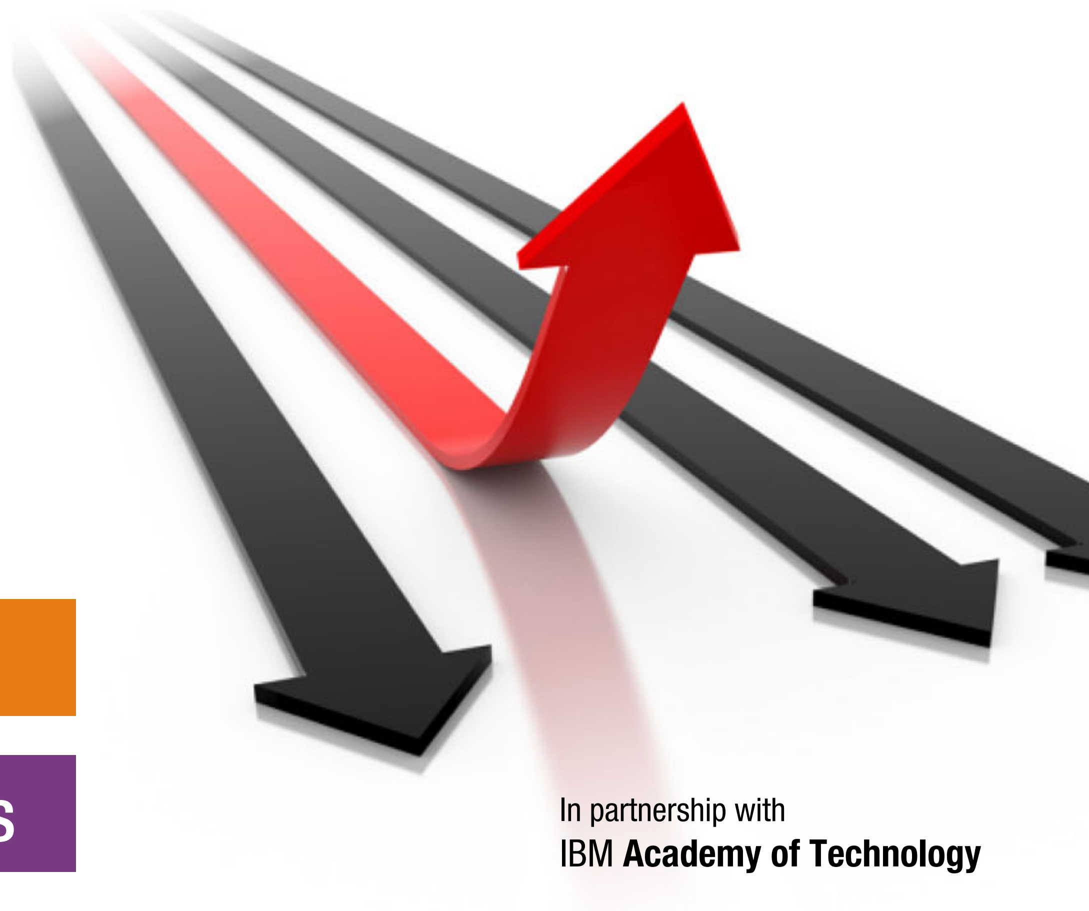
Redbooks
ibm.com/redbooks

Enterprise Data Warehouse Optimization with Hadoop on IBM Power Systems Servers

Helen Lu

Maciej Olejniczak



Analytics

Power Systems

In partnership with
IBM **Academy of Technology**

IBM

Redpaper

International Technical Support Organization

Enterprise Data Warehouse Optimization with Hadoop on IBM Power Systems Servers

January 2018

REDP-5476-00

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

First Edition (January 2018)

This edition applies to Hortonworks Data Platform (HDP) Version 2.6 running on IBM Power Systems servers.

Contents

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on 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, MD-NC119, Armonk, NY 10504-1785, US

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 jurisdictions 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 provide in any way it believes appropriate without incurring any obligation to you.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

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.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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 actual people or business enterprises 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. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

AIX®
BLU Acceleration®
DB2®
GPFS™
IBM®
IBM Elastic Storage™
IBM Spectrum™
IBM Spectrum Conductor™
IBM Spectrum Scale™
IBM Spectrum Storage™
IBM Watson®
Informix®
Micro-Partitioning®
OpenCAPI™
POWER®
Power Architecture®
POWER Hypervisor™
Power Systems™
POWER8®
POWER9™
PowerLinux™
PowerVM®
Redbooks®
Redpaper™
Redbooks (logo) ®
Watson™
WebSphere®

The following terms are trademarks of other companies:

Netezza, and TwinFin are trademarks or registered trademarks of IBM International Group B.V., an IBM Company.

Intel, Intel Xeon, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

Data warehouses were developed for many good reasons, such as providing quick query and reporting for business operations, and business performance. However, over the years, due to the explosion of applications and data volume, many existing data warehouses have become difficult to manage. Extract, Transform, and Load (ETL) processes are taking longer, missing their allocated batch windows. In addition, data types that are required for business analysis have expanded from structured data to unstructured data.

The Apache open source Hadoop platform provides a great alternative for solving these problems.

IBM® has committed to open source since the early years of open Linux. IBM and Hortonworks together are committed to Apache open source software more than any other company.

IBM Power Systems™ servers are built with open technologies and are designed for mission-critical data applications. Power Systems servers use technology from the OpenPOWER Foundation, an open technology infrastructure that uses the IBM POWER® architecture to help meet the evolving needs of big data applications. The combination of Power Systems with Hortonworks Data Platform (HDP) provides users with a highly efficient platform that provides leadership performance for big data workloads such as Hadoop and Spark.

This IBM Redpaper™ publication provides details about Enterprise Data Warehouse (EDW) optimization with Hadoop on Power Systems. Many people know Power Systems from the IBM AIX® platform, but might not be familiar with IBM PowerLinux™, so part of this paper provides a Power Systems overview. A quick introduction to Hadoop is provided for those not familiar with the topic. Details of HDP on Power Reference architecture are included that will help both software architects and infrastructure architects understand the design.

In the optimization chapter, we describe various topics: traditional EDW offload, sizing guidelines, performance tuning, IBM Elastic Storage™ Server (ESS) for data-intensive workload, IBM Big SQL as the common structured query language (SQL) engine for Hadoop platform, and tools that are available on Power Systems that are related to EDW optimization. We also dedicate some pages to the analytics components (IBM Data Science Experience (IBM DSX) and IBM Spectrum™ Conductor for Spark workload) for the Hadoop infrastructure.

Authors

This paper was produced by a team of specialists from around the world working at the International Technical Support Organization, Austin Center.

Helen Lu is a Cognitive Solution Architect in the Advanced Computing Solutions team at IBM Canada. In her 20 years of working in the IT industry, Helen has developed a set of well-rounded skills in IT solution design and implementation. She provides services to worldwide IBM clients across industries such as financial services, insurance, telecommunications, government agencies, and retail. She has over 15 years of experience in the areas of database, data management, data warehousing, and analytics. Her expertise is in providing end-to-end solution design from application, software, to hardware for IBM clients. Helen is also an IBM and GroupOne certified IT Architect.

Maciej Olejniczak is an IBM Certified IT Expert in Actualizing IT Solutions and The Open Group Certified Master IT Specialist. He has been with IBM since 2008. He has more than 15 years of experience in the areas of systems management, networking, big data computing, and security. He is a cross-functional technology consultant and a Global Technical Account Advocate. He achieved a master level in implementing all activities that transform IT from a vision to a working solution, and is an IBM Academy of Technology talented mentor for new hires and IT Specialists. His focus has been on IBM Security products and Platform Computing for the last four years. Maciej is recognized as an IBM Redbooks® Gold Author, and he has co-authored six IBM Redbooks so far.

The project that produced this publication was managed by:
Scott Vetter, PMP

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

Daniel de Souza Casali, Federico Fros, Beth Hoffman, Yves Serge Joseph, Craig Maddux, Steve Roberts, Lilian Romero, Fernando Vermelho, James Wang, Theresa Xu.

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.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our papers to be as helpful as possible. Send us your comments about this paper 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

Enterprise Data Warehouse overview

An Enterprise Data Warehouse (EDW) is a unified data architecture that is used to store data for an organization. It is not a technology or a tool. The purpose of an EDW is to collect data from various data sources, and store data in a unified repository for business analysis. This purpose is different from Online Transactional Processing (OLTP), which provides business transactional services. An EDW provides a clean, stable, consistent, usable, and understandable source of business data that is organized for analysis. Organizations need EDW to provide insights into their business operations, performance, trends, anomalies, clients, and products.

1.1 Traditional Enterprise Data Warehouse

A traditional EDW usually extracts data from the OLTP systems or Operational Data Store (ODS), moves data into a staging area, and then cleans, transforms, and loads data into a data warehouse. Data from OLTP or ODS processes is structured data, such as bank transactions, client information, or product information, and is stored in relational databases. A data warehouse must have a predefined schema before loading data into a data warehouse so the data can be worked with.

It requires significant amount of effort to modeling and the implementation of that model. Therefore, a traditional EDW takes a long time to develop, from developing a schema to developing Extract, Transform, and Load (ETL) jobs for various data sources within an enterprise, and is expensive to build and maintain. In addition, after a schema is defined, it is inflexible and hard to change. In recent years, due to the explosion of data volume, many EDWs have reached their capacity, and are struggling to keep up with business needs.

A traditional EDW flow is shown in Figure 1-1.

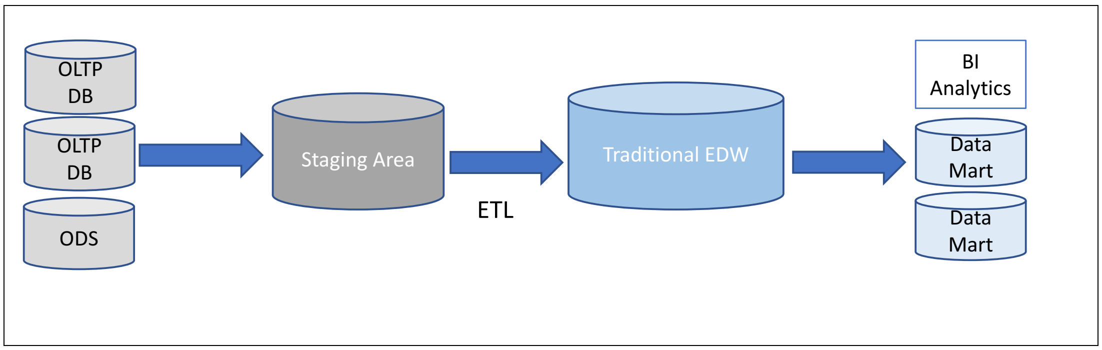


Figure 1-1 Traditional EDW data flow

1.2 Enterprise Data Warehouse on Hadoop

In the past decade, the world has changed tremendously. For example, e-business, online stores, and social media have become more popular and part of many people's daily life. As a result of these changes and technology advancement comes the arrival of big data. *Big data* is data of such volume, velocity, and variety that it cannot be handled by traditional relational database management systems (RDBMSs). In addition to traditional relational structured data, enterprises are dealing with more semi-structured and unstructured data.

Semi-structured and unstructured data have diversified data types and formats, for example, semi-structured data has XML and JSON, and unstructured data has emails, healthcare records, legal contracts, and call center correspondence. There is no convenient definition of records, keys, and attributes or an apparent and obvious context for unstructured data, which makes doing analytical processing more difficult than doing analytical processing for relational and structured data.

Because business operation and business models change more frequently now than ever, EDW must change too. Although traditional EDW still has its place in analytics, two new opportunities exist for building EDW on Hadoop.

The traditional EDW does not scale out well with huge data volume in a cost-effective way. It has a rigid data model and cannot handle unstructured data. The open source Apache Hadoop platform provides an alternative to these growing problems. This is an opportunity to build new EDWs on Hadoop, which complements existing EDW.

The second opportunity is to offload data and processes from existing, traditional EDW to Hadoop, which can reduce costs and make existing data warehouses more efficient. Hadoop provides massive and cheap storage, which solves the data capacity problem for the traditional data warehouse. Hadoop provides a landing zone for raw data, which retains data in its original atomic form, and keeps data lineage and data history. Infrequently used data, also known as *cold data*, from the existing data warehouse can be offloaded into Hadoop, which can free resources from the existing data warehouse and improve its performance. Additionally, migrating ETL processes from the existing data warehouse to Hadoop can result in reduced costs and faster processing times.

This paper describes EDW optimization with Hadoop that is supported by Hortonworks Data Platform (HDP) running on Power Systems servers.

IBM Power Systems servers are perfect choices for the Hadoop platform because they provide high performance, greater query throughput, and higher price performance than x86 servers. They also provide superior flexibility to scale up or out to meet increasing workload requirements. IBM Elastic Storage Server (ESS) reduces Hadoop storage infrastructure by 60 percent. In addition, using PowerAI next to the Hadoop infrastructure enables deep learning, machine learning, and AI that is more accessible and useful to an enterprise.

1.3 Hadoop technology overview

This section provides an overview of the Hadoop technology as a major element in storing big data.

Hadoop is an open source software framework for storing data and running applications on clusters of commodity hardware. It provides massive storage for any kind of data, enormous processing power, and the ability to handle virtually limitless concurrent tasks or jobs.

From a technology perspective, Hadoop brings the power of big data to the enterprise by making large-scale computing sufficiently flexible and affordable so that any size organization can use this technology.

The widespread adoption of Hadoop to solve the challenge of big data has started a growth of innovations and supporting technologies for manipulating, managing, and extracting value from big data. Hadoop has become a preferred real-time platform because of its low cost (as compared to most commercial real-time platforms) and its massive storage capabilities.

Normally, an EDW contains data 5 - 10 years and data is added into the warehouse over time in the form of snapshots. A Hadoop data warehouse architecture enables deeper analytics and allows advanced reporting from these diverse sets of data.

Hadoop for the enterprise is driven by several rising needs. On a technology level, many organizations need data platforms to scale up to handle exploding data volumes. They also need a scalable extension for existing IT systems in warehousing, archiving, and content management. Others must get BI value out of non-structured data. Hadoop fits the bill for all these needs.

On a business level, everyone wants to get business value and other organizational advantages out of big data instead of merely managing it as a cost center. Analytics has arisen as the primary path to business value from big data, and that is why the two come together in the term *big data analytics*

1.3.1 Advantages of using Hadoop

Hadoop is not just a storage platform for big data. It is also a computational platform for business analytics. This makes Hadoop ideal for firms that want to compete in analytics, and retain customers, grow accounts, and improve operational excellence by using analytics. Hadoop provides the following advantages:

- Ability to store and process huge amounts of any kind of data.

  With data volumes and varieties constantly increasing, especially from social media and the Internet of Things (IoT), this ability is a key consideration.

- Computing power.

  Hadoop's distributed computing model processes big data fast. The more computing nodes that you use, the more processing power you have.

- Fault tolerance.

  Data and application processing are protected against hardware failure. If a node goes down, jobs are automatically redirected to other nodes to make sure that the distributed computing does not fail. Multiple copies of all data are stored automatically.

- Flexibility and life span.

  Unlike traditional relational databases, you do not have to preprocess data before storing it. You can store as much data as you want and decide how to use it later. A common use case is to extend a data warehouse environment by integrating Hadoop into the environment. Hadoop extends systems for content management, records management, content archives, and data archives. In these configurations, Hadoop does not replace existing systems. Instead, it offloads those systems and provides economical components so that each system has greater capacity and a longer life span at a lower cost, which in turn means more value for the enterprise over the long haul.

- Low cost.

  The open source framework is a no-charge product that uses commodity hardware to store large quantities of data. The high-end relational databases are expensive in configurations that are large enough to deal with big data. Data warehouse appliances are almost as expensive.

- Scalability.

  Organizations need scalability for all data. They can easily grow their system to handle more data simply by adding nodes. Hadoop has established itself as an enterprise-scope data management platform for multiple data types and domains, and new preferred practices are also established for them, as seen in Hadoop-based data lakes and enterprise data hubs.

- New and exotic data types.

  Hadoop helps organizations obtain business value from data that is new to them or previously unmanageable because Hadoop supports widely diverse data and file types. Hadoop is adept with schema-free data staging and machine data from robots, sensors, meters, and other devices.

- Key use cases demand better structured query language (SQL) on Hadoop.

