DB2 Analytics Accelerator Loader status table.

One of the primary uses of the status table is to manage restart processing. For example, a single job step might have 20 LOAD statements, and they all run successfully except for number 19. It failed because a work file ran out of space. This puts the DB2 utility in a STOPPED state, and it can be restarted after the user fixes the work file allocation.

DB2 Analytics Accelerator Loader records the success of the first 18 LOAD utilities in the HLOUSTAT status table, and it uses that data to restart the correct LOAD statement (#19 in this case) when the job is rerun (restarted). If you re-ran your job in this scenario, and you had already run HLOMAINT to clean up the HLOUSTAT table, DB2 Analytics Accelerator Loader would have no prior knowledge of your batch job.

It would start over at the first LOAD utility statement, rather than restarting at the LOAD utility statement that failed. Only run HLOMAINT when you are sure that you want to discard any work that ran successfully before the failure.

10.7.2 Clean up after Group Consistent Load failures

This section describes what to do if a Group Consistent Load job fails.

Unlike failures for dual (external) load jobs, there is no utility status table that can be used to help clean up after a failure. It might not be obvious at first that a clean-up is required. The HLO#PIPE batch job is supplied in hlq.SHLOSAMP to run a program to clean up common storage, to avoid errors with the data pipe.

Care must be taken to avoid running this job if another DB2 Analytics Accelerator Loader job is processing on the same DB2 subsystem as the one named in this job. The program in this batch job, HLOCOMCL, has only one input value, which is the DB2 subsystem (DB2SSID).

10.7.3 Stored procedures

There are two stored procedures that might be affected by a failed load job:

DSNUTILU Used for loading the DB2 tables.

ACCEL_LOAD_TABLES Used for loading DB2 Analytics Accelerator tables.

You can check on the status of these stored procedures (and others) by using the **`-DISPLAY PROCEDURE(*)`** DB2 command, as shown in Figure 10-46. DB2 Analytics Accelerator-provided stored procedures, and DB2-provided stored procedures used by DB2 Analytics Accelerator, currently all have the SCHEMA=SYSPROC.

```
 -DISPLAY PROCEDURE(*)


******************************** Top of Data **********************************
DSNX940I  !QA1A DSNX9DIS DISPLAY PROCEDURE REPORT FOLLOWS -

------- SCHEMA=SYSPROC
PROCEDURE      STATUS ACTIVE QUED MAXQ TIMEOUT FAIL WLM_ENV
DSNUTILU
               STARTED     4    0    4      11    0 QA1ATCB1

ACCEL_ADD_TABLES
               STARTED     0    0    1       0    0 QA1AACC1
```

Figure 10-46 DB2 DISPLAY PROCEDURE command and output

The STATUS column should be examined. In Figure 10-47, the status is shown as STARTED for both stored procedures. However, if the stored procedure has stopped for some reason, the status might be STOPABN. The stored procedure was stopped because of a previous abnormal termination of the stored procedure application. SQL CALL requests for the stored procedure are rejected.

It might be possible to start the stopped stored procedure by issuing the command:

```
-START PROCEDURE(stored-procedure-name)
```

However, before doing so, you should check the status of the WLM application environment where the stored procedure is run, because it might need to be restarted.

In Figure 10-46, the value for the WLM_ENV column for the stopped stored procedure should be used to determine the status of the WM application environment.

Using that value, you should issue a WLM display command from the z/OS console, or the System Display and Search Facility (SDSF), as shown in Figure 10-47. Note that we have chosen the QA1ATCB1 application environment name as an example.

```
/D WLM,APPLENV=QA1ATCB1


IWM029I  08.23.11  WLM DISPLAY 653
  APPLICATION ENVIRONMENT NAME     STATE     STATE DATA
  QA1ATCB1                         AVAILABLE
  ATTRIBUTES: PROC=QA1ATCB1 SUBSYSTEM TYPE: DB2
```

Figure 10-47 WLM DISPLAY command and output

In Figure 10-47 on page 252, the STATE of the WLM application environment (QA1ATCB1) is shown as AVAILABLE, indicating that there are no issues. However, the STATE might be STOPPED, in which case you need to restart the WLM application environment before restarting the stored procedure.

To restart the WLM application environment, it is best to perform a QUIESCE, and then a resume. Figure 10-48 shows the WLM commands to QUIESCE and RESUME the WLM application environment. You can issue the command from the z/OS (or SDSF) console.

```
/V WLM,APPLENV=QA1ATCB1,QUIESCE

/V WLMAPPLENV=QA1ATCB1,RESUME
```

Figure 10-48 WLM commands to quiesce and resume

A

Preparing input data for IBM DB2 Analytics Accelerator Loader

This Appendix describes how to load data into IBM DB2 Analytics Accelerator from external sources. Those external sources can be non-DB2 data from virtually any source, or they can also be DB2 data from IBM DB2 for Linux, UNIX, and Windows, or a different IBM DB2 for z/OS subsystem.

This external data can then be loaded into both DB2 for z/OS and DB2 Analytics Accelerator, or just DB2 Analytics Accelerator. Before the external data can be loaded, the data must first be converted into a specific Load format that is accepted by DB2 Analytics Accelerator Loader.

In this appendix, we describe two examples:

- First, we show how to convert IBM IMS data to the appropriate format, and load it into DB2 Analytics Accelerator.
- Second, we show how to convert Oracle data and load it into DB2 Analytics Accelerator.

For those interested in understanding how to load Virtual Storage Access Method (VSAM) data into DB2 Analytics Accelerator, it is almost identical to the process for IMS data, but slightly simpler. The section covering IMS points out the steps that are slightly different for VSAM.

This appendix contains the following sections:

- Introduction
- Loading non-DB2 data, such as IMS and VSAM data, to DB2 Analytics Accelerator
- Loading Oracle data to DB2 Analytics Accelerator

Introduction

The IBM DB2 Analytics Accelerator Loader for z/OS external data load feature provides the capability to load external data, which is data in a data set, into a DB2 Analytics Accelerator table. The data in this data set can come from many sources, including data from other DB2 systems. There are two options available when loading data into DB2 Analytics Accelerator using the external load function:

- DB2 Analytics Accelerator Loader has the capability to load data into a DB2 for z/OS and DB2 Analytics Accelerator table simultaneously (Dual Load option).
- DB2 Analytics Accelerator Loader can load data directly into the DB2 Analytics Accelerator table without loading the DB2 for z/OS table (accelerator only option).

It is the responsibility of the user to supply DB2 Analytics Accelerator Loader with a proper source input data set to load into a DB2 Analytics Accelerator accelerated table or a DB2 for z/OS table. Therefore, the input data can come from any source, *if* the input data is in a format that can be loaded by the standard DB2 LOAD utility, and if it is on z/OS.

Delimited input is not currently supported. The user must also supply field specifications (load control cards) to map the positions and lengths of the fields of the input data set records to the columns in the target DB2 or DB2 Analytics Accelerator table. There are many potential sources of data available to load into DB2 Analytics Accelerator. The following list includes some of the more popular options:

- IMS
- VSAM
- Oracle
- Data from other DB2 subsystems

There are various data extraction methods available for users when pulling data from a non-DB2 source to load into the DB2 Analytics Accelerator. For example, an extract, transform, and load (ETL) tool, such as IBM InfoSphere DataStage, can be used to extract and move data to a z/OS data set, or user-created data extraction processes can be used.

Furthermore, other tools can assist the process as well. For example, InfoSphere Classic Federation can be used to make it much easier to extract data from non-relational sources, such as IMS or VSAM.

DB2 Analytics Accelerator Loader external data load functionality simplifies and speeds the process of loading tables into DB2 Analytics Accelerator. Simplification occurs by consolidating the steps needed to load a table on both DB2 for z/OS and DB2 Analytics Accelerator, or by optionally eliminating the need to first load the table on DB2 when the source data is not in DB2 for z/OS. In both scenarios, DB2 Analytics Accelerator Loader reduces the elapsed time and processor usage associated with the loading process.

Preparatory steps for external data

Loading non-DB2 source data into a DB2 Analytics Accelerator table for query acceleration is a multi-step process that requires preparation of the source data before it can be loaded into DB2 Analytics Accelerator:

1. If the source of the data is non-relational, it needs to be mapped to a relational model, such as a DB2 table.
2. The relational table must be created in DB2 for z/OS.

3. The table must be added (defined) to the DB2 Analytics Accelerator.
4. A source data extraction process must occur to produce a data set that can be used as input to the LOAD process.

The process of using the DB2 Analytics Accelerator Loader external load feature to load both DB2 for z/OS and the DB2 Analytics Accelerator in the same LOAD utility is also called *Dual Load*. When using DB2 Analytics Accelerator Loader Dual Load to load data into DB2 for z/OS and DB2 Analytics Accelerator, DB2 Analytics Accelerator Loader performs the following steps:

1. Reads input source data file once.
2. Converts external format data to DB2 internal format data for faster loading.

   This conversion is IBM System z Integrated Information Processor (zIIP)-enabled when zIIP processors are available.
3. Simultaneously sends the data to DB2 for z/OS to run the LOAD utility, and to the DB2 Analytics Accelerator.

The Dual Load option provides the following benefits:

- Eliminates the step to run the UNLOAD utility to first unload data from DB2.
- Reduces overall elapsed time and central processing unit (CPU) use.

The process of using the DB2 Analytics Accelerator Loader external load feature to load only the DB2 Analytics Accelerator is called *external data load*. When using DB2 Analytics Accelerator Loader feature to load data only into DB2 Analytics Accelerator, DB2 Analytics Accelerator Loader performs the following steps:

1. Reads input source data file once.
2. Converts external format data to DB2 internal format data for faster loading.

   This conversion is zIIP-enabled when zIIP processors are available.
3. Bypasses the load into DB2 for z/OS, and simply loads the DB2 Analytics Accelerator.
4. The DB2 for z/OS table is emptied (Load REPLACE with zero rows).

The external data load provides the following benefits:

- Eliminates the manual step to load external data into the DB2 for z/OS table.
- Eliminates the manual step to run the UNLOAD utility to unload data from DB2.
- Reduces elapsed time and CPU use significantly.
- Reduces CPU and storage resource usage on z/OS, because the table on DB2 is not loaded, resulting in a smaller footprint for the DB2 table definition.

When you use the DB2 Analytics Accelerator Loader external data load feature to load non-DB2 or non-z/OS data to only the DB2 Analytics Accelerator, any data in the DB2 for z/OS table will be deleted. The data in the table is deleted by running a LOAD REPLACE and loading zero rows.

If the data in the DB2 table must be retained for any reason, or if the data being stored in DB2 for z/OS tables did not originate in DB2, the Dual Load feature or the Group Consistent Load feature can be used rather than the external data load (accelerator only) feature.