  As data management professionals have gone deeper into Hadoop usage, many have determined that important use cases require so-called “SQL on Hadoop”, that is, where Hadoop natively supports the execution of ANSI-standard SQL. Desirable use cases include SQL-based analytics and the management of tabular data. One of the hotter trends is to push ETL’s transformation processing down into Hadoop Distributed File System (HDFS), following an ELT model that is often used with relational databases. Transformation logic varies considerably, but much of it is inherently relational because it involves complex table joins and SQL.

Hadoop use is well-established for advanced analytics, data visualization, and data warehousing; newer use cases are arising for data integration, data archiving, and content management. Hadoop is a mature tool for broad, enterprise-scale use across multiple departments and use cases.

1.3.2 Apache Hadoop components

Apache Hadoop helps with the following functions:

- Collection of nearly limitless amounts of data
- Loading data without a schema
- Exploration
- Cleansing
- Organization
- Making the data available for analysis

There are two main components of Apache Hadoop: HDFS and Yet Another Resource Negotiator (YARN).

- HDFS is a distributed file system that provides high-throughput access to application data. HDFS is designed to run on commodity hardware. It has many similarities with existing distributed file systems. However, the differences from other distributed file systems are significant. HDFS is highly fault-tolerant and is deployed on low-cost hardware. HDFS provides high throughput access to application data and is suitable for applications that have large data sets.

  HDFS was originally built as an infrastructure for the Apache Nutch web search engine project. HDFS is part of the Apache Hadoop Core project.

- YARN is a framework for job scheduling and cluster resource management. YARN enables a range of access methods to coexist in the same cluster against shared data sets. This feature avoids unnecessary and costly data silos. HDP enables multiple data processing engines that range from interactive structured query language (SQL) and real-time streaming to data science and batch processing to use data that is stored in a single platform.

Apache Hadoop related projects

Here is a brief description of some popular Hadoop-related projects that are available from Apache:

Ambari
: A web-based tool for provisioning, managing, and monitoring Apache Hadoop clusters.

  Ambari also provides a dashboard for viewing cluster health, such as heatmaps, and the ability to view MapReduce, Pig, and Hive applications visually along with features to diagnose their performance characteristics.

  Ambari supports many services such as:

  HDFS
  : Hadoop Distributed File System.

  MapReduce
  : A YARN-based system for parallel processing of large data sets. MapReduce, the programming paradigm that enables this massive scalability, is the heart of Hadoop. The term MapReduce refers to two separate and distinct tasks that Hadoop programs perform.

  Hive
  : A data warehouse software project that is built on top of Apache Hadoop that provides data summarization, query, and analysis. Hive provides an SQL-like interface to query data that is stored in various databases and file systems that integrate with Hadoop.

  HCatalog
  : Storage Management Layer for Hadoop that helps users of different data processing tools in the Hadoop infrastructure such as Hive, Pig, and MapReduce to easily read and write data from the cluster.

  HBase
  : Hadoop non-relational database, which is a distributed and scalable big data store. Use HBase when you need random, real-time read/write access to your big data.

  ZooKeeper
  : A centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services. These services are used in some form or another by distributed applications.

  Oozie
  : Workflow scheduler system to manage Apache Hadoop jobs. Oozie is integrated with the rest of the Hadoop stack, supporting several types of Hadoop jobs and system-specific jobs (such as Java programs and shell scripts).

  Pig
  : A platform for analyzing large data sets that consists of a high-level language for expressing data analysis programs, which are coupled with infrastructure for evaluating these programs. The salient property of Pig programs is that their structure is amenable to substantial parallelization, which in turns enables them to handle large data sets.

  Sqoop
  : A tool that is designed for efficiently transferring bulk data between Apache Hadoop and structured data stores, such as relational databases.

Cassandra
: A scalable multi-master database with no single points of failure. Linear scalability and proven fault-tolerance on commodity hardware or cloud infrastructure make it the perfect platform for mission-critical data. Cassandra’s support for replicating across multiple data centers is best-in-class, providing lower latency for your users and the peace of mind of knowing that you can survive regional outages. For more information, see the Apache Cassandra website.

Hive	A data warehouse infrastructure that provides data summarization and ad hoc querying. For more information, see the Apache Hive website.
Pig	A high-level data flow language and execution framework for parallel computation. For more information, see the Apache Pig website.
Spark	A fast and general compute engine for Hadoop data. Spark provides a simple and expressive programming model that supports a wide range of applications, including ETL, machine learning, stream processing, and graph computation. For more information, see the Apache Spark website.

Note: For more information about all Apache Hadoop projects, see the Apache Hadoop website.

1.3.3 IBM and Hadoop technology

IBM continuously integrates open source Apache Hadoop with enterprise functions to deliver large-scale analysis with built-in resiliency and fault tolerance. IBM solutions support structured, semi-structured, and unstructured data in its native format for maximum flexibility. It adds features for administration, discovery, development, provisioning, security, and support, along with best-in-class analytical capabilities. The result is a solution for complex, enterprise-scale projects based on Hadoop.

Here are some key big data use cases for Hadoop:

- Data Science Sandbox

  Data Science is an interdisciplinary field that combines machine learning, statistics, advanced analysis, and programming. It is a new form of art that draws out hidden insights and puts data to work in the cognitive era.

- Data Lake Analytics

  A data lake is a shared data environment that is composed of multiple repositories and capitalizes on big data technologies. It provides data to an organization for various analytics processes.

- Streaming Data / IoT Platform

  Stream computing enables organizations to process data streams that are always on and never ceasing. Stream computing helps organizations spot opportunities and risks across all data.

Note: You can download Hadoop trials from Apache Hadoop: Built for big data, insights, and innovation.

1.3.4 The Hortonworks Data Platform on IBM Power Systems

On September 2016, IBM and Hortonworks announced Power Systems support for HDP. Then, in June 2017, IBM and Hortonworks announced an expansion to their relationship that is focused on extending data science and machine learning to more developers and across the Apache Hadoop infrastructure. The companies are combining HDP with IBM Data Science Experience (IBM DSX)and IBM Big SQL into new integrated solutions to help everyone from data scientists to business leaders better analyze and manage their mounting data volumes, and accelerate data-driven decision-making.

Note: For more information, see Hortonworks.

HDP on IBM Power Systems delivers a superior solution for the connected enterprise data platform. With industry-leading performance and IT efficiency that is combined with the best of open technology innovation to accelerate big data analytics and artificial intelligence, organizations can unlock and scale data-driven insights for the business like never before.

An industry-leading, secure, and enterprise-ready open source Apache Hadoop distribution, HDP addresses a range of data-at-rest use cases, powering real-time customer applications and delivering robust analytics to accelerate decision-making and innovation.

HDP uses the HDFS for scalable, fault-tolerant big data storage, and Hadoop's centralized YARN architecture for resource and workload management. YARN enables a range of data processing engines, including SQL, real-time streaming, and batch processing, among others, to interact simultaneously with shared data sets, avoiding unnecessary and costly data silos and unlocking an entirely new approach to analytics.

This HDP solution is illustrated in Figure 1-2.

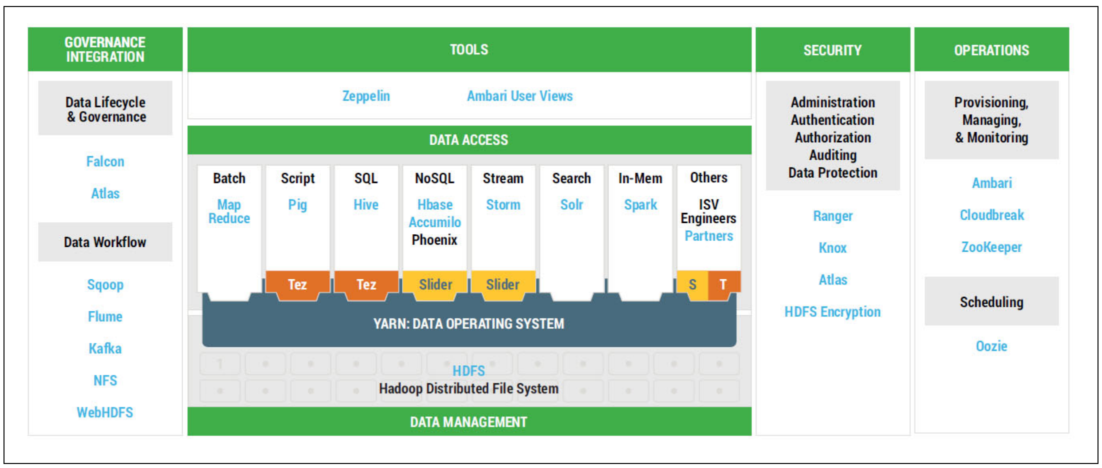


Figure 1-2 Hortonworks Data Platform solution

HDP helps enterprises transform their businesses by unlocking the full potential of big data with the benefits that are listed in Table 1-1 on page 9.

Table 1-1 The Hortonworks difference

Open	Central	Interoperable	Enterprise-ready
HDP is composed of numerous Apache Software Foundation (ASF) projects that enable enterprises to deploy, integrate, and work with unprecedented volumes of structured and unstructured data. The ASF approach is to deliver enterprise-grade software that fosters innovation and prevents vendor lock-in.	YARN is the architectural center of open-enterprise Hadoop. It allocates resources among diverse applications that process data. YARN coordinates clusterwide services for operations, data governance, and security. YARN also maximizes data ingestion by enabling enterprises to analyze data to support diverse use cases. This process empowers Hadoop operators to confidently extend their big data assets to the largest possible audience in their organizations.	Its 100 percent open source architecture enables HDP to be interoperable with a broad range of data center and business intelligence applications. HDP's interoperability helps minimize the expense and effort that is required to connect customers' IT infrastructures with HDP's data and processing capabilities. With HDP, customers can preserve their investment in existing IT architecture as they adopt Hadoop.	HDP is built for enterprises. Open-enterprise Hadoop provides consistent operations, with centralized management and monitoring of clusters through a single GUI. With HDP, security and governance are built into the platform. This feature helps provide a security-rich environment that is consistently administered across data access engines.

Hortonworks Data Platform on Power Systems performance

Power Systems with IBM POWER8® processors and differentiated hardware acceleration technology are designed to deliver breakthrough performance for big data analytics workloads. The POWER8 processor delivers industry-leading performance for big data analytics applications running on HDP, with multi-threading designed for fast execution of analytics (eight threads per core), multi-level cache for continuous data load and fast response (including an L4 cache), and a large, high-bandwidth memory workspace to maximize throughput for data-intensive applications.

IBM Power Systems OpenPOWER LC server family

Designed for flexibility and seamless integration to existing clusters and clouds, the new IBM OpenPOWER LC server family offers the data-crushing POWER8 processor in a range of purpose-built system configurations, from compute-dense to storage-rich. The LC family's innovative design in partnership with the OpenPOWER Foundation offers hardware accelerator-offload for compute, storage, and networking workloads for incredible speed-ups to analytics and massive efficiencies in data movement.

OpenPOWER brings the leading processor together with the best of IBM Business Partners and users across the infrastructure, from high-performance computing (HPC) installations, enterprise IT, and hyperscale data centers and to system designers worldwide.

The HDP on IBM Power Systems reference configuration suggests options for a solution that is sized to your specific needs, which are built with a combination of the high-performance, storage-rich Power S822LC for Big Data, and the lightweight, yet compute-powerful Power S821LC.

Superior performance for Apache Hadoop and Spark workloads

HDP on Power Systems delivers more data faster, enabling valuable analytics for better and quicker decision-making. In IBM performance testing for typical Apache Hadoop workloads, HDP on Power Systems versus x86-based solutions demonstrated 70 percent more queries per hour based on an average response time and 40 percent reduction on average in query response time.

In addition to enabling faster Hadoop workloads, the POWER8 processor's leading thread density, large cache and memory bandwidth, and superior I/O capabilities are a great match for in-memory Apache Spark workloads, including SQL, streaming, graph, and machine learning analytics.

References

For more information, see the following references:

- Hortonworks Data Platform: Command Line Installation
- Step by Step: Installation of HDP 2.6 on IBM Power Systems

1.3.5 Hortonworks DataFlow

HDP is the industry's only true secure, enterprise-ready open source Apache Hadoop distribution based on a centralized architecture (YARN). HDP addresses the complete needs of data-at-rest, powers real-time customer applications and delivers robust big data analytics that accelerate decision making and innovation.

Hortonworks DataFlow (HDF) for IBM, powered by Apache NiFi, is the first integrated platform that solves the challenges of collecting and transporting data from a multitude of sources. HDF for IBM enables simple, fast data acquisition, secure data transport, prioritized data flow, and clear traceability of data from the edge of your network to the core data center. It uses a combination of an intuitive visual interface, a high-fidelity access and authorization mechanism, and an always-on chain of custody (data provenance) framework.

2

IBM Power Systems overview

Achieving significant, measurable business value and insight from big data and analytics technologies requires organizations to put an infrastructure foundation in place that supports the rapidly growing volume, variety, and velocity of data.

IBM offers a broad set of analytics capabilities that are built on the proven foundation of a single platform, IBM Power Systems. Power Systems is an open, secure, and flexible platform that is designed for big data. It has massive I/O bandwidth to deliver analytics in real time, and it can provide the necessary capabilities to handle the varying analytics initiatives of each business.

2.1 IBM Power Systems overview

Today's business leaders demand servers for big data that is optimized, secure, and adapt to changing workload requirements. Power Systems servers are built with open technologies and designed for mission-critical data applications. Power Systems servers use technology from the OpenPOWER Foundation, an open technology infrastructure that uses the IBM POWER architecture to help meet the evolving needs of big data applications. The combination of Power Systems servers with Hortonworks Data Platform (HDP) provides users with a highly efficient platform that provides leadership performance for big data workloads, such as Hadoop and Spark.

2.1.1 POWER8 server highlights

In August 2013, IBM announced the POWER8 architecture, which added major performance enhancements. The POWER8 chip has 12 cores and 96 MB of eDRAM L3 cache. The chip can use up to 128 MB of off-chip eDRAM L4 cache. On-chip memory controllers can handle 1 TB of RAM and 230 GBps of sustained memory bandwidth. The on-board PCIe controllers can handle 48 GBps of I/O to other parts of the system.

The IBM Power System S822LC for Big Data servers delivers the high data throughput that is required to handle big data workloads.

The Power S822LC for Big Data (8001-22C) server is a storage-rich server that delivers superior performance and throughput for high-value Linux workloads, such as big data, MySQL, NoSQL, PHP, MongoDB, and key open source workloads. It is the server of choice for clients who want the advantages of running their big data, Java, open source, and industry applications on a platform that is optimized for cloud, data, and Linux. Additionally, the server is simple to order and can scale from single racks to large clusters with a simple deployment.

The Power S822LC for Big Data server supports up to two processor sockets, offering 16-core 3.32 GHz, 20-core 2.92 GHz, or 22-core 2.89 GHz POWER8 configurations, and up to 512 GB DDR4 memory in a 19-inch rack-mount, 2U (EIA units) drawer configuration. All the cores and memory are activated.

The frequency for both processor options can also be boosted.

Hardware advantages

Here are the hardware advantages of this solution:

- A consolidated server footprint with up to 66% more virtual machines (VMs) per server than a competitive x86.
- Superior application performance with performances that provide up to a 2x per core performance advantage over x86-based systems.
- Leadership data throughput that is enabled by POWER8 multithreading with up to 4x more threads than x86 designs.

Superior application performance

This solution has the following superior application performance attributes:

- Up to 2x better price-performance on open source databases.
- MongoDB on a Power S822LC for Big Data server provides 2x better price-performance than Intel Xeon E5-2690 v4 Broadwell.

- EnterpriseDB 9.5 on a Power S822LC for Big Data server delivers 1.66x more performance per core and 1.62x better price-performance than Intel Xeon E5-2690 v4 Broadwell.
- 40% more operations per second in the same rack space as Intel Xeon E5-2690 v4 systems.
- Acceleration of big data workloads with graphical processing units (GPUs) and superior I/O bandwidth with the Coherent Accelerator Processor Interface (CAPI).

Note: The source for the points in “Superior application performance” on page 12 can be found at IBM Power System S822LC for Big Data.

POWER8 processor advantages

The POWER8 main features that can augment the performance of the POWER8 processor are the following ones:

- POWER8 processor-based systems use memory buffer chips to interface between the POWER8 processor and DDR4 memory. Each buffer chip also includes an L4 cache to reduce the latency of local memory accesses.
- POWER8 processor advancements in multi-core and multi-thread scaling are remarkable. A significant performance opportunity comes from parallelizing workloads to enable the full potential of the microprocessor and the large memory bandwidth. Application scaling is influenced by both multi-core and multi-thread technology.

  Simultaneous multithreading (SMT) enables a single physical processor core to dispatch simultaneously instructions from more than one hardware thread context. With SMT, each POWER8 core can present eight hardware threads. Because there are multiple hardware threads per physical processor core, more instructions can run concurrently.

  SMT is primarily beneficial in commercial environments where the speed of an individual transaction is not as critical as the number of transactions that are performed. SMT typically increases the throughput of workloads with large or frequently changing working sets, such as database servers and web servers.
- Support for DDR4 memory through memory buffer chips that offload the memory support from the POWER8 memory controller.
- An L4 cache within the memory buffer chip that reduces the memory latency for local access to memory behind the buffer chip. The operation of the L4 cache is not apparent to applications running on the POWER8 processor. Up to 128 MB of L4 cache can be available for each POWER8 processor.
- Hardware transactional memory.
- On-chip accelerators, including on-chip encryption, compression, and random number generation accelerators.
- CAPI, which enables accelerators that are plugged into a PCIe slot to access the processor bus by using a low-latency, high-speed protocol interface. For more information, see 2.5, “Coherent Accelerator Processor Interface and OpenCAPI” on page 21.
- Adaptive power management.

2.1.2 POWER9 server highlights

Based on IBM Power Architecture®, IBM POWER9™ processor-based systems target technical computing segments by providing superior floating-point performance and off-chip floating-point acceleration. POWER9 processor-based systems, which consist of superscalar multiprocessors that are massively multithreaded, support cloud operating environments. With the CAPI attached, POWER9 processor-based systems offer a robust platform for analytics and big data applications.

The IBM Power System AC922 server is the next generation of the IBM POWER processor-based systems, which are designed for deep learning and artificial intelligence, high-performance analytics, and high-performance computing (HPC).

The system is a co-designed with OpenPOWER Foundation infrastructure members, and it will be deployed as the most powerful supercomputer on the planet with a partnership between IBM, NVIDIA, Mellanox, and others. It provides the latest technologies that are available for HPC, improving even more the movement of data from memory to GPU accelerator cards and back, enabling faster and lower latency and faster data processing.

POWER9 processor-based system highlights

Among the new technologies the system provides, here are the most significant:

- Increased CPU fabric bandwidth (both X-Bus and A-Bus): Two POWER9 processors with up to 40 cores and improved buses.
- Increased memory capacities over POWER8 (2x and 4x): 1 TB of DDR4 memory with improved speed.
- Using IS DIMMs to provide more competitive offerings in 2-socket and 4-socket spaces.
- Increased I/O bandwidth with PCIe GEN4 slots and future PCIe GEN4 Expansion Drawer.
- BlueLink ports for High Speed GPU / IBM OpenCAPI™ acceleration.
- Integrated NVMe Flash support.
- Up to six NVIDIA Tesla V100 (Volta) GPUs, delivering up to 100 TFlops each, which is a 5x improvement over the previous generation.
- External DVD slot.

Figure 2-1 illustrates the POWER9 processor-based system architecture.

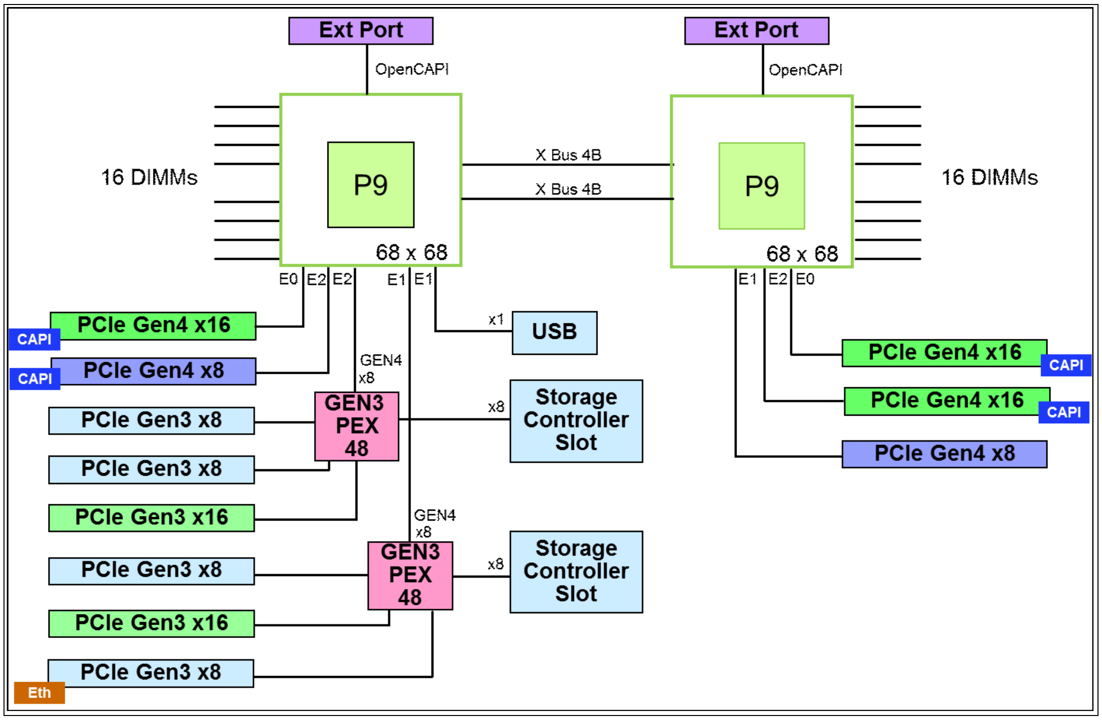


Figure 2-1 POWER9 processor-based system highlights

Key server features

The Power AC922 system addresses the demanding needs of deep learning and AI, high-performance analytics, and HPC.

An updated list of ported HPC applications that can use the IBM POWER technology is found in IBM Power Systems HPC Applications Summary.

The system includes several features to improve performance:

- POWER9 processors:
  - Each POWER9 processor module has either a 16 or 20 cores based 64-bit architecture.
  - Clock speeds for 16-core chip of 2.6 GHz (3.09 GHz turbo).
  - Clock speeds for 20-core chip of 2.0 GHz (2.87 GHz turbo).
  - 512 KB of L2 cache per core, and up to 120 MB of L3 cache per chip.
  - Up to four threads per core.
  - 120 GBps memory bandwidth per chip.
  - 64 GBps SMP interconnect between POWER9 chips.
- DDR4 Memory:
  - Sixteen DIMM memory slots.
  - Maximum of 1024 GB DDR4 system memory.
  - Improved clocking 1333 - 2666 MHz for reduced latency.

- NVIDIA Teslas V100 GPUs:
  - Up to six NVIDIA Tesla V100 GPUs, based on the NVIDIA SXM2 form factor connectors.
  - 7.8 TFLOPs per GPU for double precision.
  - 15.7 TFLOPs per GPU for single precision.
  - 125 TFLOPs per GPU for deep learning. New 640 Tensor Cores per GPU, designed for deep learning.
  - 16 GB HBM2 internal memory with 900 GBps bandwidth, with 1.5x the bandwidth compared to Pascal P100.
  - Liquid cooling for six GPUs configurations to improve compute density.
- NVLink 2.0:
  - Twice the throughput, compared to the previous generation of NVLink.
  - Up to 200 GBps of bidirectional bandwidth between GPUs.
  - Up to 300 GBps of bidirectional bandwidth per POWER9 chip and GPUs, compared to 32 GBps of traditional PCIe Gen3.
- OpenCAPI 3.0:
  - Open protocol bus to allow for connections between the processor system bus in a high speed and cache coherent manner with OpenCAPI compatible devices, such as accelerators, network controllers, storage controllers, and advanced memory technologies.
  - Up to 100 GBps of bidirectional bandwidth between CPUs and OpenCAPI devices.
- PCIe Gen4 Slots: Four PCIe Gen4 slots up to 64 GBps bandwidth per slot, twice the throughput from PCIe Gen3. Three CAPI 2.0 capable slots.

IBM for OpenPOWER

The OpenPOWER Foundation is an open development alliance that is rethinking the infrastructure for hyperscale and cloud data centers. As a founding member, IBM is sharing its industry-leading POWER technology.

To learn more about IBM OpenPOWER, see IBM Portal for OpenPOWER.

For more information about the OpenPOWER Foundation, its mission, and how to join, see the OpenPOWER Foundation.

2.2 POWER versus Intel x86 performance

Power Systems servers are designed for big data, with key architectural features that enhance its performance for data-intensive workloads. Power Systems servers operate at industry-leading levels of efficiency, ensuring that the system performs as warranted while at a sustained 65% utilization, which is a rate higher than common competitive platform utilization levels.

- Four times the threads of x86 processors: A POWER CPU has up to eight threads, but Intel x86 has only two threads. POWER9 technology-based processors are available in two versions, 4-way SMT and 8-way SMT to better match different workloads.
- Although x86 CPU performance has improved in each generation, it does so by increasing the number of cores in the CPU, but the per core performance decreases.

- Three times the memory bandwidth of x86 processors.
- Six times the cache of x86 processors.
- Multiple page size support feature:

  The Power Architecture supports multiple virtual memory page sizes, which in turn provide performance benefits to an application because of hardware efficiencies that are associated with larger page sizes. Large pages provide several technical advantages, such as the following examples:

  - Reduced page faults and Translation Lookaside Buffer (TLB) misses

    A single large page that is being constantly referenced remains in memory, eliminating the possibility of swapping out several small pages.

  - Unhindered data prefetching

    A large page enables unhindered data prefetch, which is constrained by page boundaries.

  - Increased TLB Reach

    This feature saves space in the TLB by holding one translation entry instead of *n* entries, which increases the amount of memory that can be accessed by an application without incurring hardware translation delays.

  - Increased Effective to Real Address Translation (ERAT) Reach

    ERAT on IBM POWER is a first-level and fully associative translation cache that can go directly from effective to real address. Effective addresses are the addresses that are used by the software, and real addresses refer to the physical memory that is assigned to the software by the system. Both the ERAT and the TLB are involved in translating addresses. Large pages also improve the efficiency and coverage of this translation cache.

- Multi-core and multi-thread features, and affinity performance effects - SMT

  The release of IBM POWER8 processors introduced up to eight SMT threads per processor core (logical or physical). POWER9 enhances the flexibility by recognizing some workloads perform better by using fewer threads and have scale-out versions with four SMT threads per core.

  The POWER8 chip is available in configurations with up to 12 cores per chip, and POWER9 is available with up to 24 cores per chip. Each POWER9 processor core supports running in single-threaded mode with one hardware thread, an SMT2 mode with two hardware threads, an SMT4 mode with four hardware threads, or an SMT8 mode with eight hardware threads (depending on variation).

  Each SMT hardware thread is represented as a logical processor in AIX, IBM i, or Linux. When the hardware runs in SMT8 mode, the operating system has eight logical processors for each dedicated POWER8 processor core that is assigned to the partition. To gain the full benefit from the throughput improvement of SMT, applications must use all of the SMT threads of the processor cores.

- Each POWER8 chip has memory controllers that enable direct access to a portion of the memory DIMMs in the system. Any processor core on any chip in the system can access the memory of the entire system, but it takes longer for an application thread to access the memory that is attached to a remote chip than to access data in the local memory DIMMs.

  POWER9 chips extend this capability by offering a choice of buffered access to memory better suited for scale-up and direct memory access that is suited for scale-out servers.

- An efficient hypervisor that is built into the hardware and firmware

  The IBM POWER Hypervisor™ manages the virtualization of processor cores and memory for the operating system. It also ensures that the affinity between the processor cores and memory that a logical partition (LPAR) is using is maintained as much as possible. However, system administrators must also consider affinity issues. Another key aspect of POWER Hypervisor is the impact of application thread and data placement on the cores and the memory that is assigned to the LPAR in which the application is running.

  IBM PowerVM® Hypervisor and the AIX operating system (AIX 7.1 TL3 SP3 and later) on POWER8 implement enhanced affinity in several areas. This feature achieves optimized performance for workloads that are running in a virtualized shared processor LPAR (SPLPAR) environment. These areas can include virtual processors, LPAR page table sizes, and placing LPAR resources to attain higher memory affinity.

- An overall system design that optimizes hardware, software, and storage for fast performance.
- Built-in enterprise mainframe features, such as reliability, security, and elasticity, which are described at IBM, Lenovo Servers Deliver Top Reliability, Cisco UCS, HPE Integrity Gain.

2.2.1 System performance comparison

The following reports provide information about POWER versus x86 technology performance:

- *Infrastructure Matters: POWER8 versus XEON x86 - Clabby Analystic report*

  Clabby Analytics examines the key differences between the two architectures. They find that this new generation of POWER8 processor-based servers are more efficient that x86 based systems, deliver results more quickly, and cost less than x86 alternatives.

- POWER8 versus Intel x86 Performance

  This report shows a comparison of web application performance on POWER8 versus Intel x86 processors. You can view it at Seeing is Believing: POWER8 versus Intel x86 Performance - Short Version or at its YouTube mirror.

- *IBM Confrontation report - POWER versus x86*
- *IBM System Performance Report*

  This document contains performance and benchmark results for IBM servers and workstations running the UNIX (AIX), IBM i, and Linux OS.

- *What Power Can Do That Intel Can't*

2.3 NVIDIA GPU accelerators

Compute-intensive accelerators are GPUs that are developed by NVIDIA. With NVIDIA GPUs, the Power S822LC for Big Data server can offload processor-intensive operations to a GPU accelerator and boost performance.

NVIDIA Tesla GPUs are massively parallel accelerators that are based on the NVIDIA Compute Unified Device Architecture (CUDA) parallel computing platform and programming model. Tesla GPUs are designed for power-efficient, HPC, computational science, supercomputing, big data analytics, and machine learning applications, delivering dramatically higher acceleration than a CPU-only approach.

These NVIDIA Tesla GPU Accelerators are based on the NVIDIA Kepler Architecture, and are designed to run the most demanding scientific models faster and more efficiently.

On systems with x86 CPUs (such as Intel Xeon), the connectivity to the GPU is only through PCI-Express (although the GPUs connect to each other through NVLink). On systems with POWER8 CPUs, the connectivity to the GPU is through NVLink (in addition to the NVLink between GPUs).

NVLink is the NVIDIA high-speed interconnect technology for GPU-accelerated computing. Supported on SXM2-based Tesla P100 accelerator boards, NVLink increases performance for both GPU-to-GPU communications and for GPU access to system memory.

NVLink has three major advantages for application acceleration:

- Performance: The POWER8 with NVLink processor and the Tesla P100 GPU have four NVLink interfaces that support 5x faster communication than PCIe x16 Gen3 connections that are used in other systems, enabling faster data exchange and application performance.
- Programmability: The CUDA 8 software and the Page Migration Engine in Tesla P100 enable a unified memory space with automated data management between the system memory that is connected to the CPU and the GPU memory. Coupled with NVLink, unified memory makes programming GPU accelerators much easier for developers. Applications can be easily accelerated with GPUs by incrementally moving functions from the CPU to the GPU, without having to deal with data management.
- More application acceleration: Because NVLink reduces the communication time between the CPU and GPU, it enables smaller pieces of work to be moved to the GPU for acceleration.

NVLink uses the new NVIDIA High-Speed Signaling (NVHS) interconnect. NVHS transmits data over a differential pair running at up to 20 Gbps. Eight of these differential connections form a *sublink* that sends data in one direction, and two sublinks (one for each direction) form a *link* that connects two processors (GPU-to-GPU or GPU-to-CPU). A single link supports up to 40 GBps of bidirectional bandwidth between the endpoints. Multiple links can be combined to form *gangs* for even higher-bandwidth connectivity between processors. The NVLink implementation in Tesla P100 supports up to four links, enabling a gang with an aggregate maximum theoretical bandwidth of 160 GBps bidirectional bandwidth.

Although NVLink primarily focuses on connecting multiple NVIDIA Tesla P100 GPUs together, it can also connect Tesla P100 GPUs with IBM Power CPUs with NVLink support. Figure 2-2 shows how the CPUs are connected with NVLink in the Power System S822LC for HPC server. In this configuration, each GPU has 180 GBps bidirectional bandwidth to the other connected GPU and 80 GBps bidirectional bandwidth to the connected CPU.