Take into account the following considerations when loading non-DB2 data into only the DB2 Analytics Accelerator using the external data load feature of DB2 Analytics Accelerator Loader:

- The data cannot be updated by DB2, and can only be used for query purposes.
- All queries run against the table being loaded must qualify to run on the DB2 Analytics Accelerator. This might require a change to the application's Structured Query Language (SQL) to ensure that it will run on the DB2 Analytics Accelerator.
- All DB2 security is enforced when using the DB2 Analytics Accelerator Loader, so all data access is still performed through DB2 for z/OS. Therefore, no security is compromised, and no additional security definitions are required.
- The DB2 Optimizer still drives data access for tables on the DB2 Analytics Accelerator. The SET CURRENT QUERY ACCELERATION=ALL statement must be used so that all application's queries are routed to the table on the DB2 Analytics Accelerator. If SET CURRENT QUERY ACCELERATION=ALL is not specified, the Optimizer might send the query to the table on DB2 for z/OS, and not the table on DB2 Analytics Accelerator, causing zero rows or unexpected data to be returned.

Summary of Dual Load and accelerator only load

Figure A-1 illustrates how data is loaded into DB2 Analytics Accelerator using the Dual Load feature of DB2 Analytics Accelerator Loader. Notice that data is not unloaded from the DB2 for z/OS table and sent to DB2 Analytics Accelerator. Instead, the file is read once, and data is loaded directly into DB2 Analytics Accelerator and DB2 for z/OS simultaneously. The load to both DB2 Analytics Accelerator and DB2 for z/OS greatly reduces elapsed time, ensuring that your production DB2 objects and accelerated DB2 for z/OS objects are online sooner.

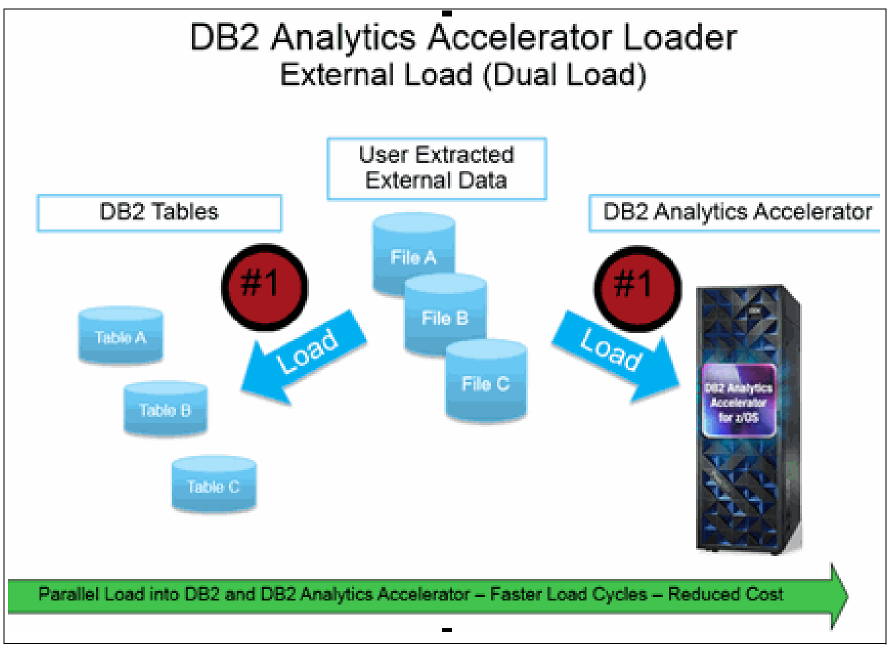


Figure A-1 DB2 Analytics Accelerator Loader Dual Load

Figure A-2 on page 259 illustrates how data is loaded only into the DB2 Analytics Accelerator using DB2 Analytics Accelerator Loader. In this use case, the data is external to DB2 for z/OS and never updated by DB2 for z/OS. Therefore, it is not required in DB2 for z/OS.

However, DB2 for z/OS still requires the table definition to exist. Non-DB2 for z/OS data is loaded directly into DB2 Analytics Accelerator, with a significant reduction in elapsed time and CPU use. DB2 Analytics Accelerator Loader also eliminates the large storage footprint required to store non-DB2 data in DB2 for z/OS.

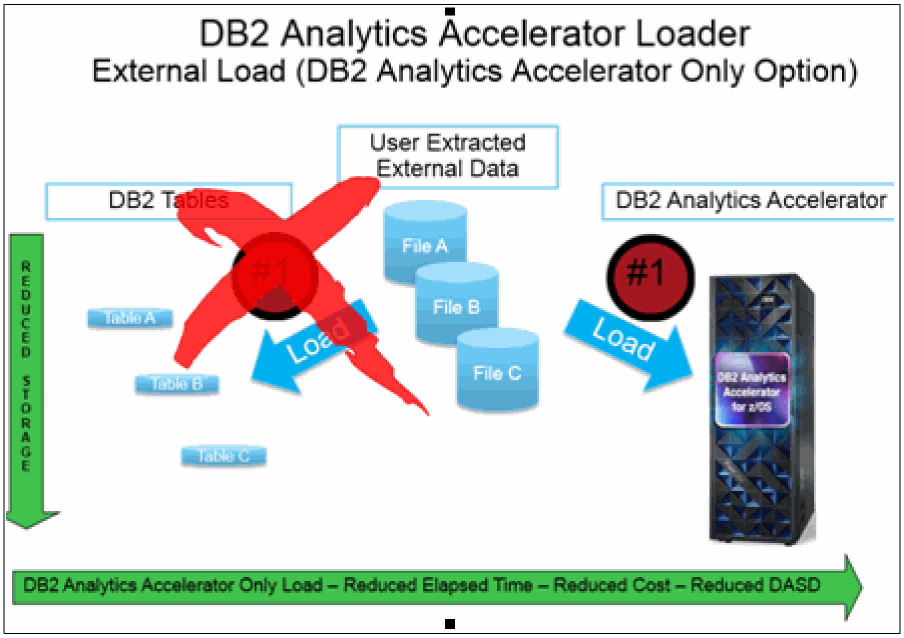


Figure A-2 Accelerator Only Load

Loading non-DB2 data, such as IMS and VSAM data, to DB2 Analytics Accelerator

Even though DB2 Analytics Accelerator was originally developed to store DB2 for z/OS data, companies today have many different data sources across their enterprise, and they want to make the most of the power of DB2 Analytics Accelerator by using it with non-DB2 data. This includes but is not limited to IMS, VSAM, and Oracle data.

If we focus on IMS data, there are several considerations to take into account when developing a process to extract IMS data to load into the DB2 Analytics Accelerator. IMS is a hierarchical database, and does not map directly to a relational model. Consider the following factors when developing a process to extract IMS data to load into DB2 Analytics Accelerator:

- Mapping and transforming data

  A database record in IMS can be made up of many different data elements, called *segments*. There can be one to many occurrences of each segment type in a single database record. Typically, a segment consists of a *key* and several *associated fields*. It is common to map an IMS segment to a table in a relational database management system (DBMS). Each field in a segment relates to a column in the table with an additional key column that will help flatten the record (see the following bullet for more details).

  IMS does not enforce content in a segment based on defined data types. In fact, a data type is not even required to be defined to store data in IMS. As a result, most IMS customers have the data type of each field in a segment defined in Common Business Oriented Language (COBOL) or PL/I copybooks. Any extraction process would most likely need to look at the data structures in these copybooks to understand the data types.

IMS data types are also not the same as in most relational DBMSs. When extracting data, many fields might require a conversion process to convert the IMS data into a format understood by DB2 for z/OS. An example of this would be a data field containing a date in IMS. This field in IMS might be a packed decimal Julian representation of a date. This would require some conversion to an acceptable DB2 format.

- Flattening hierarchical data

  There are two issues with IMS data that might require flattening of data in order for it to be usable for processing by SQL queries:

  - Because IMS is a hierarchical database, some information related to a segment might not be stored with other data in the segment. For instance, say a segment was at the third hierarchical level. To be able to get to a particular instance of a segment, the request to access the data most likely would need to know the key fields of the first- and second-level segments for the segment occurrence.

    When extracting IMS data for SQL processing, the key fields for any parent segments typically need to be stored with the other fields in a segment, and saved in the key column of the table that maps to the IMS segment. These key columns can be used to write SQL in the DBMS to join all of the necessary segment tables together. This results in a relational representation of the IMS data record.

  - Another item that might need to be handled is the existence of structures that might occur multiple times in an individual segment. This might be something, such as OCCURS clauses defined in a COBOL copybook. In this case, typically a separate row or column would be needed for instances of the field or set of fields that can occur multiple times in a single segment.

  Some IMS customers write custom COBOL or PL/I programs to perform the extraction and flattening of IMS data. These programs produce flat files of the IMS data. Other tools can be used to perform this extraction as well, such as IBM File Export for z/OS, or DataStage (an ETL tool).

  Additionally, a tool, such as InfoSphere Classic Federation, can be used to make it easier to extract IMS data. InfoSphere Classic Federation handles many of the data type conversion and flattening issues that were mentioned previously. It also makes IMS or VSAM data appear as just another relational data source to DataStage, making it much easier to build the DataStage job.

  Each extracted segment file maps to a DB2 table, and can be used as input to the DB2 Analytics Accelerator Loader to be loaded into DB2 for z/OS and DB2 Analytics Accelerator. The user must construct the field specifications that map the data extracted from IMS to the appropriate DB2 table that matches the IMS segment.

  A field specification needs to be built that describes the data in the extracted file, and describes how the extracted data maps to the DB2 table columns. The DB2 LOAD utility is capable of performing some transformation by specifying the data type of the field in the extracted file in the field specification.

Figure A-3 on page 261 illustrates the high-level process steps that are needed to load a DB2 for z/OS and DB2 Analytics Accelerator table using IMS as the data source:

1. The IMS source database must be identified and mapped to a DB2 table.
2. An IMS source extraction process must occur.
3. Then the data is loaded into DB2 for z/OS and DB2 Analytics Accelerator using DB2 Analytics Accelerator Loader.

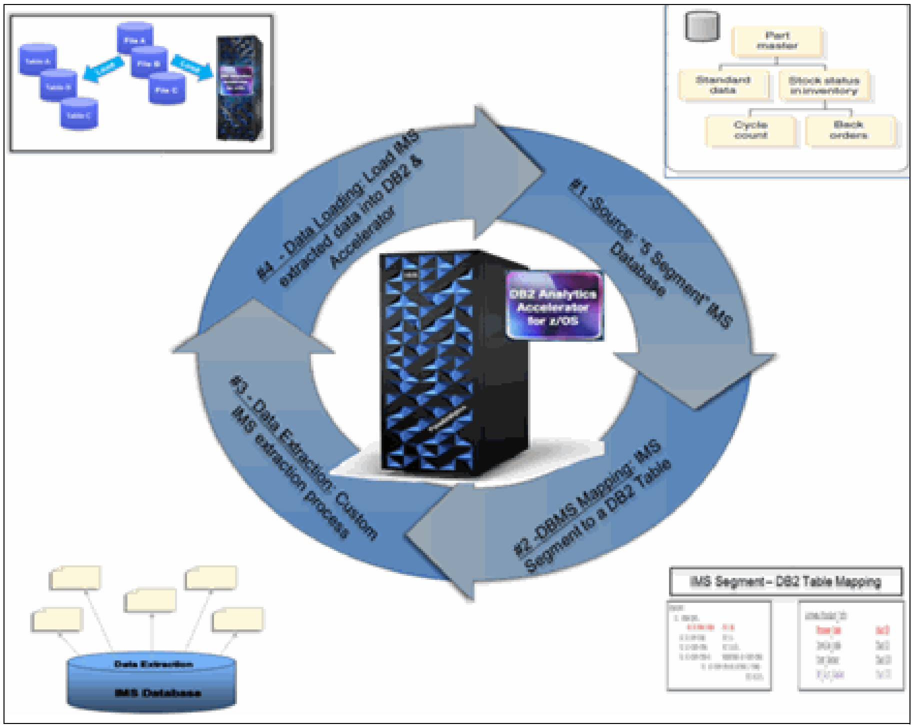


Figure A-3 Loading IMS data into DB2 Analytics Accelerator

Loading the IMS sample database (DI21PART) into DB2 Analytics Accelerator

Figure A-4 shows the structure of the IMS sample database. The root segment is called *Part master*. It has additional segments that store related data.

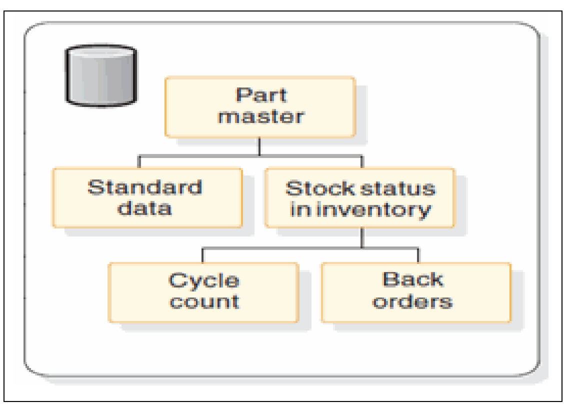


Figure A-4 DI21PART sample database

This section shows how to copy the PARTROOT (part master) and STOKSTAT (stock status) segments to DB2 Analytics Accelerator with InfoSphere Classic Federation and DataStage:

1. The first step is to use InfoSphere Classic Federation to create virtual relational tables on top of these two segments. InfoSphere Classic Federation enables you to take the database descriptor (DBD) and the segment copybooks, and create virtual relational tables on top of IMS segments. InfoSphere Classic Federation handles converting IMS data types into relational data types.

   It can also handle OCCURS and REDEFINES clauses. After the virtual relational tables are defined, we can query the IMS data with standard SQL statements.

These SQL statements use Java Database Connectivity (JDBC) or Open Database Connectivity (ODBC). The IMS data stays in IMS, and InfoSphere Classic Federation converts the incoming SQL statements to Data Language/I (DL/I) calls to access the data.

InfoSphere Classic Federation can also create virtual relational tables on top of VSAM files. Again, it does this using copybooks. Incoming SQL statements are converted into VSAM calls. This is the only step that differs between copying IMS and VSAM data to DB2 Analytics Accelerator. This chapter does not cover the details required to create the InfoSphere Classic Federation virtual relational tables. For more information about these steps, see the IBM InfoSphere Classic Federation Information Center:

http://pic.dhe.ibm.com/infocenter/iisclzos/v11r1/index.jsp

2. After we have created the InfoSphere Classic Federation virtual relational tables for PARTROOT and STOKSTAT, we have two tables that look like Figure A-5 and Figure A-6.

PARTROOT

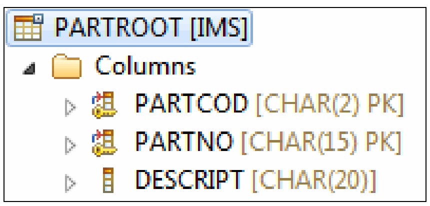


Figure A-5 PARTROOT virtual relational table

STOKSTAT

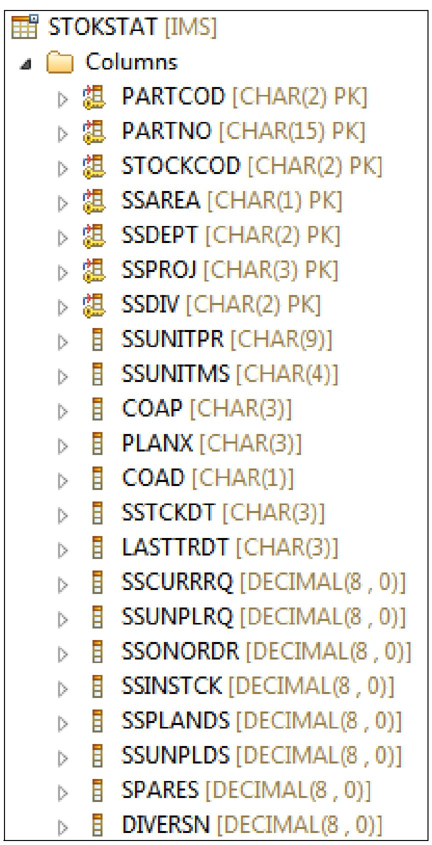


Figure A-6 STOKSTAT virtual relational table

3. The same process must be applied to any IMS segment.

Now that we have our virtual relational tables, the next step is to create a DataStage job to read data from these tables using SELECT statements, and convert the data into the appropriate format for DB2 Analytics Accelerator Loader.

DataStage job to convert IMS data

Before you can create DataStage jobs that use InfoSphere Classic Federation, you first must configure DataStage to use InfoSphere Classic Federation. For more information about how to configure DataStage to use InfoSphere Classic Federation, see the following website:

http://pic.dhe.ibm.com/infocenter/iisinfsv/v9r1/index.jsp?topic=%2Fcom.ibm.swg.im.iis.ds.zosfilestage.doc%2Ftopics%2Fconfiguring_zosfile.html

In this example, we are going to assume that the IMS data will go into DB2 as is, and that the DB2 table structures will exactly match the structures of the virtual relational tables. DataStage obviously has many capabilities for data transformation and data type conversions. However, we will keep this example simple. We will show some examples of data type conversions when we cover Oracle data later in this chapter.

Our DataStage job is simple. It consists of two InfoSphere Classic Federation stages and two Complex Flat File stages. Each InfoSphere Classic Federation stage is responsible for extracting the data for one IMS segment. Each Complex Flat File stage is responsible for creating the input to DB2 Analytics Accelerator Loader for one DB2 table. The job looks like that shown in Figure A-7.

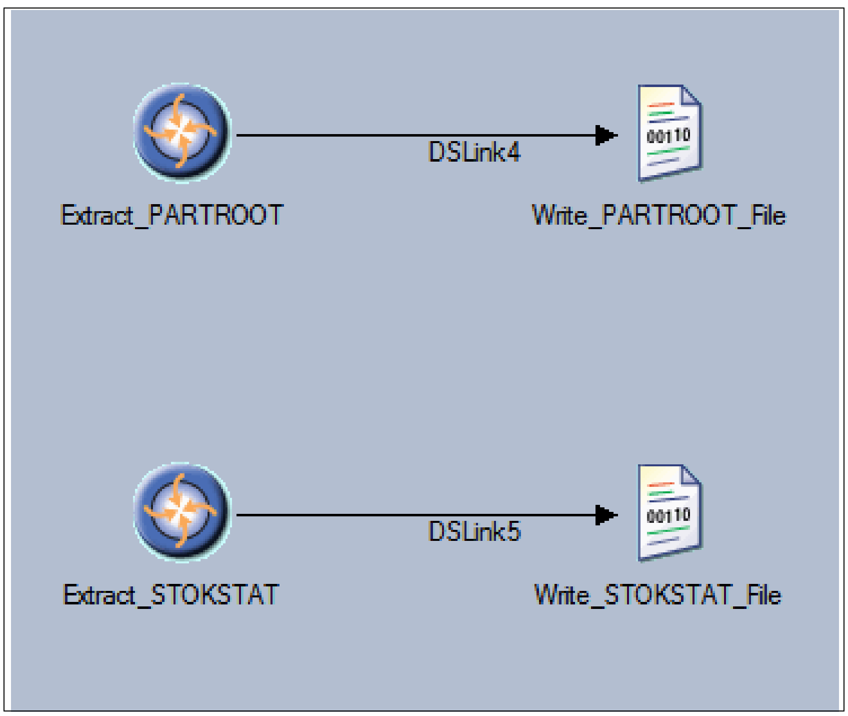


Figure A-7 DataStage job for preparing IMS Data

When you create a DataStage job like this, with disjoint sections, all of those sections run in parallel. In this case, therefore, we process the data for PARTROOT and the data for STOKSTAT in parallel.

The InfoSphere Classic Federation stages are easy to configure, as shown in Figure A-8.