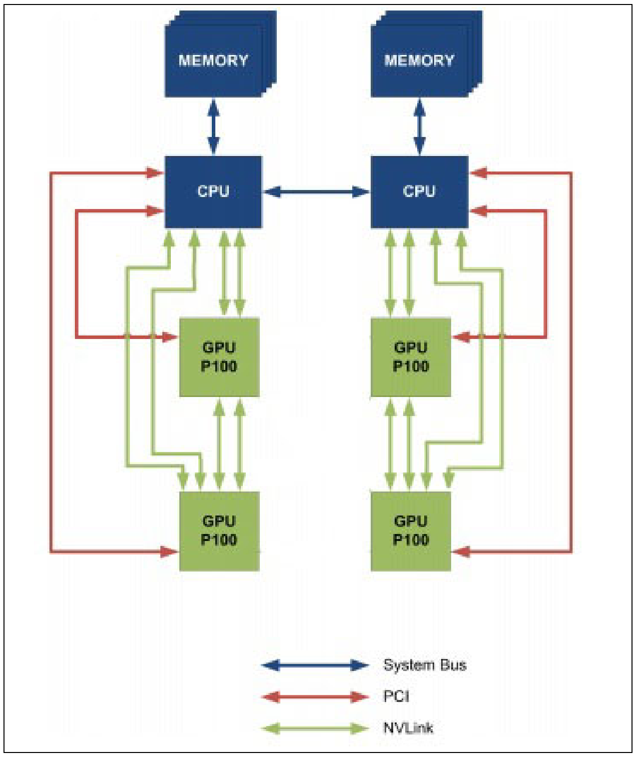


Figure 2-2 NVLink topology in POWER8 for big data

All the initialization of the GPU is through the PCIe interface. The PCIe interface also contain the side-band communication for status, power management, and so on. After the GPU running, all data communication uses the NVLink.

> **Note:** For more information about IBM POWER8 with NVIDIA NVLink Technology systems, see the IBM Portal for OpenPOWER.

2.4 Linux on Power advantages

With IBM Power Systems servers, you can run Linux applications while taking advantage of POWER hardware performance, availability, and reliability features.

All of these Linux distributions provide the tools, kernel support, optimized compilers, and tuned libraries for Power Systems servers to achieve excellent performance. For advanced users, more application and customer-specific tuning approaches are also available.

Additionally, IBM provides many added value packages, tools, and extensions that provide for more tuning, optimizations, and products for the best possible performance on POWER8 processor-based systems. The typical Linux open source performance tools that Linux users are comfortable with are available on Linux on Power Systems.

Under a PowerVM hypervisor, Linux on Power supports small, virtualized IBM Micro-Partitioning® partitions up through large dedicated partitions containing all of the resources of a high-end server. Under a PowerKVM hypervisor, the Linux on Power supports running as a KVM guest on POWER8 processor-based systems.

IBM premier products, such as IBM XL compilers, IBM Java products, IBM WebSphere®, and IBM DB2® database products, all provide Power Systems optimized support with the RHEL, SUSE Linux Enterprise Server, and Ubuntu operating systems.

For more information about IBM Linux on Power, see IBM Knowledge Center.

2.5 Coherent Accelerator Processor Interface and OpenCAPI

Existing system interfaces are insufficient to address the following disruptive forces: A traditional I/O architecture results in a high CPU impact when applications communicate with I/O or accelerator devices at the necessary performance levels. Systems must be able to integrate multiple memory technologies with different access methods and performance attributes.

CAPI eliminates the complexity and impact of an I/O subsystem. CAPI defines a coherent accelerator interface structure for attaching special processing devices to the POWER8 processor bus.

CAPI can attach accelerators that have coherent shared memory access with the processors in the server and share full virtual address translation with these processors, which use a standard PCIe Gen3 bus.

Applications can have customized functions in field-programmable gate arrays (FPGAs) and queue work requests directly in shared memory queues to the FPGA, and by using the same effective addresses (pointers) it uses for any of its threads running on a host processor. From a practical perspective, CAPI enables a specialized hardware accelerator to be seen as an extra processor in the system, with access to the main system memory, and coherent communication with other processors in the system.

The benefits of using CAPI include the ability to access shared memory blocks directly from the accelerator, perform memory transfers directly between the accelerator and processor cache, and reduce the code path length between the adapter and the processors. This is possible because the adapter is not operating as a traditional I/O device, and there is no device driver layer to perform processing. It also presents a simpler programming model.

OpenCAPI

The technology behind OpenCAPI is governed by the OpenCAPI Consortium, which was founded in October 2016 by AMD, Google, IBM, Mellanox and Micron together with NVIDIA, Hewlett Packard Enterprise, Dell EMC, and Xilinx.

OpenCAPI is an open interface architecture that enables any microprocessor to attach to the following items:

- Coherent user-level accelerators and I/O devices
- Advanced memories that are accessible through read/write or user-level DMA semantics
- Vendor-neutral to processor architectures

Figure 2-3 illustrates the OpenCAPI interface.

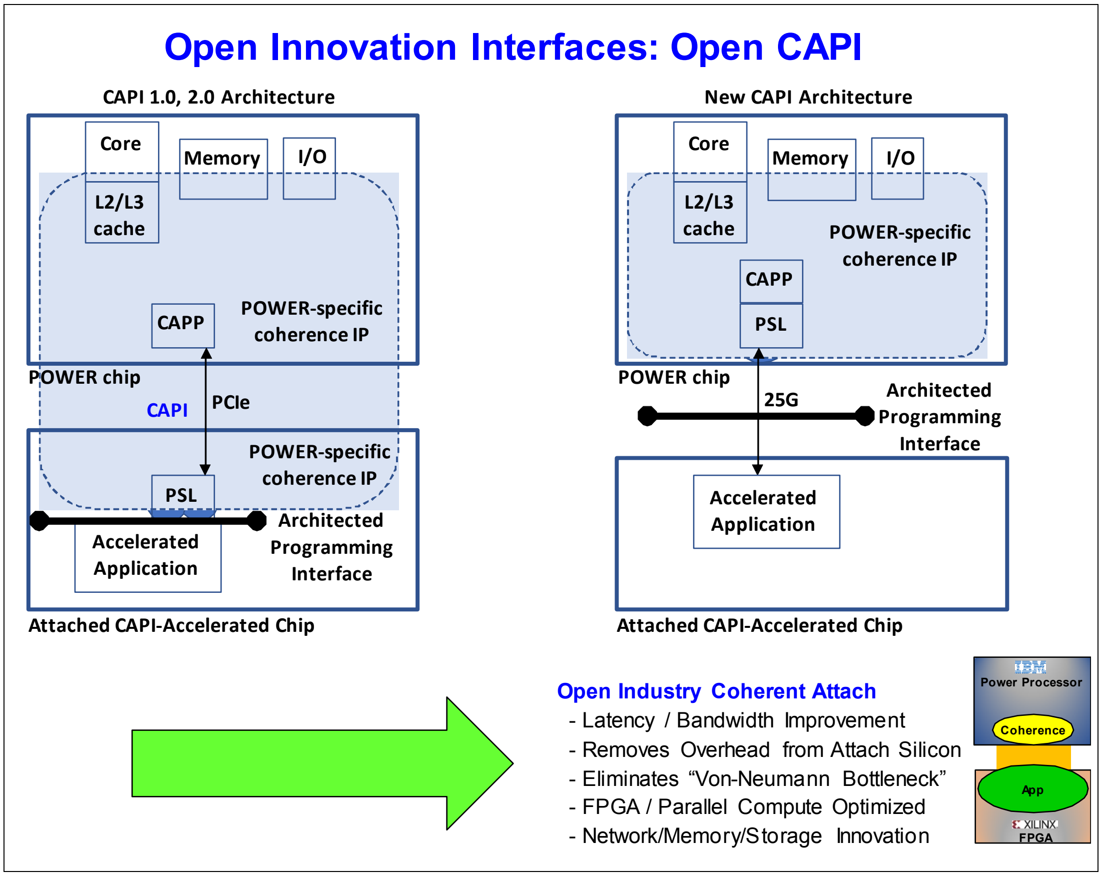


Figure 2-3 OpenCAPI interface

Note: For more information about CAPI for POWER8 Systems, see *CAPI for POWER8 Systems*.

3

Hortonworks Data Platform on IBM Power Systems reference architecture

This chapter describes the Hortonworks Data Platform (HDP) on Power Systems reference architecture. It provides a quick definition for the HDP Hadoop workload types and HDP Hadoop Cluster composition because they are some of the key factors affecting an architecture design. Then, this chapter introduces the HDP on Power Systems reference architecture. The chapter ends with a physical rack design for the reference architecture.

3.1 Hadoop workload categorization

Workload types can have impact on architectural design. Hortonworks categorizes Hadoop workloads into three categories: balanced, performance (or compute-intensive), and storage-dense (or storage-oriented).

Balanced Balanced workloads are distributed equally across the various job types, such as CPU-bound, disk I/O-bound, or network I/O-bound. If workload is unknown in the early stage of the implementation, choose a balanced workload as the starting point.

Performance Performance workloads are CPU-bound and are characterized by the need for many CPUs and much memory to store in-process data. Spark workloads are generally compute-intensive.

Storage-dense Storage-dense workloads such as MapReduce jobs require little compute power. Instead, these jobs rely more on the I/O-bound capacity of the cluster (for example, much cold data from existing data warehouse offloading).

Optimization considerations vary based on workload types. For example, for storage-dense workloads, separation of storage can reduce the number of nodes (central processing complexes (CPCs), and reduce system acquisition cost and cluster maintenance cost.

Another example is Spark workloads. IBM Spectrum Conductor™ with Spark can increase performance and scale, maximize the usage of resources, and eliminate silos of resources that otherwise are tied to separate application implementations. For more information, see Chapter 4, “Enterprise Data Warehouse on Hadoop optimization” on page 31.

3.2 Hadoop cluster node composition

A Hadoop cluster is composed of a set of nodes. There are three types of nodes: master node, worker node, and edge node. The nodes are provisioned with an operating system and configured to have appropriate network connectivity.

Master node A master node is used to host management functions and some storage functions. There are typically one or more master nodes in a cluster, and three master nodes are common to provide basic high-availability (HA) capability.

Worker node A worker node serves two primary roles:

- Each worker node contains some physical storage, which is used for Hadoop Distributed File System (HDFS), and it hosts some storage functions that enable it to manage this storage as part of HDFS. These storage functions communicate and cooperate to form the distributed file system across the collection of worker nodes in the cluster.
- A worker node is also used by the management functions to run applications that are parts of jobs. Job execution is typically distributed across multiple worker nodes to provide parallel execution of the job.

	There are typically three or more (often many more) worker nodes in a cluster. Three worker nodes provide the ability to directly support the common HDFS replication factor of three. Worker nodes are usually the most common node type in a cluster, accounting for perhaps 80 - 90% (or more) of the nodes in the cluster.
Edge node	An edge node serves as the host for functions that require both an “external” and “internal” connection. Edge nodes commonly serve as the pathway or interface between the cluster and external applications. Therefore, they are sometimes referred to as Gateway nodes. Topologies vary, but in many deployments the master and worker nodes have only internal (private) connections to each other, and access to the cluster is controlled and routed through functions running on the edge nodes.

3.3 Hortonworks Data Platform on Power Systems reference architecture

This section describes the HDP on IBM Power Systems reference architecture at four levels: logical architecture, infrastructure view, network topology, and reference design with a POWER8 server.

The first level is the HDP Hadoop logical architecture with the functions and nodes layouts. Figure 3-1 shows a typical HDP cluster with nodes and functions running on each node, including redundancy.

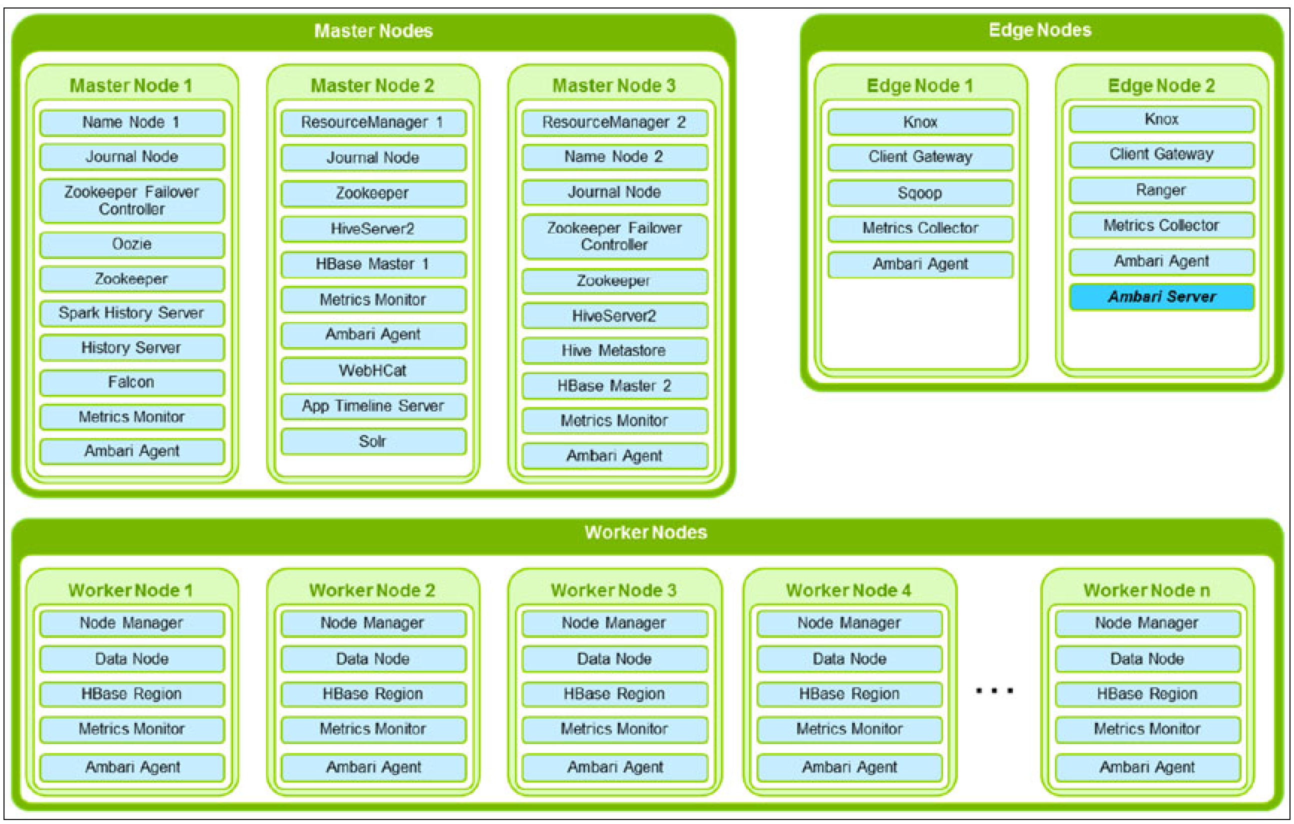


Figure 3-1 Functions and nodes layouts on Hortonworks Data Platform cluster

The second-level architecture view is the infrastructure view, which is shown in Figure 3-2. The infrastructure view provides a detailed view into Hadoop subsystems: edge, management, compute, and storage subsystems, and the components within each system, and the network view of the Hadoop cluster.

For storage-oriented workloads, the storage subsystems can be separated from the compute subsystems (the worker nodes) so that you can increase the storage capacity without adding extra compute nodes. In addition, with IBM Elastic Storage Server (ESS), which uses IBM Spectrum Scale™ RAID with de-clustered erasure coding to distribute and protect data, further reducing the 3x data redundancy that is required for Hadoop separation of storage from compute resources increases the efficiency and cost saving for the cluster.