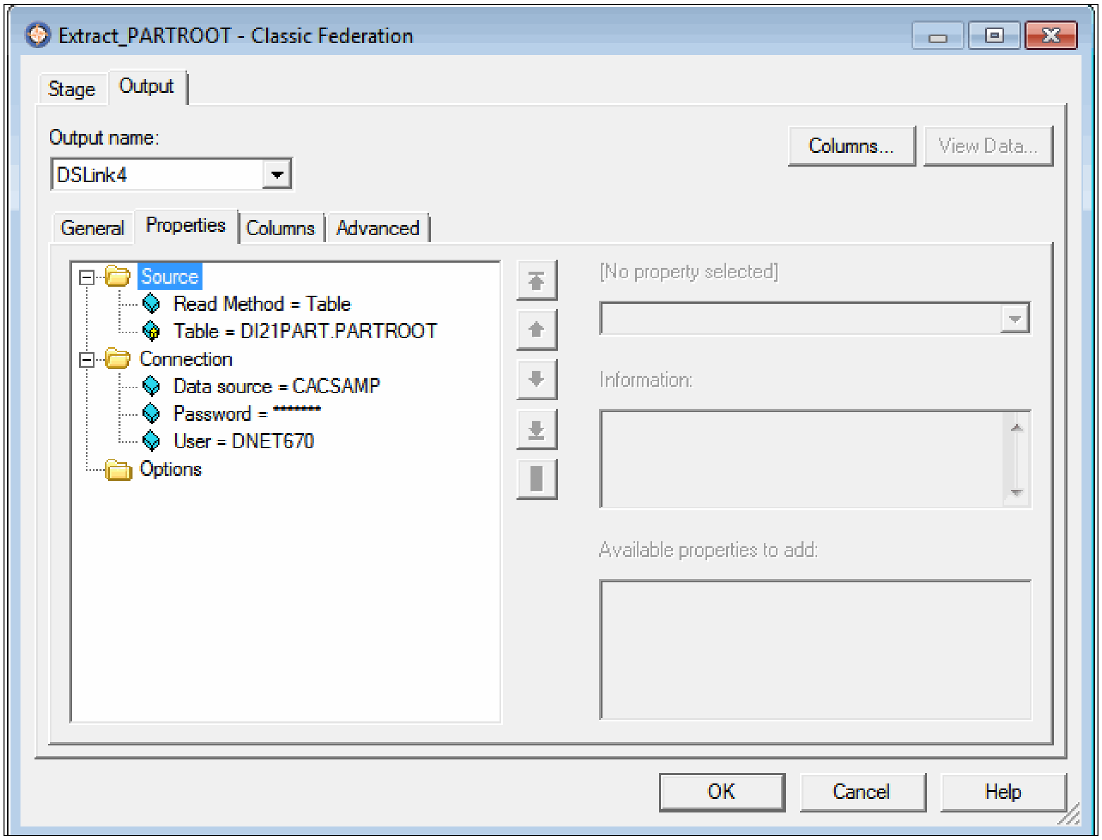


Figure A-8 InfoSphere Classic Federation stage properties

To configure the InfoSphere Classic Federation stage, follow these steps:

1. As you can see in Figure A-8, there are not many properties that you need to set. Provide the following information:
   a. Complete the name of the virtual relational table as defined to InfoSphere Classic Federation.
   b. Provide a user ID and password for connecting to InfoSphere Classic Federation.
   c. Provide the InfoSphere Classic Federation data source name.

      The data source name is set when the InfoSphere Classic Federation server is configured. The default value is CACSAMP.
2. Next, you have to complete the information on the Columns tab. The easiest way to do this is to use the ODBC table import wizard to have DataStage extract all of the column information from the InfoSphere Classic Federation server.
3. The table metadata is then saved as part of your DataStage project.

4. You can click **Load** on the columns page to set the column information for the InfoSphere Classic Federation stage. See Figure A-9.

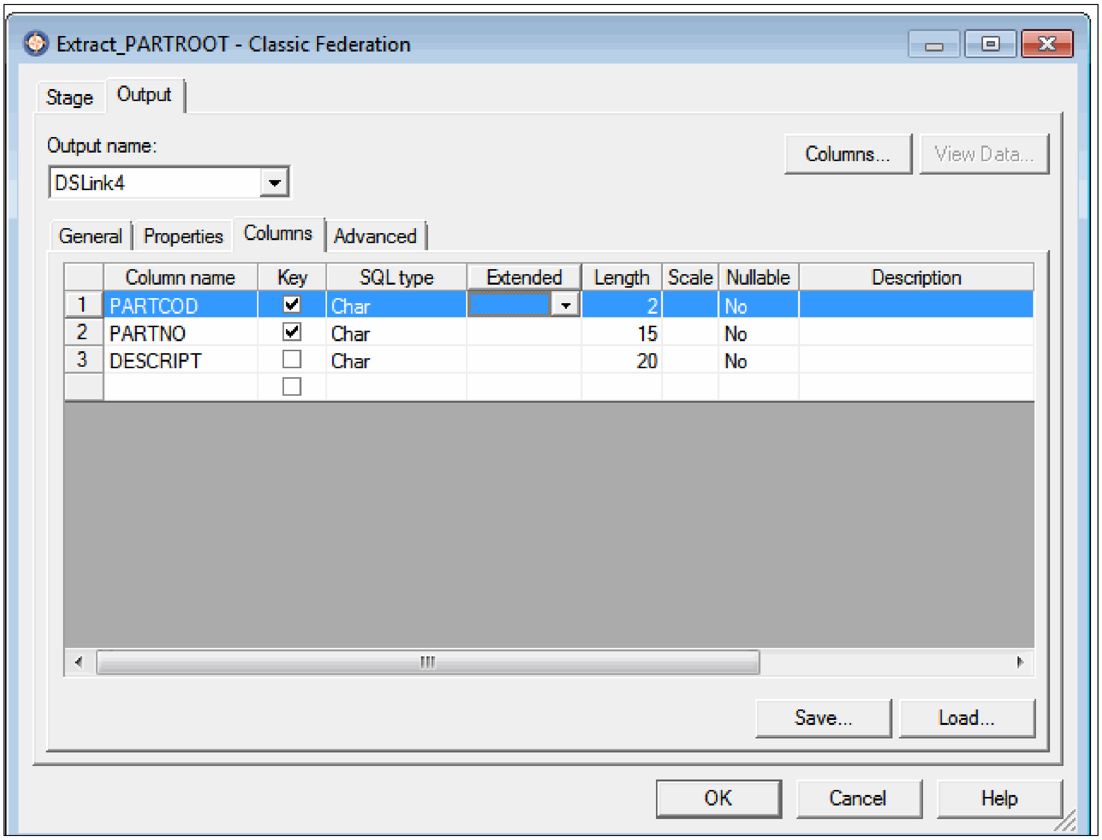


Figure A-9 InfoSphere Classic Federation stage Columns tab

5. The exact same process can be done for STOKSTAT.

Next, we need to set the correct properties in the Complex Flat File stages. Our goal is to produce Extended Binary Coded Decimal Interchange Code (EBCDIC) files that can be sent to the mainframe using FTP[1], and used by DB2 Analytics Accelerator Loader.

To set the Complex Flat File stages properties, follow these steps:

1. On the File Options tab, provide a file name of where the data should be written. This file will exist on the DataStage server. See Figure A-10.

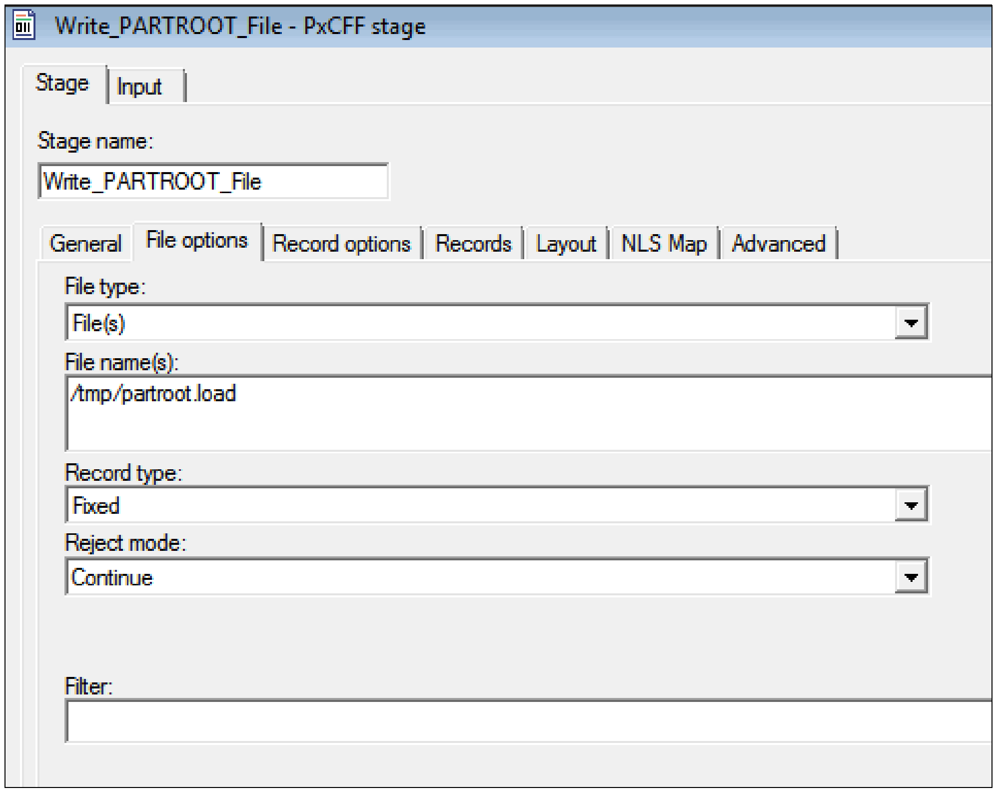


Figure A-10 File Options Tab

[1] FTP is the File Transfer Protocol, a standard network protocol used to transfer computer files from one host to another host over a TCP-based network.