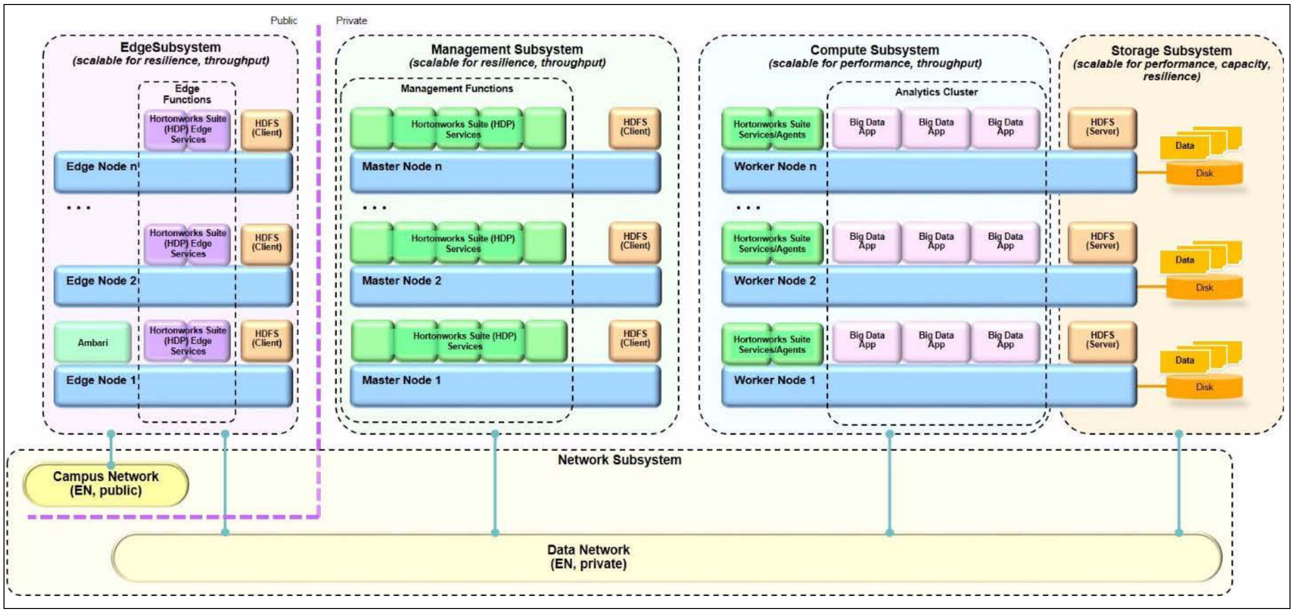


Figure 3-2 Hortonworks Data Platform Hadoop on Power Systems infrastructure view

The third level of the reference architecture, which is shown in Figure 3-3, is the cluster network design. At the platform level, the “Partial-Homed” network pattern is specified for this design, and three infrastructure-level logical networks are included in the network topology as distinct and separate networks. The networks are realized by two pairs of Ethernet switches. One pair is 10 Gb, and it realizes the data network. The other pair is 1 Gb, and it realizes the provisioning and balance of the networks. The switch pairs exist to provide resilience to the network and more bandwidth for the data network.

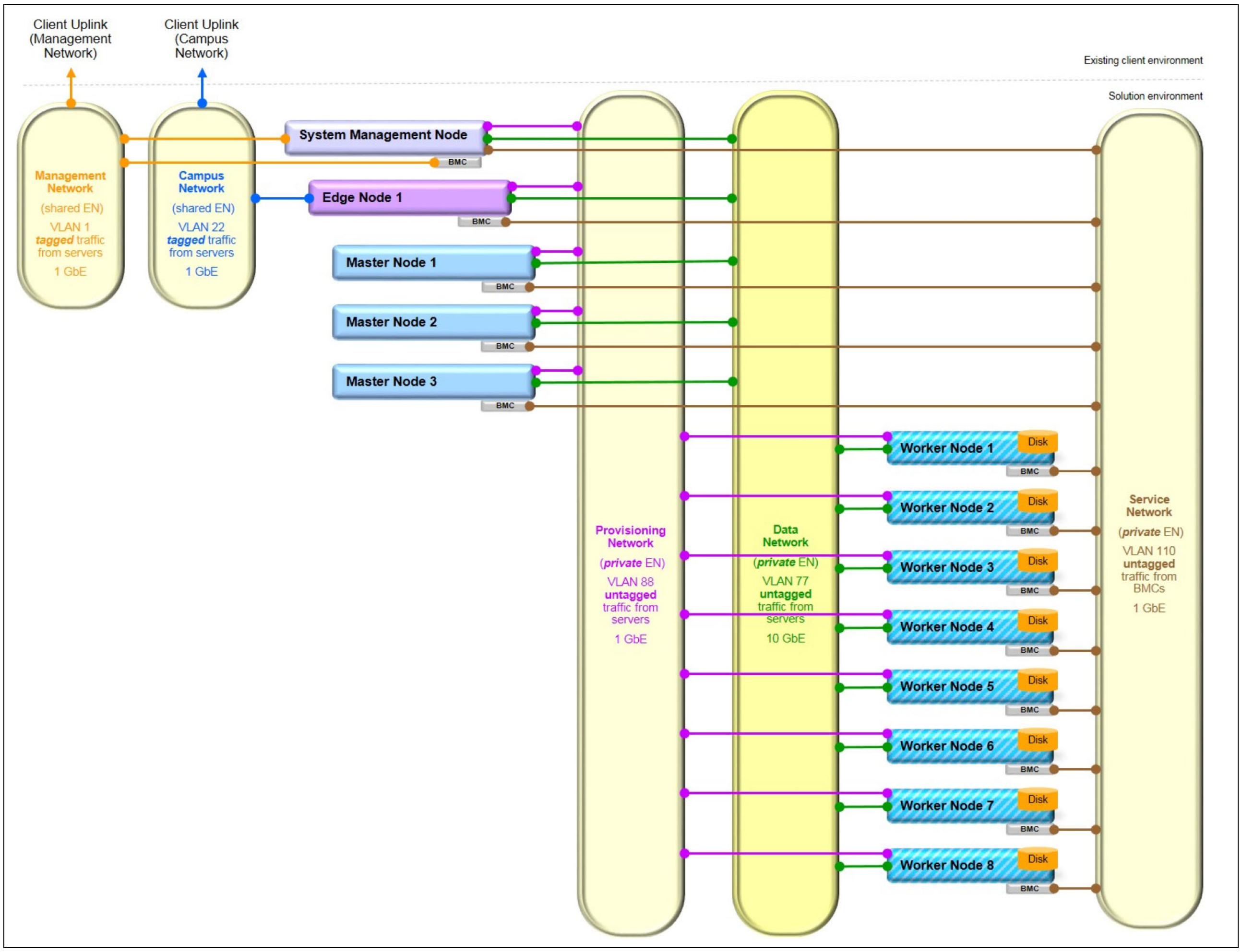


Figure 3-3 Hortonworks Data Platform Hadoop cluster on Power Systems network topology

The fourth level in the reference architecture is the physical design of the cluster. Figure 3-4 provides a sample hardware design for the three types of HDP workload on IBM POWER8 servers. It is an example of a system design that complies with the described architecture. This reference design can be considered to be a “minimum production” configuration because it is designed and sized with a minimum set of elements that are generally appropriate for consideration as a minimum starting point for a production deployment.

The system management node is the cluster manager. It provides the initial provisioning and ongoing monitoring and management of the infrastructure. This architecture specifies Genesis (for provisioning) and OpsMgr (for monitoring and management) as the cluster manager.

	System node	Master node	Edge node	Worker Node		
Cluster type	All	All	All	Balanced	Performance	Storage-dense
Server model	1U S821LC	1U S821LC	1U S821LC	2U S822LC	2U S822LC	2U S822LC
Node count (min/max)	1/1	3/Any	1/Any	4/Any	4/Any	4/Any
CPU cores	8	20	20	22	22	22
Memory	32 GB	256 GB	256 GB	256 GB	512 GB	128 GB
Storage HDDs Storage SSDs	2X 4 TB HDDs	4X 4 TB HDDs	4X 4 TB HDDs	12X 4 TB HDDs	8X 6TB HDDs 4X 3.8 TB SDDs	12X 8 TB HDDs
Storage controller	(Internal)	MegaRAID 2 GB cache	MegaRAID 2 GB cache	MegaRAID 2 GB cache	MegaRAID 2 GB cache	MegaRAID 2 GB cache
Network 1 GbE	4 Ports (Internal)	4 Ports (Internal)	4 Ports (Internal)	4 Ports (Internal)	4 Ports (Internal)	4 Ports (Internal)
Network 10 GbE	1X 2-port	1X 2-port	2X 2-port	1X 2-port	1X 2-port	1X 2-port
Operating system	RHEL 7.2	RHEL 7.2	RHEL 7.2	RHEL 7.2	RHEL 7.2	RHEL 7.2

Figure 3-4 Hortonworks Data Platform Hadoop cluster On POWER8 server reference design

> **Important:** This reference design is intended as a reference only. Any specific design, with appropriately sized components that are suitable for a specific deployment, requires extra review and sizing that is appropriate for the intended use.

3.4 Physical configuration with rack layout

Figure 3-5 on page 29 shows the physical layout of a typical Power Systems server configuration (with eight worker nodes in the HDP cluster) within a rack. All of the components for the reference architecture design fit within a single 42U rack with space for growth and expansion.

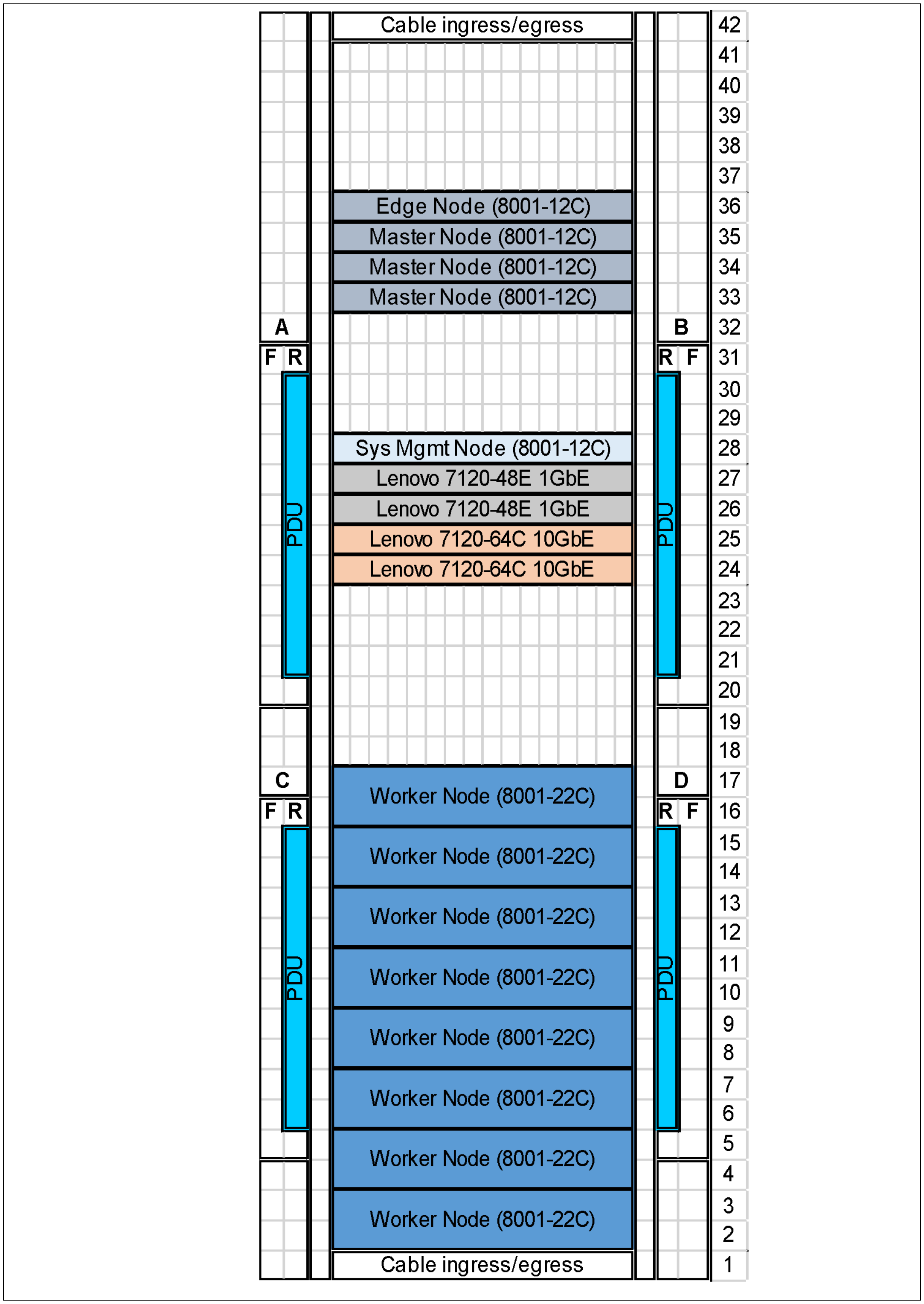


Figure 3-5 Physical configuration with rack layout

4

Enterprise Data Warehouse on Hadoop optimization

This chapter describes various Enterprise Data Warehouse (EDW) optimization techniques and tools that are available on IBM Power Systems servers, including:

- Traditional EDW offload to Hadoop
- Using IBM Big SQL as a federated structured query language (SQL) engine for Hadoop and distributed platform to deliver easy data querying across the enterprise
- Using Elastic Storage System with IBM Spectrum Scale for storage-dense Hadoop Hortonworks Data Platform (HDP) EDW workloads, and HDP on Power Systems sizing guidelines
- Performance tuning preferred practices and tools that are available on Power Systems servers

4.1 Traditional Enterprise Data Warehouse offload

There are two use cases for traditional EDW offloading to Hadoop. The first is for those existing data warehouses that reached their capacity and are facing performance and scalability issues. The second is to offload Extract, Transform, and Load (ETL) functions from an existing EDW to Hadoop EDW.

The Hadoop platform offers a cost-effective storage and compute solution. To analyze data in the current EDW, identify and move infrequently used data (cold data) or data that needs to be archived (for example, for regulatory purpose) into a Hadoop EDW. This action reduces the storage challenge on the existing EDW and frees resources. An existing traditional EDW continues to provide day-to-day analysis reports for the business, and EDW on Hadoop complements the existing EDW and provides extra and deeper analysis and insights.

Traditional EDW spends much effort on the ETL process. The ETL process also consumes EDW's precious resources that instead should be used for analysis purpose. As traditional EDW grows, the ETL process time also grows, and sometimes there is not a large enough batch window to process data from Online Transactional Processing (OLTP) data sources into EDW. To solve this problem, the ETL processes can be offloaded to Hadoop.

Hadoop does not require a predefined data schema, which is a perfect place to store data in the raw atomic format, that is, Hadoop provides a perfect data landing area for raw data. From this landing zone, data can be ingested, cleaned, and transformed before loading it into an existing EDW. Identify and analyze current ETL jobs, and choose a data integration tool that can access and connect to various data sources, provide data quality control, and run on the Hadoop platform.

In summary, offloading an existing EDW to Hadoop frees resources on an existing EDW keeps an existing EDW focused on business operational analysis and business intelligence, and increases the EDW efficiency in a cost-effective way.

An example of a traditional EDW offload to Hadoop is shown in Figure 4-1.

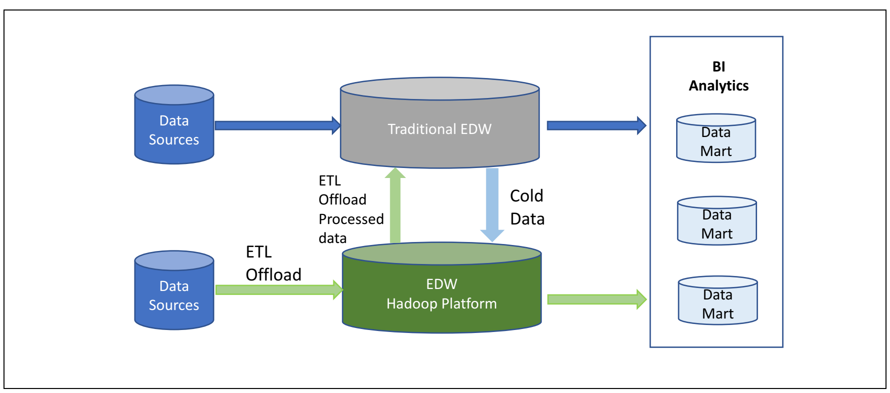


Figure 4-1 Traditional EDW offload to Hadoop

4.2 IBM Elastic Storage Server and IBM Spectrum Scale

Storage-dense workloads on the HDP data warehouse can be optimized by separating the data storage from the compute nodes. IBM Elastic Storage Server (ESS) with IBM Spectrum Scale provide a perfect optimization solution for these workloads.

Data storage

The ESS is a high-performance storage solution that is built by using the latest modular storage technology and POWER processor-based servers that are combined with IBM Spectrum Scale, which is a scalable, high-performance file management system. These elements are tuned, tested, and are offered in a scalable building block configuration that provides up to 5 TB of raw capacity at a 40 GBps peak data rate per storage enclosure.

A storage server is an element that provides an IBM Spectrum Scale file system that is accessed by the HDP cluster (specifically, the IBM Spectrum Scale components that are installed within the HDP cluster nodes). The storage server can be administered and managed by its native UI and functions independently from the HDP cluster.