2. On the Record Options tab, select that we are writing a binary big-endian EBCDIC file. See Figure A-11.

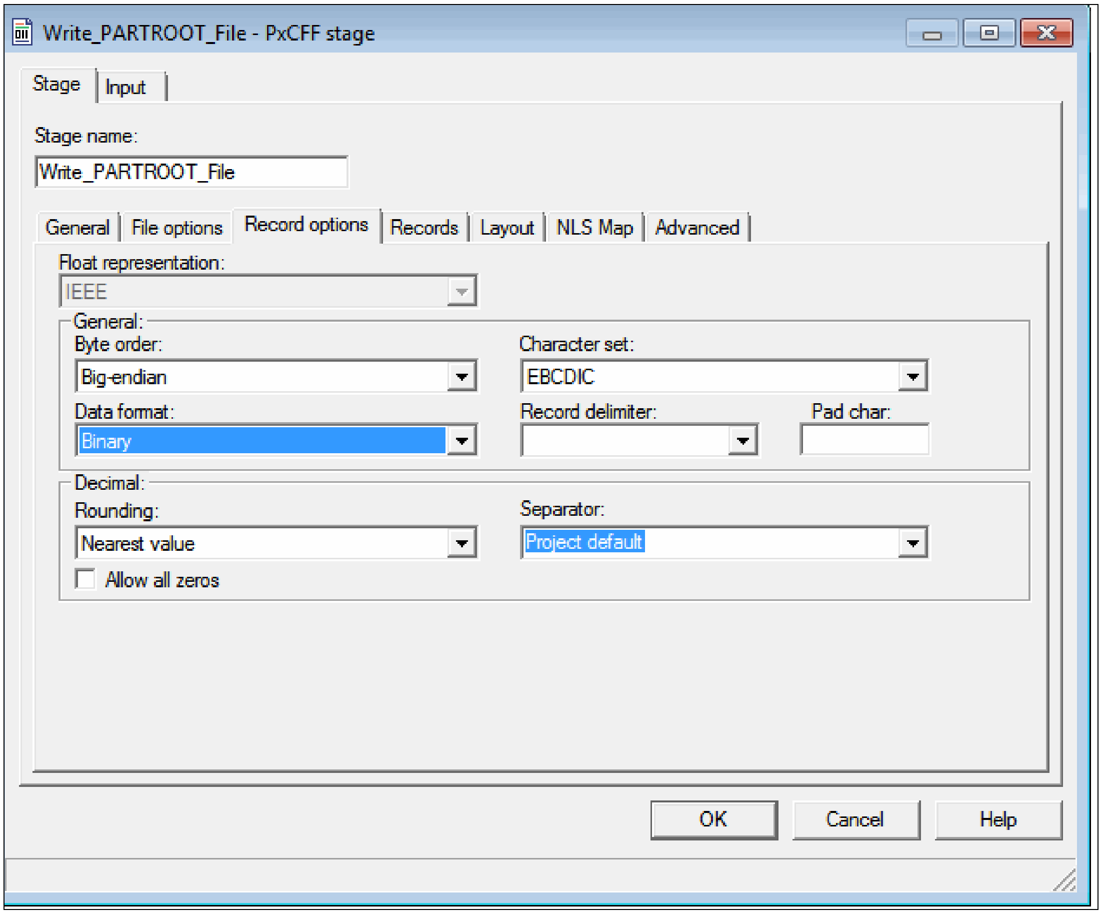


Figure A-11 Record Options Tab

3. You then might need to change a few things on the Records tab:
   a. For fields that are going into DB2 CHAR columns, make sure that the native type is set to CHARACTER, and that the length is set.
   b. For fields that are going into DB2 SMALLINT, INTEGER, or BIGINT columns, make sure the native type is set to BINARY. Also, set the length field to the appropriate value:
      i. For a SMALLINT, 4
      ii. For an INTEGER, 9
      iii. For BIGINT, 18.

      Also, verify that the Usage property is set to COMP.
   c. For fields that are going into DB2 DECIMAL columns, confirm the following settings:
      i. The native type is set to DECIMAL.
      ii. The length and scale are set.
      iii. The Usage property is set to COMP-3.

4. Lastly, click the Layout tab and then select the COBOL radio button. Take note of the ending column value for the last field. This tells us the length of the record, which we will need to know when we send the file using FTP the file. See Figure A-12.

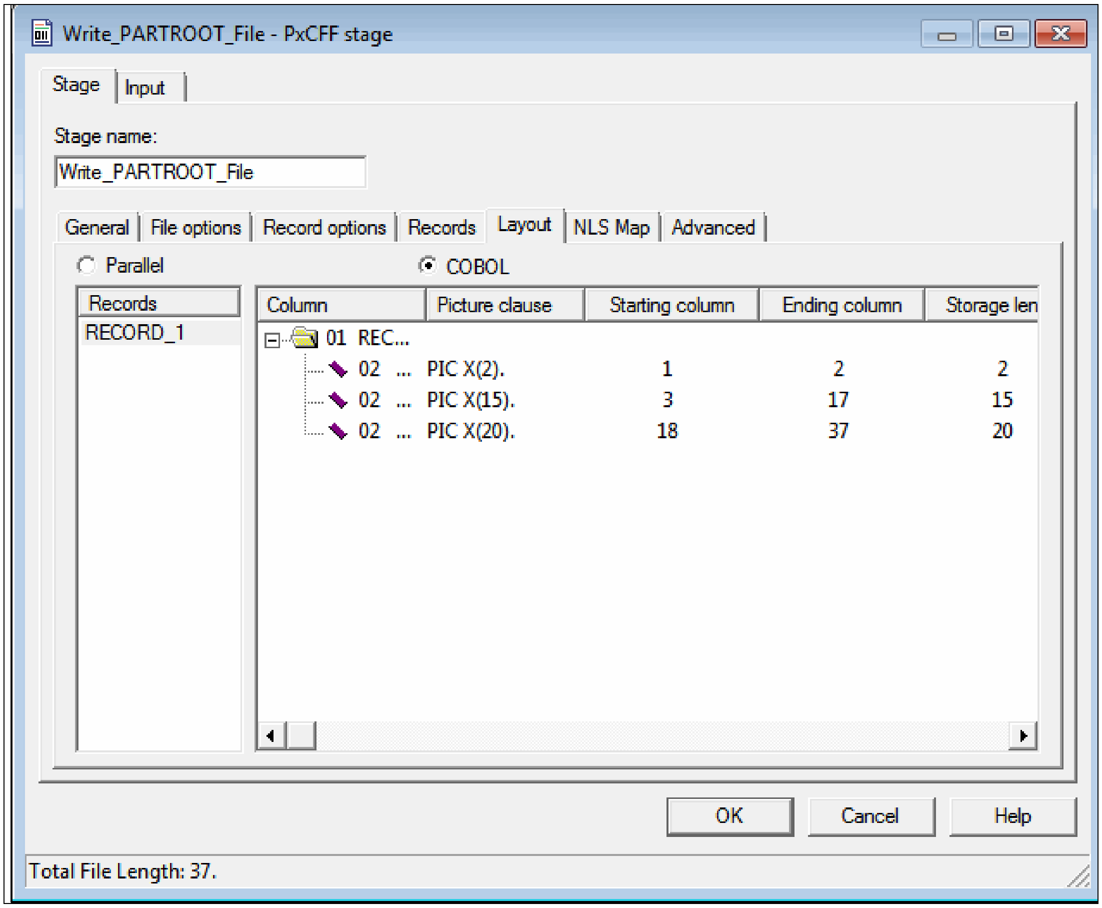


Figure A-12 The ending column value on line 3 is the record length

5. Follow the exact same steps for other Complex Flat File stages.

At this point, we are ready to create the output files:

1. Save, compile, and run the job.

   When this job completes, we will have both of our output files in the `/tmp` directory of the DataStage server.

2. The next step is to use FTP to send these files to z/OS. The process is the same for both files. The only thing that changes is the record length. For example, we used the following commands to send the PARTROOT file to z/OS:

```
ftp 9.12.44.159
site recfm=FB
site lrecl=37
bin
put /tmp/partroot.load PARTROOT.LOAD
```

Lastly, we need to create the field specification list that will be used by DB2 Analytics Accelerator Loader. This will get saved in a separate data set on z/OS. Also, don't forget that you still need to create DB2 tables and add them to DB2 Analytics Accelerator even if you are only loading into DB2 Analytics Accelerator. Creating the field specification list is simple. The DataStage layout tab gives you all of the starting and ending position information you need.

To create the field specification list, follow these steps:

1. CHAR columns should be defined as CHAR(length) in the field specification.
2. SMALLINT, INTEGER, and BIGINT columns should be defined as is.
3. DECIMAL columns should be defined as DECIMAL PACKED.
4. Make sure to separate lines with commas, as in the field specification for PARTROOT:

```
PARTCOD POSITION(1:2) CHAR(2),
PARTNO POSITION(3:17) CHAR(15),
DESCRIPT POSITION(18:37) CHAR(20)
```

Now that you have the DB2 tables created, the field specification written, and the data to be loaded in the proper format, you can create either external load jobs (DB2 Analytics Accelerator only) or Dual Load job (load both DB2 Analytics Accelerator and DB2 for z/OS) following the steps outlined in Chapter 10, “DB2 Analytics Accelerator Loader” on page 213.

Loading Oracle data to DB2 Analytics Accelerator

In this section, we show how to load three of the Oracle sample tables into DB2 Analytics Accelerator. We are working with HR.COUNTRIES, HR.LOCATIONS, and HR.EMPLOYEES. These tables are defined in Oracle as shown in Example A-1.

Example: A-1 Oracle sample tables

```
HR.COUNTRIES
Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 COUNTRY_ID                                NOT NULL CHAR(2)
 COUNTRY_NAME                                       VARCHAR2(40)
 REGION_ID                                          NUMBER

HR.LOCATIONS
Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 LOCATION_ID                               NOT NULL NUMBER(4)
 STREET_ADDRESS                                     VARCHAR2(40)
 POSTAL_CODE                                        VARCHAR2(12)
 CITY                                      NOT NULL VARCHAR2(30)
 STATE_PROVINCE                                     VARCHAR2(25)
 COUNTRY_ID                                         CHAR(2)

HR.EMPLOYEES
Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMPLOYEE_ID                               NOT NULL NUMBER(6)
 FIRST_NAME                                         VARCHAR2(20)
 LAST_NAME                                 NOT NULL VARCHAR2(25)
 EMAIL                                     NOT NULL VARCHAR2(25)
 PHONE_NUMBER                                       VARCHAR2(20)
 HIRE_DATE                                 NOT NULL DATE
 JOB_ID                                    NOT NULL VARCHAR2(10)
 SALARY                                             NUMBER(8,2)
 COMMISSION_PCT                                     NUMBER(2,2)
 MANAGER_ID                                         NUMBER(6)
 DEPARTMENT_ID                                      NUMBER(4)
```

Obviously, there are many different ways that this data can be mapped to DB2 for z/OS data types. So, the first thing we should do is figure out how we want our DB2 for z/OS tables to be defined. We decided to define our DB2 tables as shown by the DDL in Example A-2.

Example: A-2 DDL for the sample tables

```
CREATE TABLE COUNTRIES(COUNTRY_ID CHAR(2) PRIMARY KEY,
                       COUNTRY_NAME VARCHAR(40),
                       REGION_ID INTEGER);
CREATE TABLE LOCATIONS(LOCATION_ID DECIMAL(4,0) PRIMARY KEY,
                       STREET_ADDRESS VARCHAR(40),
                       POSTAL_CODE VARCHAR(12),
                       CITY VARCHAR(30),
                       STATE_PROVINCE VARCHAR(25),
                       COUNTRY_ID CHAR(2));
CREATE TABLE EMPLOYEES(EMPLOYEE_ID DECIMAL(6,0) PRIMARY KEY,
                       FIRST_NAME VARCHAR(20),
                       LAST_NAME VARCHAR(25),
                       EMAIL VARCHAR(25),
                       PHONE_NUMBER VARCHAR(20),
                       HIRE_DATE DATE,
                       JOB_ID VARCHAR(10),
                       SALARY DECIMAL(8,2),
                       COMMISSION_PCT DECIMAL(2,2),
                       MANAGER_ID DECIMAL(6,0),
                       DEPARTMENT_ID DECIMAL(4,0));
```

DataStage job to convert Oracle data

Before you can create DataStage jobs that connect to Oracle, you first must configure DataStage to use the Oracle client. For more information about how to configure DataStage to use the Oracle client, see the following website:

http://pic.dhe.ibm.com/infocenter/iisinfsv/v9r1/index.jsp?topic=%2Fcom.ibm.swg.im.iis.conn.common.usage.doc%2Ftopics%2Ft_configuring_access_oracle_db.html

Our DataStage job is going to be simple:

1. We first use an Oracle Enterprise stage to read the data from the Oracle table.
2. We then pass that data to a Transformer stage to do any necessary data transformations.
3. Lastly, we will pass the output of the Transformer stage to a Complex Flat File stage, which will write the data to an output file.

Because we are creating three tables, we have three sets of these stages all in one job. That way, all three tables are processed in parallel. The job looks like Figure A-13.

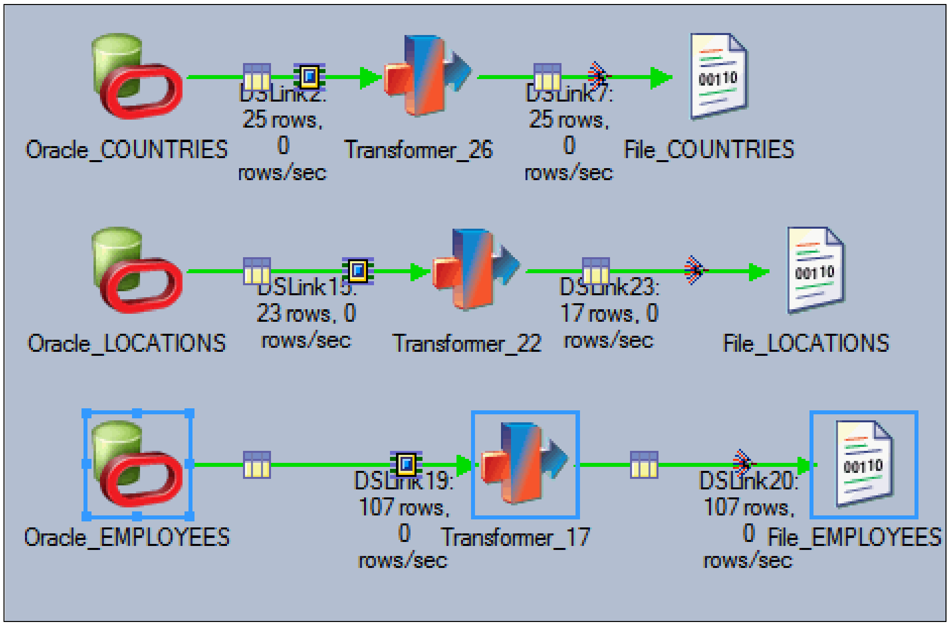


Figure A-13 DataStage Job to convert oracle data

The Oracle Enterprise stages are easy to configure, as shown in Figure A-14.

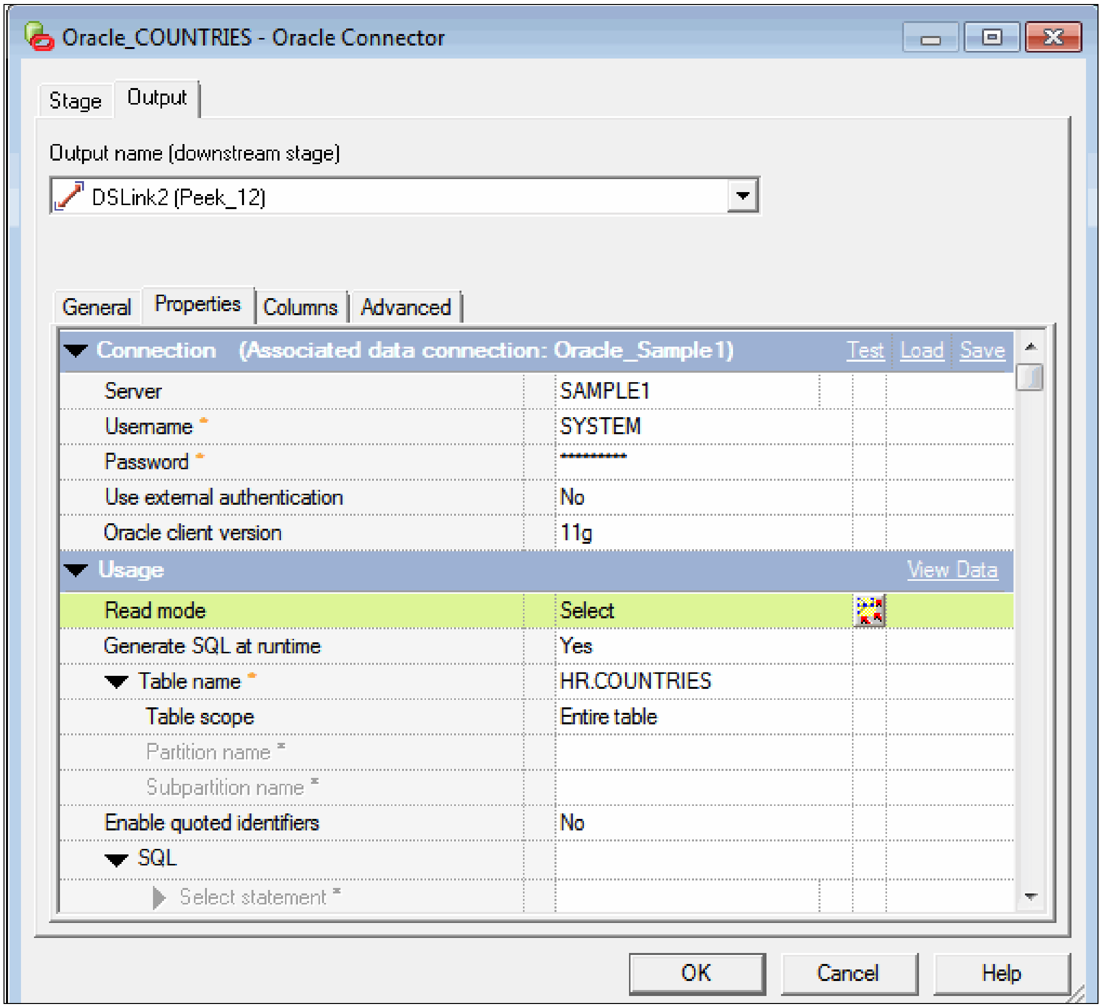


Figure A-14 Oracle Enterprise Stage Properties

As you can see, we followed these steps:

1. Set the Oracle connection information.
2. Choose a read mode of select.
3. Set generate SQL at run time to Yes.
4. Complete the table name.
5. Next, you have to complete the information on the Columns tab.

   The easiest way to do this is to use the Connector table import wizard to have DataStage extract all of the column information from Oracle. The table metadata is then saved as part of your DataStage project.

6. You can click **Load** on the columns page to set the column information for the Oracle Enterprise stage. See Figure A-15.

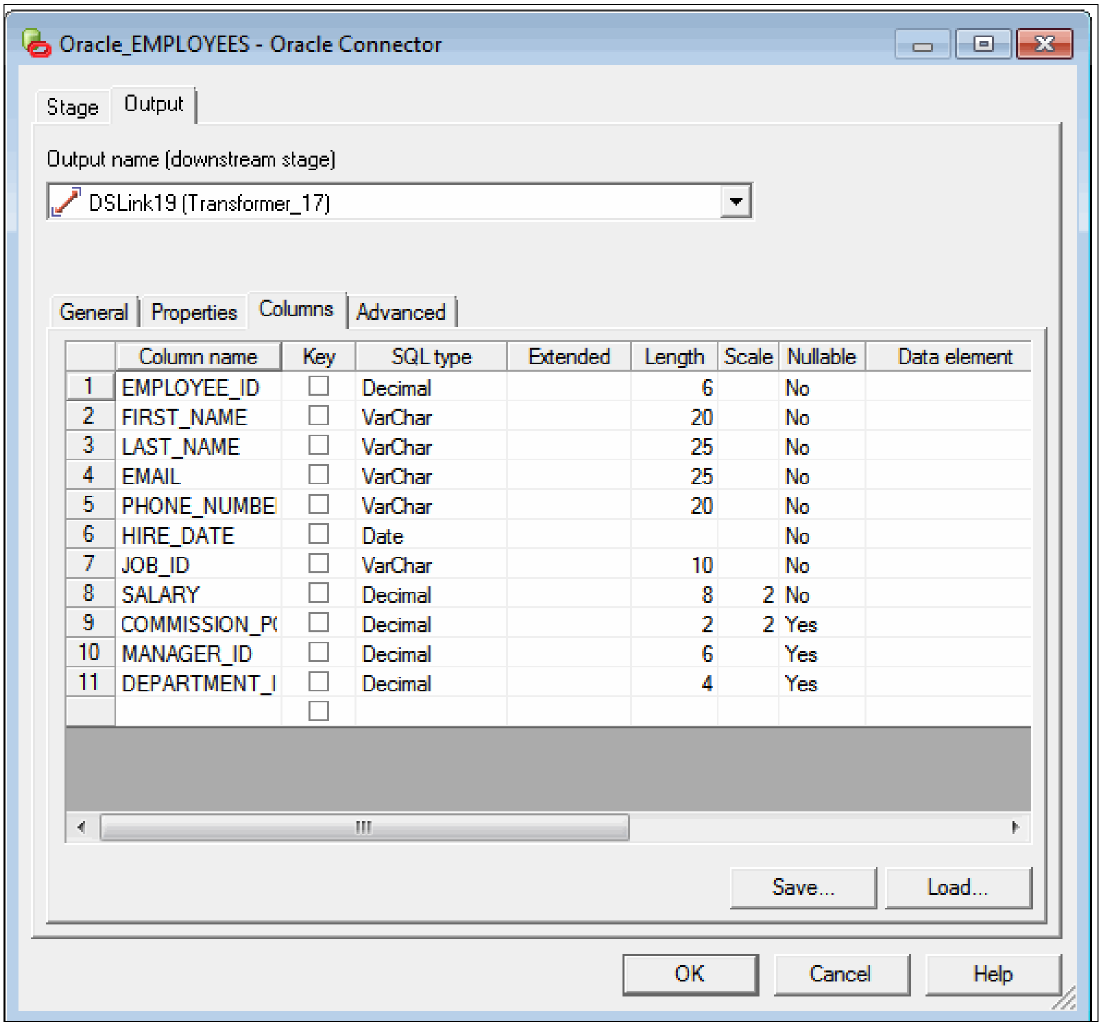


Figure A-15 Oracle Enterprise Stage Columns Tab

7. The other two Oracle Enterprise stages can be configured in the same manner.

Next, we need to configure the Transformer stages. The Transformer stages will be used to do two types of data conversion. It will be used to convert date columns to string representations, and to handle null values:

1. The first thing to do in the Transformer stage is to copy all input columns to output columns.
2. Next, we will handle date columns.