File system

In 2015, IBM rebranded IBM General Parallel File System (IBM GPFS™) as IBM Spectrum Scale. GPFS was a high-performance clustered file system that was developed by IBM. Hadoop Distributed File System (HDFS) currently is the most popular file system that is used to hold data. IBM Spectrum Scale is an enterprise-level distributed file system that is a high-performance and cost-effective alternative to HDFS for Hadoop analytics services. IBM Spectrum Scale file system components can be installed on the ESS as part of the file system implementation. The IBM Spectrum Scale HDFS Transparency connector provides transparency for the underlying file system for the Hadoop applications.

4.2.1 Introduction to the IBM Elastic Storage Server

This section introduces the ESS solution, its characteristics, and where it fits in the business environments.

ESS is a modern implementation of software-defined storage (SDS), combining IBM Spectrum Scale software with IBM POWER processor-based I/O-intensive servers and dual-ported storage enclosures.

Instead of hardware-based disk RAID controllers, the IBM declustered RAID technology that is based on the IBM Spectrum Scale RAID software maintains all RAID capabilities, enabling superior data protection and rebuild times that are reduced to a fraction of the time that is needed with hardware-based RAID controllers.

Software RAID

The IBM Spectrum Scale RAID software that is used in the ESS solution runs on standard serial-attached SCSI (SAS) disks in just a bunch of disks (JBOD) arrays, which enables cost reduction, with the option of solid-state drives (SSDs) when more performance is needed. The solution does not require or use any external RAID controller or acceleration.

IBM Spectrum Scale RAID supports multiple RAID codes and distributes client data, redundancy information, and spare space across the disks in such a way that if there is a physical disk loss, or even a group of physical disk losses, it does not affect data availability.

Also, instead of relying only on the disks to detect and report faults, read or write errors, and other integrity problems, IBM Spectrum Scale RAID implements an end-to-end checksum.

Note: For more information about ESS, see IBM Elastic Storage Server.

For more information about RAID codes, see IBM Knowledge Center.

IBM Elastic Storage Server advantages

Why ESS instead of any other hardware-controlled solution or in-house built solution? There are many reasons:

- IBM Spectrum Scale RAID and support from IBM for the solution contribute to the added value, including a tuning solution from the IBM services team.
- ESS offers scalability from 40 TB to hundreds of petabytes, supporting 10-Gigabit Ethernet (10 GbE), 40-Gigabit Ethernet (40 GbE), and FDR InfiniBand.
- Big data requires easy storage growth. The ESS building block approach adds more storage servers, which add to the overall capacity, bandwidth, and performance, all within a single name space.
- By using JBODs and IBM Power Systems servers, the ESS provides price/performance storage for analytics, technical computing, and cloud computing environments.
- The ESS modern declustered RAID technology is designed to recover from multiple disk failures in minutes, versus hours and days in older technology, giving the solution predictable performance and data protection. 8+2 and 8+3 RAID protection and platter-to-client data protection are included with the ESS.
- IBM engineering and testing teams worked on designing the solution in a way that, from adapter placement on the servers, number of adapters, number of disks on the drawers, cabling, number of drawers for the software versions, and the way data is placed on disks, every part of the solution is meant to provide greater performance and data availability.
- The solution is tested after it is manufactured, and tools and scripts are used by the IBM services team during the delivery to ensure that every piece of the solution integrates together.
- IBM ESS uses IBM Spectrum Scale RAID with de-clustered erasure coding to distribute and protect data. Support for RESTFUL APIs enables reduction in OPEX.
- Support for file (POSIX, NFS, and SMB), Object (S3 and Swift), Block (iSCSI), and Hadoop (HDFS).

4.2.2 Introduction to IBM Spectrum Scale

This section introduces the IBM Spectrum Scale solution.

IBM Spectrum Scale is a cluster file system that supports varied deployment models, including IBM Spectrum Scale File Placement Organizer (FPO) for storage-rich servers, IBM Spectrum Scale on SAN shared storage, and an integrated system that is called the ESS. These solutions can host various analytics services, such as MapReduce and Spark, with the help of IBM Spectrum Scale HDFS Transparency Hadoop connector (the second-generation IBM General Parallel File System (GPFS) Hadoop Connector).

Traditional Hadoop Analytics Solutions challenges

Traditional Hadoop analytics systems have a dedicated cluster to run Hadoop services. The basic concept is to integrate compute and storage resources that are derived from a storage-rich server.

In this compute and storage-integrated system, users and system administrators face the following key challenges:

- The need to build an analytics system from scratch for compute resource and storage.
- The inability to disaggregate storage resources from compute resources. To add storage capacity in the form of data nodes, an administrator must add processing and networking even if they are not needed. This coupling of compute and storage limits an administrator's ability to apply automated storage tiering to take advantage of hybrid SSD or rotating disk architectures.
- HDFS does not support the industry-standard POSIX access interface. So, to manage data, users must import data from systems such as databases and file store systems to a Hadoop analytics cluster. Then, the analyzed result is exported back to another system. Data I/O processes can take longer than the actual query process.
- Many Hadoop systems lack enterprise data management and protection capability, such as data lifecycle management.
- A highly skilled system administrator is required to maintain HDFS.
- Data stability is highly affected by any server that is down in the cluster.

IBM Spectrum Scale advantages

The Apache Hadoop framework features open source software that enables distributed processing of large data sets across clusters of commodity servers. It is designed to scale up from a single server to thousands of machines, with a high degree of fault tolerance.

The Apache Hadoop framework enables distributed processing of large data sets across clusters of computers that use simple programming models. The Apache Hadoop framework processes data-intensive computational tasks, which include data amounts that can range from hundreds of terabytes to tens of petabytes. This computation mode differs from the computation mode that is used in traditional high-performance computing (HPC) environments.

For example, Hadoop consists of many open source modules. One of the primary open source modules is the HDFS, which is a distributed file system that runs on commodity hardware. HDFS lacks enterprise class capabilities that are necessary for reliability, data management, and data governance. IBM Spectrum Scale, which is a scale-out distributed file system, offers an enterprise-class alternative to HDFS.

Built upon GPFS, IBM Spectrum Scale includes NFS, SMB, and object services, and meets the performance that is required by many industry workloads, such as technical computing, big data, analytics, and content management:

- Compute clusters (technical computing)
- Big data and analytics
- HDFS
- Private cloud
- Content repositories
- FPO

In-place data analytics

IBM Spectrum Scale is POSIX-compatible, which supports various applications and workloads. With IBM Spectrum Scale HDFS Transparency Hadoop connector, you can analyze file and object data in place with no data transfer or data movement. Existing systems and the analytics systems are using and sharing data that is hosted on an IBM Spectrum Scale file system, as shown in Figure 4-2.

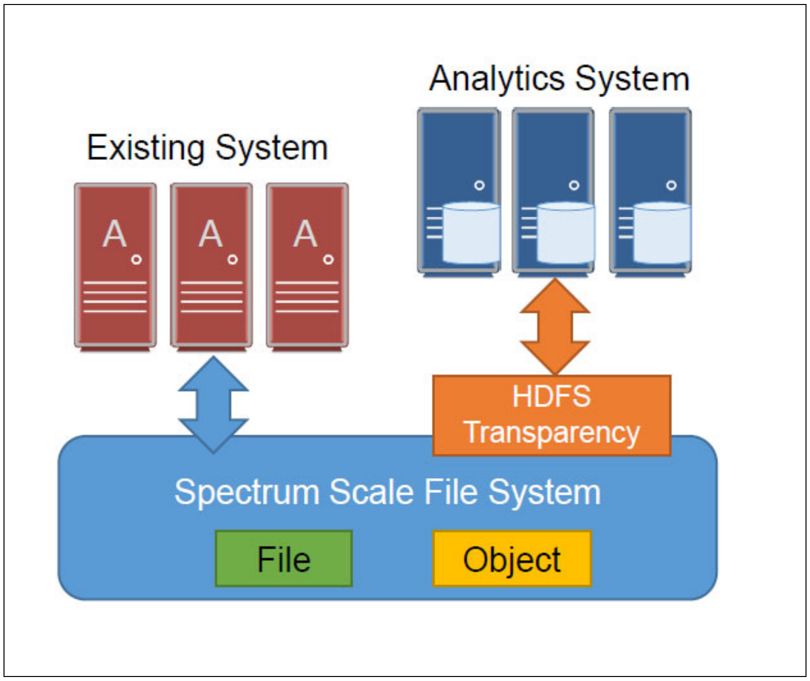


Figure 4-2 IBM Spectrum Scale in-place data analytics

Hadoop services can use a storage system to save on IT costs because no special-purpose storage is required to perform the analytics. IBM Spectrum Scale features a rich set of enterprise-level data management and protection features, such as snapshots, ILM, compression, and encryption, which provide more value than traditional analytic systems.

IBM Spectrum Scale HDFS Transparency connector

The IBM Spectrum Scale HDFS Transparency connector is the second generation of the IBM Spectrum Scale Hadoop connector, which features significant improvement in performance, scalability, and compatibility. It features the following key advantages:

- The IBM Spectrum Scale Client is not needed on every Hadoop node.
- Includes full Kerberos support.
- Supports more Hadoop components and native features, such as Impala, distcp, and webhdfs.
- Uses an HDFS client cache.

The IBM Spectrum Scale HDFS Transparency node communication paths to the HDFS Client and HDFS Node are shown in Figure 4-3.

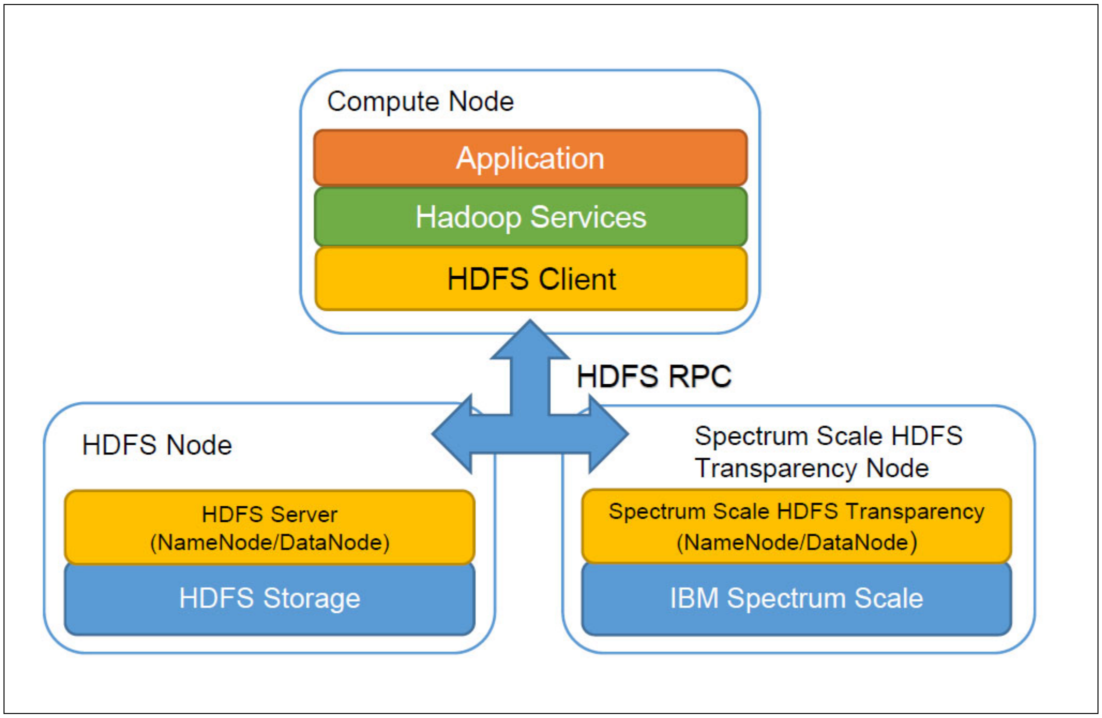


Figure 4-3 IBM Spectrum Scale HDFS Transparency

The IBM Spectrum Scale HDFS Transparency connector communicates with the HDFS client directly to service HDFS RPC requests and sends I/O requests to IBM Spectrum Scale. Hadoop analytics services. HDFS clients use the same method to communicate to IBM Spectrum Scale HDFS Transparency nodes and other HDFS server nodes. With the HDFS federation feature, IBM Spectrum Scale and HDFS can service a Hadoop cluster concurrently.

The IBM Spectrum Scale HDFS Transparency connector supports the following features:

- Auto High Availability (HA) of HDFS Transparency NameNode

  IBM Spectrum Scale Transparency NameNode can be configured with auto-HA. When the Active NameNode is down or it detects some failure, for example, when the IBM Spectrum Scale file system is unmounted, it automatically fails over to the standby NameNode to provide continues I/O service.

- Distcp

  With the distcp feature in IBM Spectrum Scale HDFS Transparency, you can migrate data from HDFS into IBM Spectrum Scale or share data between HDFS storage and the IBM Spectrum Scale file system.

- Federation

  With the IBM Spectrum Scale HDFS Transparency federation feature, applications can access data from native HDFS storage and IBM Spectrum Scale simultaneously.

4.2.3 Hadoop support for IBM Spectrum Scale

IBM Spectrum Scale provides integration with Hadoop applications that use the Hadoop connector (so you can use IBM Spectrum Scale enterprise-level functions on Hadoop):

- POSIX-compliant APIs or the command line
- FIPS- and NIST-compliant data encryption
- Disaster recovery
- Simplified data workflow across applications
- Snapshot support for point-in-time data captures
- Simplified capacity management by using IBM Spectrum Scale (for all storage needs)
- Policy-based information lifecycle management capabilities to manage PBs of data
- Infrastructure to manage multi-tenant Hadoop clusters based on service-level agreements (SLAs)
- Simplified administration and automated recovery
- Multiple clusters

4.2.4 Hortonworks Data Platform on IBM Power Systems with IBM Elastic Storage Server: Reference Architecture and Design

If you want to produce a design for an HDP on IBM Power Systems with ESS solution, a proper architecture provides guidance and much of the fundamental material (including references to related materials and other sources) that should be understood to produce a satisfactory design. Reference designs provide examples that can assist with understanding the architecture and provide design elements and various design patterns that may be adopted when producing the design of interest.

If you want to adopt and adapt a design for an HDP on IBM Power Systems with ESS solution, proper reference designs may be used as a starting point for the design of interest. Reference designs may be modified to potentially meet the requirements for the design of interest.

An excellent place to get started on building an effective platform on IBM Power Systems is *Hortonworks Data Platform on IBM Power with IBM Elastic Storage Server Reference Architecture and Design Version 1.0*.

Note: For more information, see the following resources:

- *Deploy a Big Data Solution on IBM Spectrum Scale*
- IBM Spectrum Scale HDFS Transparency
- *IBM Spectrum Scale FPO Deployment and Maintain Guide*

4.3 SQL engine on Hadoop: IBM Big SQL

IBM Big SQL is a high-performance, massively parallel processing (MPP) SQL engine for Hadoop, with support for various data sources, including HDFS, relational database management systems (RDBMSs), NoSQL databases, object stores, and WebHDFS, by using a single database connection. IBM Big SQL offers low latency queries, security, SQL compatibility, and federation capabilities, enabling organizations to derive value from enterprise data.

IBM Big SQL V5.0.1 is a hybrid, high-performance SQL engine for Hadoop, with support for various data sources, including HDFS, RDBMSs, NoSQL databases, object stores, and WebHDFS. IBM Big SQL offers low latency queries, security, SQL compatibility, and federation capabilities, enabling organizations to derive value from enterprise data.

The IBM Big SQL server is multi-threaded, so scalability is limited only by the performance and number of cores on the computer that runs the server. If you want to issue larger queries, you can increase the hardware performance of the server computer that IBM Big SQL runs on, or chain multiple IBM Big SQL servers together to increase throughput.

All of the data is stored in Hadoop. You can save on costs by offloading data to Hadoop and use next-generation, fit-for-purpose analytics with the SQL optimizer for Hadoop.