   For all columns that are dates, we need to edit the output expression. Basically, we are trying to convert the date into a string expression so that it can be written to the output file as character data and loaded into the DB2 DATE column as DATE EXTERNAL. Out of the three tables that we are working with, we only have one date column, which is the HIRE_DATE column in HR.EMPLOYEES. Our output expression for the HIRE_DATE column looks like the following expression:

```
DateToString(DSLink19.HIRE_DATE, "%yyyy-%mm-%dd")
```

We also had to change the data type of the output HIRE_DATE column from DATE to CHAR(10).

3. Next, we need to go back through our Transformer stages and change the output expressions for any columns that have null values. We need to pick a value for each column that we can use to represent nulls:

   – For example, for character data, it might be the string ‘NULL’ or ‘NONE’.
   – For numeric data, it might be negative or all 9’s.

   You can pick a different value to represent null for each column that has nulls, if you so choose. After you’ve picked the value to represent nulls, you need to change the output expression for the column so that it replaces nulls with the value that you have chosen. For example, for the MANAGER_ID column in HR.EMPLOYEES, I chose to use -1 to represent nulls. Therefore, the follow expression is our column expression:

   ```
   if IsNull(DSLink19.MANAGER_ID) then -1 else  DSLink19.MANAGER_ID
   ```

4. Next, we need to handle the fields that will go into VARCHAR columns in DB2. For each one of these fields, you need to precede it with a 2 byte field that will hold the length of the VARCHAR data:

   a. To do this, add a new column on the output side immediately before the VARCHAR field.

   b. The data type for the new column should be SMALLINT. You will need to assign an expression to each of these new SMALLINT fields that evaluates to the length of the VARCHAR data:

      ```
      Len(DSLink19.FIRST_NAME)
      ```

      This is the expression to use for the smallint field that immediately proceeds the FIRST_NAME field in the HR.EMPLOYEES data.

5. Lastly, you should change the metadata for the output columns, so that all of them are non-nullable. The final output column metadata for the HR.EMPLOYEES table looks like Figure A-16.

Column name	Key	SQL type	Extended	Length	Scale	Nullable	Description
EMPLOYEE_ID	☐	Decimal		6		No	
FIRST_LEN	☐	Integer		4		No	
FIRST_NAME	☐	Char		20		No	
LAST_LEN	☐	Integer		4		No	
LAST_NAME	☐	Char		25		No	
EMAIL_LEN	☐	Integer		4		No	
EMAIL	☐	Char		25		No	
PHONE_LEN	☐	Integer		4		No	
PHONE_NUMBER	☐	Char		20		No	
HIRE_DATE	☐	Char		10		No	
JOB_LEN	☐	Integer		4		No	
JOB_ID	☐	Char		10		No	
SALARY	☐	Decimal		8	2	No	
COMMISSION_PCT	☐	Decimal		2	2	No	
MANAGER_ID	☐	Decimal		6		No	
DEPARTMENT_ID	☐	Decimal		4		No	
	☐						

Figure A-16 Transformer Stage output column metadata

Next, we need to set the correct properties in the Complex Flat File stages. Our goal is to produce EBCDIC files that can be sent using FTP to the mainframe and used by DB2 Analytics Accelerator Loader:

1. On the File Options tab, provide a file name of where the data should be written. See Figure A-17. This file will exist on the DataStage server.

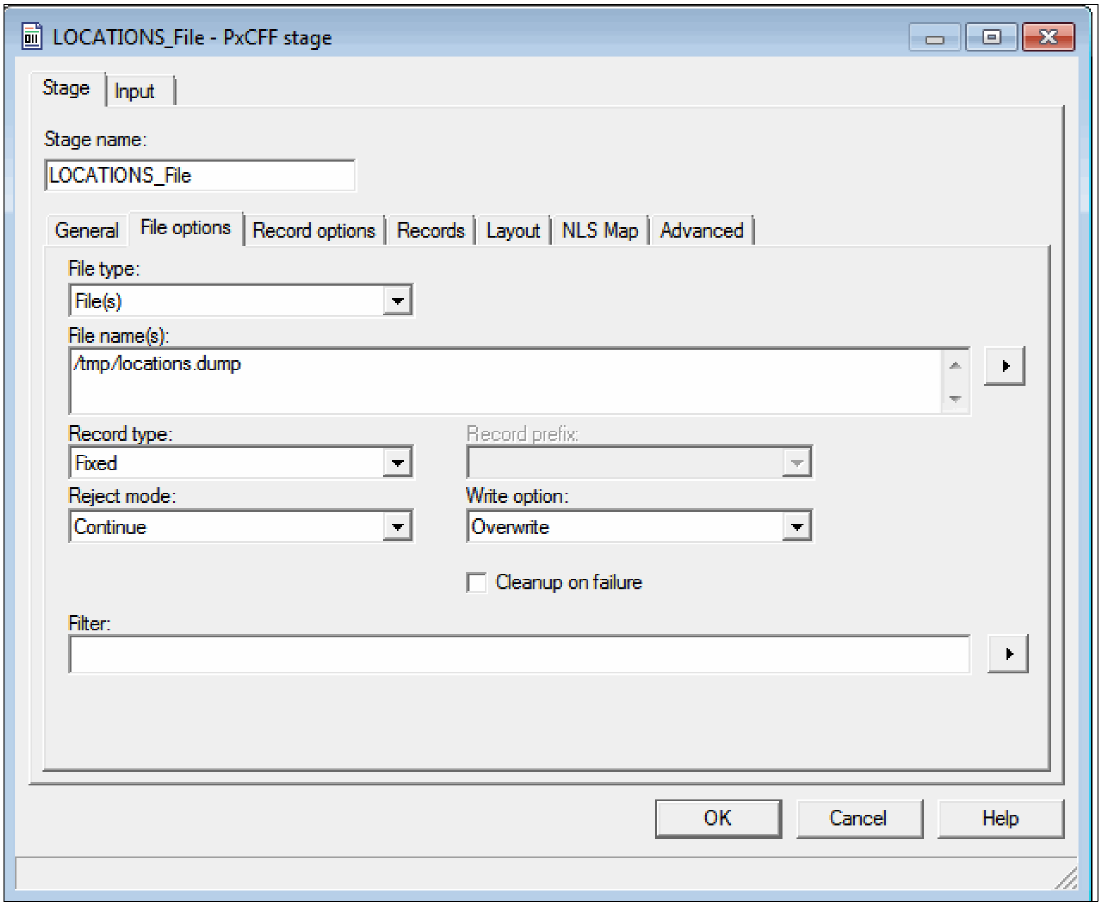


Figure A-17 File Options Tab

2. On the Record Options tab, select that we are writing a binary big-endian EBCDIC file. See Figure A-18.

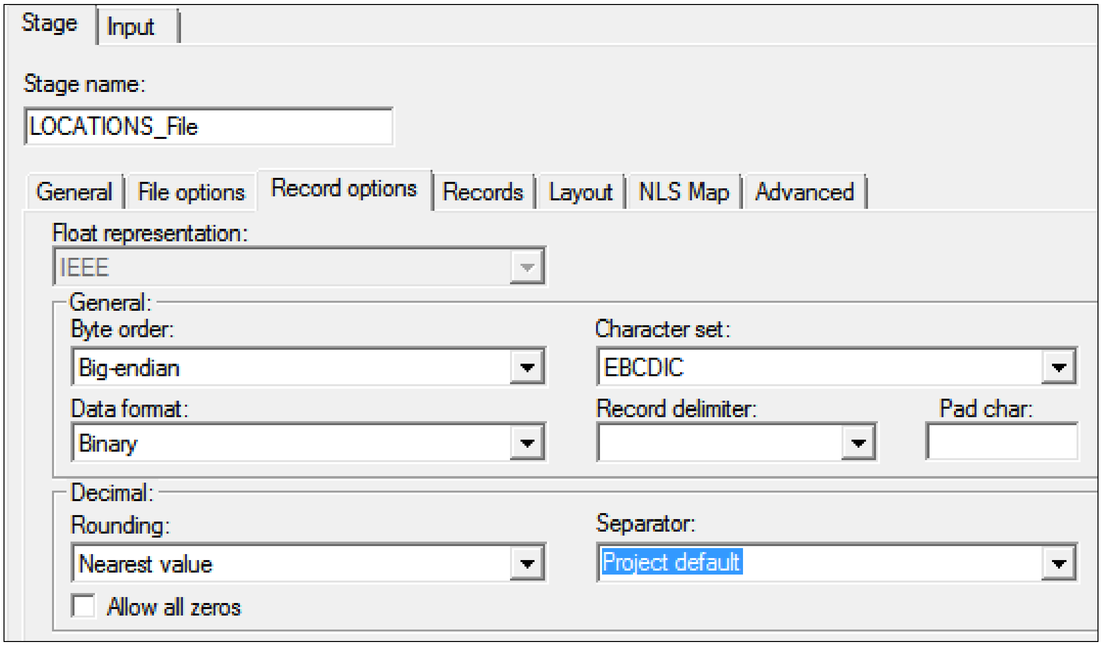


Figure A-18 Record Options Tab

3. Lastly, you might need to change a few things on the Records tab:

   a. For fields that are going into DB2 CHAR columns, make sure the native type is set to CHARACTER, and that the length is set.

   b. For fields that are going into DB2 DATE columns, make sure the native type is set to CHARACTER, and that the length is set to 10.

   c. For fields that are going into DB2 SMALLINT, INTEGER, or BIGINT columns, make sure the native type is set to BINARY. Also, set the length field to the appropriate value:

      i. For a SMALLINT, 4
      ii. For an INTEGER, 9
      iii. For BIGINT, 18

      Also, verify that the Usage property is set to COMP.

   d. For fields that are going into DB2 DECIMAL columns, confirm the following settings:

      i. The native type is set to DECIMAL.
      ii. The length and scale are set.
      iii. The Usage property is set to COMP-3.

e. Lastly, for fields that are going into DB2 VARCHAR columns, set the native type to CHARACTER. You also need to set the native type for the proceeding length field to BINARY and set a length of 4. See Figure A-19.