4.3.1 Key IBM Big SQL features

IBM Big SQL features include easy-to-use tools and its MPP SQL engine, which provide powerful SQL processing features:

- Superior SQL-on-Hadoop performance by using elastic boost technology.
- The ability to understand SQL dialects from other vendors and products for Oracle, IBM DB2 and IBM Netezza®, which helps make it faster and easier to offload old data from existing EDWs or data marts to free capacity, while preserving most of the SQL from those platforms.
- A hybrid engine for Hadoop that uses Hive, HBase, and Apache Spark concurrently for best-in-class analytic capabilities.
- Deeper integration with Spark than other SQL-on-Hadoop technologies, enabling new use cases.
- Advanced row and column security: The integration of IBM Big SQL and Spark enhances “shopping for data” security because sensitive attributes can be masked by default with no back doors to the data, which empowers self-service access to data in a safe and governed manner.
- With IBM Big SQL, you can manage your system and run your queries with our easy-to-use tools:
  - With the Ambari dashboard, an administrator can easily start and stop services, set up users, groups, and views, and define alerts and notifications.
  - With Data Server Manager, administrators can manage schemas and tables, and developers can create and run IBM Big SQL queries and UDFs in a simple and powerful user interface.
  - Developers can use Apache Zeppelin notebooks to interactively run data-intensive applications by using Apache Spark on the Hadoop cluster.

SQL processing features

The IBM Big SQL MPP SQL engine provides several SQL processing features.

- Standards-compliant ODBC and JDBC: For developers, the usage pattern enables you to access the database with specific products or tools that allow only ODBC or JDBC.
- Point queries: Enables short rapid queries that search by key or key ranges. Uses HBase for point queries and rapid insert, and to update and delete workloads.
- Apache Spark integration: Integrates with Spark for easier insight delivery and faster processing. It provides an ANSI-compliant SQL parser that can run 99 TPC-DS queries and structured streaming with new APIs.
- Federation capabilities: Uses a single database connection, enabling you to access data across Hadoop and relational databases, whether they are on the cloud, on premises, or both. Includes Fluid Query capabilities to enhance virtualization with various data warehouses. You can also federate with S3 Object Storage and WebHDFS (technology preview only).
- Integration with IBM BLU Acceleration®: As a technology preview, integrates with BLU Acceleration, a set of advanced query processing technologies that includes column-organized tables, actionable compression, data skipping, CPU optimization, and scan-friendly memory caching.
- Integration with Yet Another Resource Negotiator (YARN) through Slider: Improved integration with YARN is achieved through Apache Slider, enabling non-YARN services to be deployed on YARN, and control over YARN resource allocation and scheduling.
- Compatibility with multiple SQL dialects: Understands SQL dialects from offerings such as IBM DB2 database, IBM Netezza data warehouse appliances, and Oracle database. This makes the platform suited for RDBMS offload, and fast and easy consolidation. Data can be offloaded from existing EDWs or data marts to free capacity while preserving most of the familiar SQL.
- Elastically scalable: Offers the only SQL engine to successfully run all 99 TPCDS queries up to 100 TB with numerous concurrent users. Ability to run multiple workers per node for efficient CPU and memory utilization.
- Secure SQL: Robust role-based access control (RBAC), row-based dynamic filtering, column-based dynamic masking, and Ranger integration to provide centralized security administration and auditing for data lakes.

4.3.2 IBM Big SQL architecture

Built on IBM common SQL database technology, IBM Big SQL is an MPP database engine that has all the standard RDBMS features and is optimized to work with the Apache Hadoop infrastructure.

Figure 4-4 illustrates the IBM Big SQL architecture.

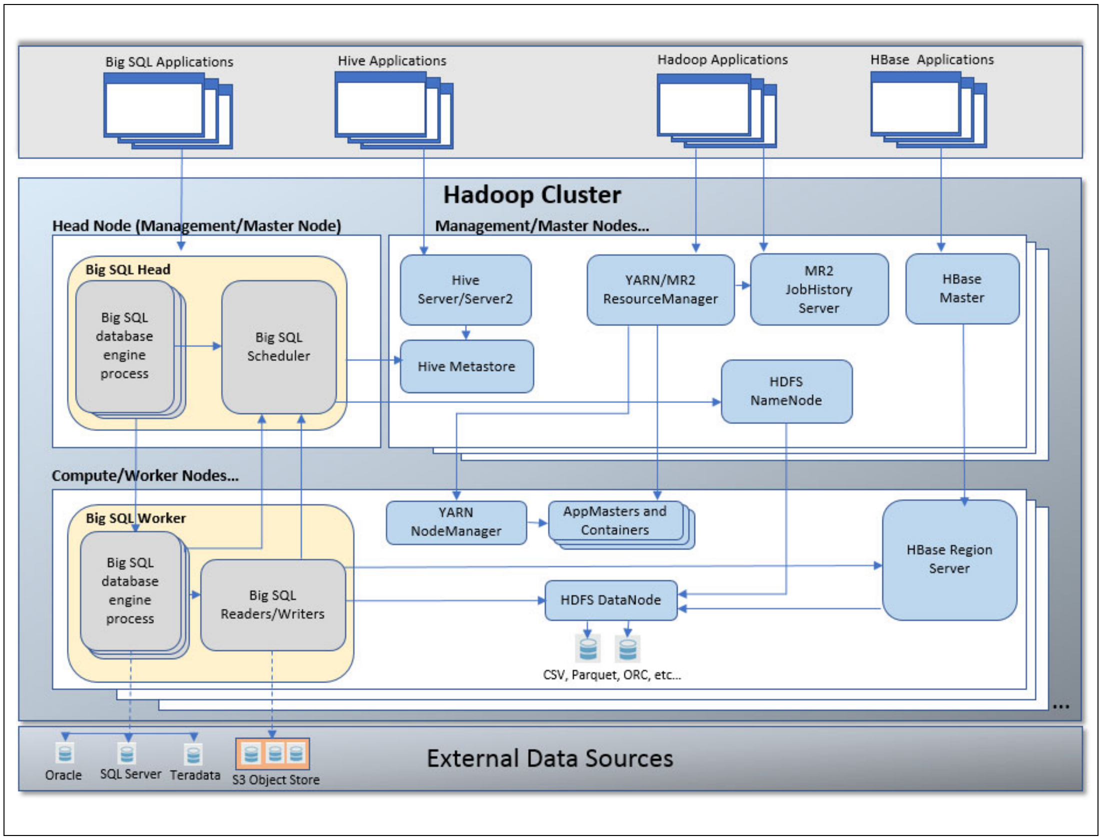


Figure 4-4 IBM Big SQL architecture

The IBM Big SQL server or service consists of one IBM Big SQL head (two heads in an HA configuration) that is installed on a node that is called the head node, and multiple IBM Big SQL workers that are installed on nodes that are called worker nodes.

How IBM Big SQL processes HDFS data

The following steps represent a simple overview of how IBM Big SQL processes HDFS data:

1. Applications connect to the IBM Big SQL head on the head node.
2. Queries that are submitted to the IBM Big SQL head are compiled into optimized parallel execution plans by using the IBM common SQL engine's query optimizer.
3. Plans are distributed. The parallel execution plans are then distributed to IBM Big SQL workers on the worker nodes.
4. Data is read and written. Workers have separate local processes called native HDFS readers and writers that read or write HDFS data in its stored format. An IBM Big SQL reader is composed of IBM Big SQL processes that run on the IBM Big SQL worker nodes and read data at the request of IBM Big SQL workers. Similarly, an IBM Big SQL writer is composed of IBM Big SQL processes that run on the IBM Big SQL worker nodes and write data at the request of IBM Big SQL workers. The workers communicate with these native readers or writers to access HDFS data when they process execution plans that they receive from the IBM Big SQL head.
5. Predicates are applied. Native HDFS readers can apply predicates and project columns to minimize the amount of data that is returned to workers.

4.3.3 Advantages of using IBM Big SQL with big data

IBM Big SQL on POWER helps HDP to compete in a market segment that requires full SQL compliance, high performance, and currency with complex queries.

IBM Big SQL offers the following capabilities for big data and modern data warehouse needs:

- Federation and integration.

  With its federation capabilities, which use Fluid Query, IBM Big SQL can virtualize data from many different data stores, such as Hive, HBase, Spark, DB2, Oracle, SQL Server, Netezza, IBM Informix®, Teradata, WebHDFS, and object store. These federation capabilities can be managed from a UI.

  IBM Big SQL offers bidirectional integration with Spark, Apache Spark 2.1 integration, and features efficient synthesis between Spark executors and IBM Big SQL worker nodes. IBM Big SQL uniquely uses Apache Hive, Apache HBase, and Spark concurrently for best-in-class analytics capabilities.

- SQL compatibility.

  IBM Big SQL is the IBM SQL interface in the Apache Hadoop environment. You can use the familiar standard SQL syntax in IBM Big SQL, and SQL extensions, with Hadoop-based technologies. You can use IBM Big SQL to query, analyze, and summarize data.

  IBM Big SQL supports several underlying storage mechanisms on the HDFS. IBM Big SQL also supports HDFS transparent encryption, which means that data is decrypted as it is read, but the files themselves remain encrypted.

  IBM Big SQL supports JDBC and ODBC client access from Linux and Windows platforms, which enables you to use your SQL skills, SQL-based business intelligence applications, and query or reporting tools to query data.

- Elastic scalability and unmatched performance.

  IBM Big SQL now offers YARN integration through Slider for efficient allocation of resources. In addition, IBM Big SQL added a technology that is called *Elastic Boost* that can significantly improve its performance (up to 50%). It enables allocation of multiple workers per node for more efficient CPU and memory utilization.

  IBM Big SQL comes with an ANSI-compliant SQL parser that can run all the 99 TPC-DS queries without the need for query modifications and structured streaming with new APIs. It can scale up to 100 TBs with multiple concurrent users with fewer number of nodes than many other SQL engines.

- Concurrency.

  The IBM Big SQL engine can handle and coordinate queries even if hundreds of concurrent users are running the most difficult, complex queries. Again, this capability is great for data warehousing workloads in which you definitely have many users hitting the platform.

  IBM Big SQL works exceptionally well in environments where data exceeds available memory. Architecturally, this factor is the only way that you can get maximum concurrency, and it is what IBM Big SQL is built for. IBM Big SQL makes offloading your data warehouse and giving mainstream users a query-able archive of older data far easier to support more use cases such as predictive modeling and machine learning, which get more powerful with more data. Because the data is all in Hadoop, data scientists can do whatever they like with it, using whatever tools are best for the job.

- Why is IBM Big SQL better than other SQL engines for Hadoop?

  Hive gets more performance by scaling to more nodes by using MapReduce, but just because you can scale to hundreds or thousands of machines to reach your performance objectives does not mean that you want to do so. Spark changes scale-out performance by using in-memory processing, and now you seeing the same workloads run faster with fewer nodes by using Spark.

  Although you try to get more performance by using memory, IBM Big SQL goes in the opposite direction. IBM Big SQL works exceptionally well in environments where data exceeds available memory. Architecturally, this is the only way that you can get maximum concurrency, and this is what IBM Big SQL is built for.

  In-memory approaches that are used by technologies such as Spark SQL are fast but have challenges scaling to support many concurrent users running complex SQL. SQL is a language, not a workload. IBM Big SQL is engineered for warehousing on Hadoop. Spark SQL and IBM Big SQL can be used concurrently for the same data sets on Hadoop because they are open. Use Spark SQL, IBM Big SQL, and other SQL tools together.

- Enterprise security.

  IBM Big SQL comes with built-in row and column level security. It is granular, rule-based, and supports Hive tables, native (DB2) tables, and HBase tables uniformly. It also provides RBAC for easier management of access and privileges. In addition, IBM Big SQL supports policy-based authorization through Ranger with its Ranger plug-in. IBM Big SQL can also help enhance Hive security by providing selective data masking capabilities without the need for lookup tables.

  All these features work seamlessly with HDP.

4.3.4 Overview of IBM Big SQL federation

Federation can be described as an architecture for data warehousing in which different data sources connect and exchange information through a central federation database. IBM Big SQL can play the role of the central federation database when the federation feature is enabled.

Federation relies on three base objects to create a connection to a remote database: a wrapper, a server, and one (or more) user mappings, along with other important concepts, including function mappings and nicknames:

Wrapper A wrapper is an object that enables the federation server to support a specific type of data source. These objects hold information about the versions that are supported for their particular data types and what the default function mappings are.

Server A server is an object that represents a connection to a specific remote data source. It holds information about the remote database, for example, whether the remote data source has a collation sequence matching the IBM Big SQL binary collation.

User mapping A user mapping is an object that holds information about the user, which is used as proxy when communicating with, and running at, the remote data source. User mappings can be at a user level, where each local user is mapped to a remote user, or they can be public, where any user who authenticates to IBM Big SQL is mapped to a single remote user.

Function mappings Function mappings are objects that map a local function to a remote function, usually because the function either has a different return type, different arguments, or even a different name. When a wrapper is created, some default function mappings are put into place, but it is possible to create a mapping for any function.

Nickname Nickname is an object that represents a remote table. In federation, it is possible to use three-part names to access a remote table. A nickname is more than a simple connection to the remote table. It adds an entry to the local catalog to collect statistics that are used by the optimizer to calculate the cost of a query.

Figure 4-5 illustrates the architecture and how federation objects relate to real objects.

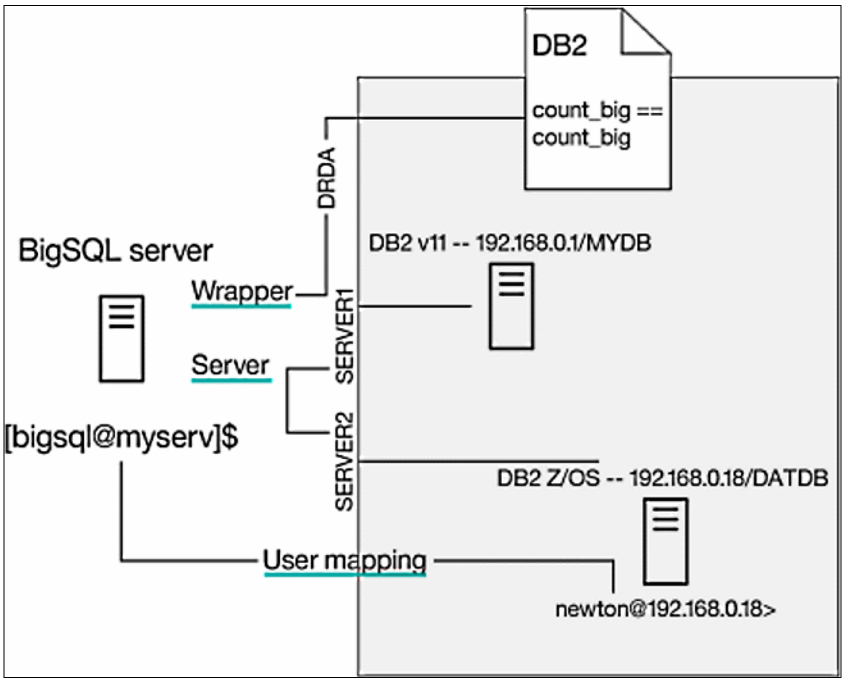


Figure 4-5 IBM Big SQL federation architecture

For more information about IBM Big SQL, see the following resources:

- Best Practices in the IBM Hadoop IBM Big SQL - IBM developerWorks
- Customer recommendations for using IBM Big SQL
- IBM Knowledge Center

4.4 Hortonworks Data Platform on Power Systems sizing guidelines

The following section describes the general factors that affect cluster sizing. Each implementation is unique and there are other factors that can affect the cluster performance and sizing. Always contact your IBM technical team for proper sizing.

- Use the reference architecture design that is provided in 3.3, “Hortonworks Data Platform on Power Systems reference architecture” on page 25 as the start point.
- Determine cluster workload types: Balanced, performance (compute-intensive), or storage-oriented (I/O-intensive). If the workload is unknown, choose balanced.
- Determine application types and complexity, for example, machine learning increases the CPU to memory ratio, so the Spark workload requires more memory.
- Data volume size affects cluster and storage sizing. Consider initial data volume, and annual growth rate, compression ratio, and temporary immediate files.
- Consider the data replication factor and node redundancy factor. For example, data might need to be replicated across three worker nodes; and minimum of three master nodes for HA redundancy is needed.
- For the master node, start with three nodes. As cluster nodes grow, the number of master nodes might need to be increased. Figure 4-6 shows some preferred settings.

Cluster Size Type	Number of Nodes in the Cluster	Number of Master Nodes
Tiny	< 8	
Mini	8 - 16	3
Small	17 - 40	4 - 6
Medium	41 - 120	7 - 9
Large	121 - 512	10 - 12
Jumbo	>512	Consulting required

Figure 4-6 Master node count

- Determine which services are exposed externally because this situation affects firewall ports and the network DMZ.
- When you use IBM Elastic Storage Systems (ESS) for storage-dense workloads, a local disk is still required for storing temporary files. Local storage in general is around 20 - 30% of the processed data volume.