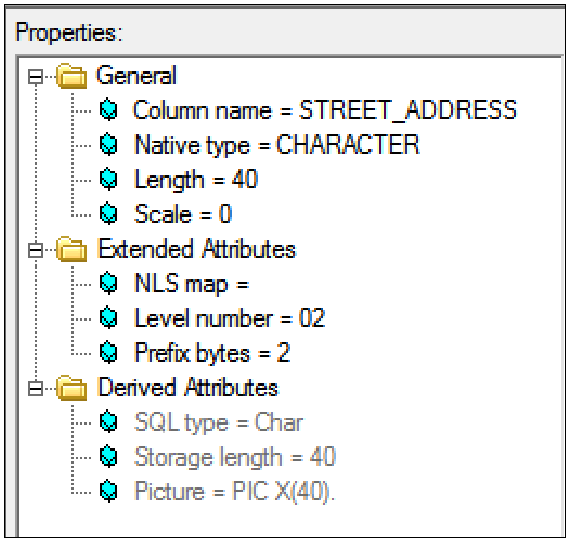
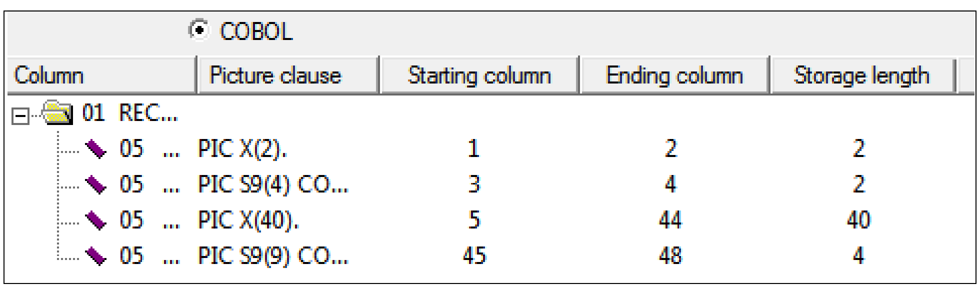


Figure A-19 VARCHAR columns should be defined as CHARACTER

4. For now, the replacement values that you used to represent nulls will be loaded into DB2 as is.
5. Lastly, click the Layout tab and then the COBOL radio button. Take note of the ending column value for the last field. This tells us the length of the record, which we will need to know when we send the file using FTP.



COBOL

Column	Picture clause	Starting column	Ending column	Storage length
01 REC...				
05 ...	PIC X(2).	1	2	2
05 ...	PIC S9(4) CO...	3	4	2
05 ...	PIC X(40).	5	44	40
05 ...	PIC S9(9) CO...	45	48	4

Figure A-20 The ending column value on line 3 is the record length (HR.COUNTRIES)

6. Follow the exact same steps for other Complex Flat File stages.

At this point we are ready to create the output files:

1. Save, compile, and run the job. When this job completes, we will have all of our output files in the `/tmp` directory of the DataStage server.
2. The next step is to send these files to z/OS using FTP. The process is the same for all files, the only thing that changes is the record length (LRECL). The LRECL value comes from the ending column value of the last field, as shown in the DataStage COBOL layout view. For example, we used the following commands to send the COUNTRIES file to z/OS FTP:

```
ftp 9.12.44.159
site recfm=FB
site lrecl=48
bin
put /tmp/countries.dump COUNTRYS.DUMP
```

Lastly, we need to create the field specification list that will be used by DB2 Analytics Accelerator Loader. Also, don't forget that you still need to create DB2 tables and add them to DB2 Analytics Accelerator even if you are only loading into DB2 Analytics Accelerator.

The DataStage layout tab can be used to help us create the field specification, but it isn't perfect, because it doesn't take into account the 2 byte length fields that we added for each VARCHAR column. We have to manually account for those when computing offsets for the field specification. Follow these steps to create the field specification:

1. Define the columns:
    a. CHAR columns should be defined as CHAR(length) in the field specification.
    b. SMALLINT, INTEGER, and BIGINT columns should be defined as is.
    c. DECIMAL columns should be defined as DECIMAL PACKED.
    d. VARCHAR columns are defined as VARCHAR, with no length.
    e. Lastly, DATE columns are specified as DATE EXTERNAL(10).
2. Make sure to separate lines with commas, as shown in Example A-3.

Example: A-3 Field specifications

```
HR.COUNTRIES
COUNTRY_ID   POSITION(1:2)   CHAR(2),
COUNTRY_NAME POSITION(3)     VARCHAR,
REGION_ID    POSITION(45:48) INTEGER
Notice how the REGION_ID field got moved to start at position 45 instead of 43,
because we had to add in the 2 byte length field for COUNTRY_NAME.

HR.LOCATIONS
LOCATION_ID    POSITION(1:3)     DECIMAL PACKED,
STREET_ADDRESS POSITION(4)       VARCHAR,
POSTAL_CODE    POSITION(46)      VARCHAR,
CITY           POSITION(60)      VARCHAR,
STATE_PROVINCE POSITION(92)      VARCHAR,
COUNTRY_ID     POSITION(119:120) CHAR(2)

HR.EMPLOYEES
EMPLOYEE_ID    POSITION(1:4)     DECIMAL PACKED,
FIRST_NAME     POSITION(5)       VARCHAR,
LAST_NAME      POSITION(27)      VARCHAR,
EMAIL          POSITION(54)      VARCHAR,
PHONE_NUMBER   POSITION(81)      VARCHAR,
HIRE_DATE      POSITION(103:112) DATE EXTERNAL(10),
JOB_ID         POSITION(113)     VARCHAR,
SALARY         POSITION(125:129) DECIMAL PACKED,
COMMISSION_PCT POSITION(130:131) DECIMAL PACKED,
MANAGER_ID     POSITION(132:135) DECIMAL PACKED,
DEPARTMENT_ID  POSITION(136:138) DECIMAL PACKED
```

Now that you have the DB2 tables created, the field specification written, and the data to be loaded in the proper format, you can do either Accelerator only loads or Dual Loads following the steps outlined in Chapter 10, "DB2 Analytics Accelerator Loader" on page 213.

Index

Symbols

E

F

G

H

Q

R

S

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications in this list might be available in softcopy only.

- *DB2 11 for z/OS Technical Overview*, SG24-8180
- *DB2 11 for z/OS Performance Topics,* SG23-8222
- *Hybrid Analytics Solution using IBM DB2 Analytics Accelerator for z/OS V3.1*, SG24-8151
- *IBM DB2 Analytics Accelerator: High Availability and Disaster Recovery*, REDP-5104
- *Optimizing DB2 Queries with IBM DB2 Analytics Accelerator for z/OS*, SG24-8005
- *OS/390 MVS Parallel Sysplex Capacity Planning*, SG24-4680
- *Capacity Planning for Business Intelligence Applications: Approaches and Methodologies*, SG24-5689

You can search for, view, download or order these documents and other Redbooks, Redpapers, Web Docs, draft and additional materials, at the following website:

ibm.com/redbooks

Other publications

These publications are also relevant as further information sources:

- *DB2 11 for z/OS Messages*, GC19-4062
- *IBM DB2 Analytics Accelerator for z/OS Version 4.1.0 Getting Started*, GH12-7041
- *IBM DB2 Analytics Accelerator for z/OS Version 4.1.0 User's Guide*, SH12-7040
- *IBM DB2 Analytics Accelerator Loader for z/OS User's Guide,* SC19-4165
- *IBM DB2 Analytics Accelerator Version 4.1.0 Stored Procedures Reference*, SH12-7039
- *IBM DB2 Analytics Accelerator for z/OS Version 4.1.0 Installation Guide*, SH12-7038
- *IBM Tivoli OMEGAMON XE for DB2 Performance Expert on z/OS Version 5.2.0 Report Reference*, SH12-6991

Online resources

These websites are also relevant as further information sources:

- Product support content for DB2 Analytics Accelerator for z/OS

  http://www-947.ibm.com/support/entry/portal/product/information_management/db2_analytics_accelerator_for_z/os?productContext=1003251075

- IBM DB2 Analytics Accelerator for z/OS V4.1.0: Guides and Manuals - United States

  http://www-01.ibm.com/support/docview.wss?uid=swg27040146
- The limit of concurrent queries in IBM DB2 Analytics Accelerator was changed in V4

  https://w3-connections.ibm.com/forums/html/topic?id=cb527e32-84a9-4b03-8b5a-6f51e3359848&ps=25
- IBM DB2 Analytics Accelerator for z/OS V4.1.0: Release Notes

  http://www.ibm.com/support/docview.wss?uid=swg27040452
- Prerequisites and Maintenance for IBM DB2 Analytics Accelerator for z/OS Version 4.1

  http://www.ibm.com/support/docview.wss?uid=swg27039487
- How to store the query history in a DB2 for z/OS table

  http://www-01.ibm.com/support/docview.wss?uid=swg27039739
- Introduction to DB2 tools

  http://www.ibm.com/support/knowledgecenter/zosbasics/com.ibm.imz.doc.dmtools/tools_db2intro.html?lang=en
- IBM Query Monitor

  http://www-03.ibm.com/software/products/de/db2-query-monitor-for-zos/
- Tivoli OMEGAMON XE for DB2 Performance Monitor on z/OS

  http://www-03.ibm.com/software/products/en/tivoliomegamonxefordb2performancemonitoronzos/
- DB2 Administration Tool for z/OS 11.1.0 and Accelerator

  http://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/topic/com.ibm.db2tools.adb111.doc.ug/topics/adbu_idaa_using.htm
- InfoSphere Optim Query Workload Tuner, Version 4.1.0.1

  http://www-03.ibm.com/software/products/en/optiquerworktunefordb2forzos

  http://pic.dhe.ibm.com/infocenter/dstudio/v4r1/index.jsp?topic=%2Fcom.ibm.datatools.qrytune.nav.doc%2Ftopics%2Fhelpindex_qt.html
- Setting up a WLM application environment

  https://publib.boulder.ibm.com/infocenter/dzichelp/v2r2/index.jsp?topic=%2Fcom.ibm.datatools.aqt.doc%2Finstallmanual%2Ftask%2Ft_idaa_inst_db2_ptf_sps.html

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Reliability and Performance with IBM DB2 Analytics Accelerator Version

(0.5” spine)
0.475”<->0.873”
250 <-> 459 pages

Reliability and Performance with IBM DB2 Analytics Accelerator Version 4.1

Discover ways to extend your investment in IBM System z for business analytics

Learn how to transparently accelerate complex DB2 queries

Prepare to implement highly available analytics

The IBM DB2 Analytics Accelerator for IBM z/OS is a high-performance appliance which integrates the IBM zEnterprise infrastructure and the IBM Netezza technology. With this integration, you can accelerate data-intensive and complex queries in a DB2 for z/OS highly secure and available environment.

DB2 and the Analytics Accelerator appliance form a self-managing hybrid environment running online transaction processing and online transactional analytical processing concurrently and efficiently. These online transactions run together with business intelligence and online analytic processing workloads.

DB2 Analytics Accelerator V4.1 expands the value of high-performance analytics. DB2 Analytics Accelerator V4.1 opens to static Structured Query Language (SQL) applications and row set processing, minimizes data movement, reduces latency, and improves availability.

This IBM Redbooks publication provides technical decision-makers with an understanding of the benefits of version 4.1 of the Analytics Accelerator with DB2 11 for z/OS. It describes the installation of the new functions, and the advantages to existing analytical processes as measured in our test environment. This book also introduces the DB2 Analytics Accelerator Loader V1.1, a tool that facilitates the data population of the DB2 Analytics Accelerator.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, clients, and IBM Business Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-8213-00 ISBN 0738439878