4.5 Performance tuning

Here are the general steps of performance tuning:

1. Take advantage of IBM Power Systems CPU multi-threading for best performance.

   One of the advantages of IBM Power Systems CPUs is the fact that each CPU core has up to eight threads versus x86 CPUs, which have two threads per core. Therefore, some tuning is required to take full advantages of eight threads to increase Hadoop cluster performance. POWER9 processor cores come in two versions. Those with four threads per core, and those with 8.

   By default, Hortonworks sets the YARN and MapReduce processes in an aggressive way. It sets the memory that is used by each map and reduce container with a high value. Although the high value makes sense for other architectures, this value might not work well for POWER processors. For POWER processors, these values must be tuned to fit the number of hardware threads per core (SMT-8, SMT-4, SMT-2, or SMT). These values must be adjusted depending on the application that is used.The values must be calculated based on the number of CPUs and memory that is available on Power Systems servers in a cluster.

   Here are the settings that we used in `yarn-site.xml` for an HDP cluster on Power System servers as tuning examples, which are followed by a description of their effects:

   – `Yarn.nodemanager.resource.cpu-vcores = 160`

     This parameter is to set to the number of CPU cores that can be allocated for the containers. The server that is used for this cluster has 20 cores, with eight threads per core, and the total number of virtual CPU cores is 160.

   – `Yarn.nodemanager.resource.memory-mb = 307200`

     The amount of physical memory, in megabytes, that can be allocated for the containers. The Power Systems server that is used in this cluster has total of 512 GB of memory, with some of the memory that is consumed by the operating system, so 300 GB is allocated for the containers.

   – `Yarn.scheduler.maximum-allocation-mb = 307200`

     The maximum memory allocation for every container request at the resouce manager, in megabytes. Memory requests higher than this setting do not take effect, and are capped to this value.

   Here are sample settings in the `mapred-site.xml` file:

   – `mapreduce.job.reduce.slowstart.completedmaps = 0.99`

     This setting specifies the proportion of the total number of map tasks in a job that must be completed before any reduce tasks are scheduled. 0.99 means that the reduce tasks start when almost all map tasks are completed. Our tests show that this setting increases the overall performance.

   – `mapreduce.map.memory.mb = 5324`

     The amount of memory that the scheduler requests for each Map task, in megabytes.

– `mapreduce.reduce.memory.mb = 5324`

  The amount of memory that the scheduler requests for each Reduce task, in megabytes.

– `mapreduce.map.java.opts = -Xms4096m -Xmx4096m -server -XX:+UseParallelOldGC -XX:ParallelGCThreads=8 -XX:ConcGCThreads=2 -XX:+UseAdaptiveSizePolicy -XX:+PrintTenuringDistribution -XX:+PrintGCDateStamps -XX:+PrintGCDetails -XX:+PrintAdaptiveSizePolicy -Xloggc:/tmp/gclog/java_ParOld.log -XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=10 -XX:GCLogFileSize=2048K -Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.authenticate=false -Dfile.encoding=UTF-8 -Djava.io.tmpdir=/tmp/gclog -verbose:gc`

  Some of the parameters that are passed to the JVM (map/red container) at the start of the job. Set the size of the JVM to be a little less than the `map/reduce memory.mb`;, and the number of the Garbage Collector threads set to eight to take advantage of the POWER CPU's eight threads.

  The above settings are, for example, only because each value must be changed according to the size of the cluster, the server configuration, and the application running on the cluster. Also, most of the values can be set when submitting the job to overwrite the original settings, for example:

  ```
  yarn jar $YARN_EXAMPLES/hadoop-mapreduce-examples.jar wordcount
  -Dmapreduce.map.memory.mb=1024 /inputdir/inputfile /outputdir/
  ```

2. Setting the directory.

   Set the directory that is used by YARN and HDFS to be in the HDFS instead of on the Linux `rootvg` directory by using the following configurations:

   – In "`yarn-site.xml`":

   ```
   yarn.nodemanager.local-dirs
   yarn.nodemanager.log-dirs
   ```

   – In "`hdfs-site.xml`":

   ```
   dfs.datanode.data.dir
   dfs.namenode.name.dir
   ```

   Logs that are generated by YARN and metadata that is generated by the namenode (HDFS) can be high volume and high speed, and they can fill the Linux default file system quickly. Therefore, they should be configured to be distributed across disks on HDFS to avoid running out of disk space on the Linux root directory.

3. Network settings.

   Both master and compute nodes should be on the same private subnet. Network bandwidth should be uniformed for optimal cluster performance. For example, for a 10 GB connection workload, each node should have an active 10 GB card.

4. Environment variables.

   The `.bashrc` file is a shell script that is run every time that a user opens a new shell. The `.bashrc` file is located in multiple locations:

   – `/etc/skel/.bashrc`

   – `/root/.bashrc`

   – `/home/<user>/.bashrc`

When a new user is created, the `/etc/skel/.bashrc` file is copied into the user's home directory. The `.bashrc` file is a powerful tool to customize the Linux shell, so if this file is changed, makes sure that it is available throughout the cluster to ensure that the `.bashrc` file is uniform across the cluster for optimal cluster performance. For large clusters, this file can be pushed by a configuration management software, such as Puppet.

5. Transparent Huge Pages (THP).

   Huge pages are blocks of memory that come in 2 MB and 1 GB sizes. The page tables that are used by the 2 MB pages are suitable for managing multiple gigabytes of memory, and the page tables of 1 GB pages are suitable for scaling to terabytes of memory. Huge pages can be difficult to manage, so Red Hat implemented THP.

   THP is *not* recommended for a database workload. Hortonworks recommends disabling THP.

6. Disk administration:

   – Mount the disk with the same options and same mount points across the nodes, especially the compute nodes.

   – Use a minimum of 1 - 2 CPU cores per physical HDD. If you use compression, use two CPU cores per HDD.

7. Disk Mount Option.

   Mount all Hadoop disks by using the **`noatime`** option:

   ```
   /dev/sdd1 /data1 ext4 defaults, noatime 0
   ```

4.6 Analyzing data by using IBM Data Science Experience Local

The main purpose of EDW is to provide data for business analysis and insights, so data science is a logical topic in the discussion of EDW on Hadoop.

The term *data science* was first introduced in the title of the International Federation of Classification Societies's (IFCS) 1996 biennial conference in Kobe, Japan: "Data Science, classification, and related methods". It is defined as follows:

"*Data Science leverages large volumes of data generated from numerous and diverse sources, as well as any new kind of cloud interaction as it becomes available in our increasingly digital world*."[1]

[1] https://uwaterloo.ca/data-science/

4.6.1 IBM Data Science Experience

Data scientists are faced with many problems, such as a rigid toolset, multiple, disjointed environments, a separate community for each tool or environment, no metadata or data lineage, difficult to maintain and control project assets, and limited means of collaborating with teams and results are difficult to share.

IBM Data Science Experience (IBM DSX) is an interactive, collaborative environment where data scientists can use multiple tools to activate their insights. Data scientists can work with a growing set of data science tools, such as RStudio, Jupyter, Python, Scala, Spark, and IBM Watson® Machine Learning (IBM ML). IBM DSX is available in cloud and on-premises (IBM DSX Local).

IBM DSX brings together everything that a data scientist needs to be more successful. It is based on three concepts:

Community A data scientist must be updated with news from the data science community. There are plenty of new open source packages, libraries, techniques, and tutorials that become available every day. A good data scientist follows the most important sources and shares their opinion and experiments with the community. The user interface of IBM DSX brings that community into the tool so that data scientists do not need to go to different places to do their research.

Open source Many companies rely on open source for data science. The best of open source is within IBM DSX, such as Spark, RStudio, and Jupyter.

Added value IBM DSX improves open source by adding more capabilities:

- Data Shaping takes 80% of the data scientist's time. IBM DSX provides tools through a visual GUI to help users better perform this task.
- A data scientist can run Spark jobs from within the IBM DSX.
- IBM DSX also provides integrated Jupyter Notebooks for interactive and collaborative development. and has RStudio built into the experience.
- IBM ML provides self-learning systems that can score, act, monitor, and ingest feedback to drive better business outcomes. IBM ML enables organizations to create end-to-end pipelines with their data. A pipeline includes everything from connecting to your data sources, preparing your data, modeling, model evaluation, to model deployment and management.

4.6.2 IBM Data Science Experience Reference Architecture

Figure 4-7 shows the IBM DSX reference architecture, from data collection to development, deployment, and integration with business applications.

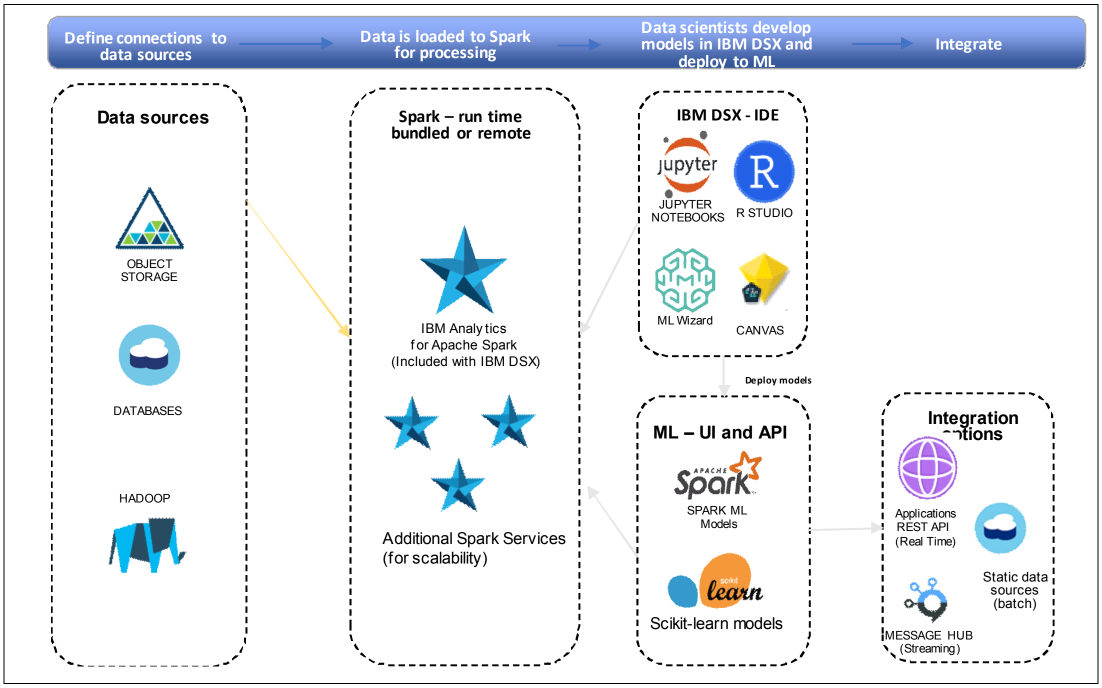


Figure 4-7 IBM Data Science Experience reference architecture

A data scientist or a data engineer defines connections to data sources. IBM provides a custom API for connecting to data sources, and if an API is not provided, the driver that is provided by the data source vendor can be used (for example, a JDBC driver).

At run time, data is loaded into Spark, where processing takes place. This can be the Spark cluster that is included with IBM DSX or an external Spark cluster that is accessed by Livy API.

Data scientists can use various tools in IBM DSX to perform analysis on data: Notebooks, R Studio, ML Wizard, and Canvas (currently in beta).

IBM ML in IBM DSX supports three types of deployment: online (real time), batch, and streaming. Online models are accessed by a REST API. Models that are deployed for batch scoring write output to static data sources (databases, Hadoop, object storage, and so on). Streaming deployments integrate with streaming data sources.

4.6.3 IBM Data Science Experience Local

Although organizations are increasingly embracing the public cloud, many require analytics operating locally within their firewall. There are a many reasons customers require this:

- Sensitivity of corporate data
- Gravity of enterprise data sources
- Performance from low latency and local data access
- Industry-specific security requirements
- Conservatism

IBM DSX Local uses the IBM competitive differentiator of deep domain expertise within verticals and proprietary data.

IBM DSX Local is a self-contained and extendable platform. it is self-contained because it includes everything that is needed to develop and deploy analytics applications. It is extendible because it works with remote Spark clusters, and multiple database and Hadoop data sources. IBM DSX Local also integrates with LDAP and supports adding more libraries to development environments.

Here are the IBM DSX Local features:

- The development tools in IBM DSX are Jupyter Notebooks and R studio.
- The runtime engine is Spark. Organizations can use Spark that is bundled with IBM DSX or remote Spark by using the Livy API.
- IBM ML capabilities provide functions for building and deploying SparkML and scikit-learn models.
- Collaboration features help data scientists organize analytics assets into projects that can be shared between users.
- A community enables researching and sharing of information, which can include anything from code samples to tutorials and data assets.
- IBM DSX Local includes Platform Administration dashboards that help IT to monitor the systems, view log files, and restart services when needed.

4.6.4 IBM Data Science Experience Local architecture

IBM DSX Local runs as a collection of *dockerized* services that are managed by Kubernetes.

Kubernetes handles the service orchestration by providing service monitoring and administration, HA, and service failure detection and automatic restart, adding or removing nodes dynamically, and online upgrades.

IBM DSX Local services running in Kubernetes include the following ones:

- UI services for browsers to connect to
- User authentication services
- Project services for user collaboration and data sharing
- Notebook services with enhanced access to Jupyter Notebooks
- A Spark service with access to sophisticated analytics libraries
- Pipeline and model building services
- Data connection building service for access to external data
- Various internal management services

Figure 4-8 shows the IBM DSX Local Cluster architecture in a 9-node configuration with the components layout on each server type.

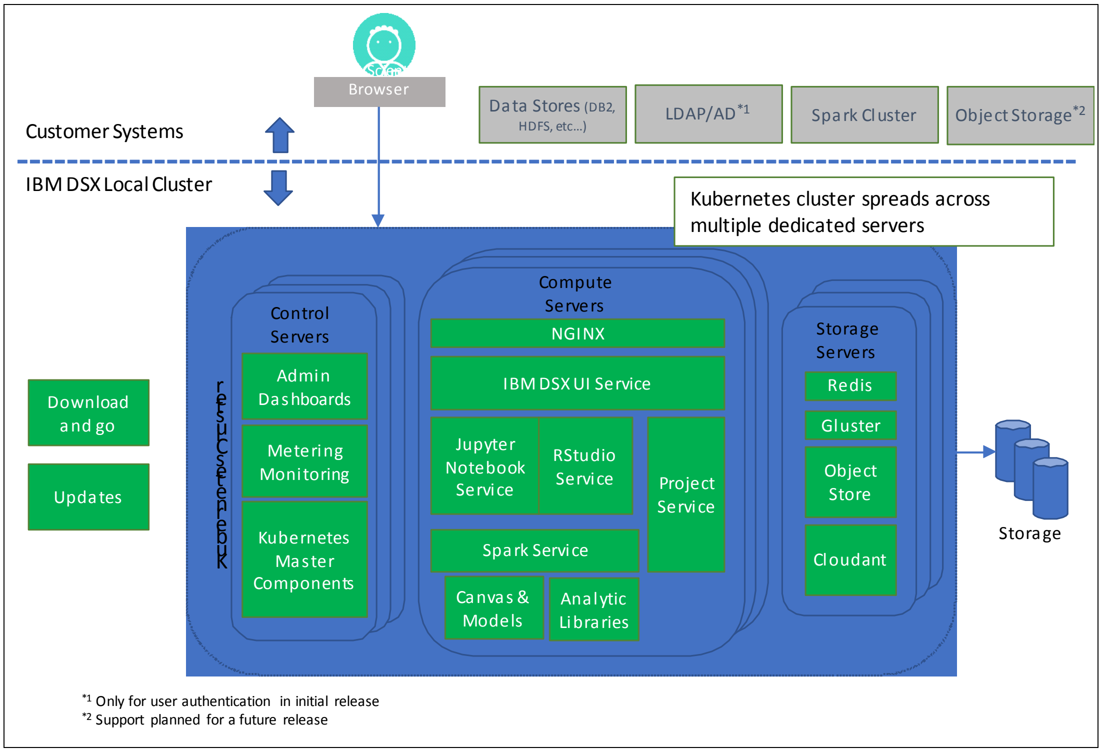


Figure 4-8 IBM Data Science Experience Local cluster architecture

The control servers run the services that Kubernetes needs to maintain the cluster and IBM DSX administration services.

The compute servers run the compute-intensive services, such as Spark, notebooks, and RStudio. This is also where models and pipelines run.

The storage servers manage the object and file system storage.

IBM DSX Local can also connect to an existing data set by creating named data connections. Data connections can be shared with other collaborators in a project, and can be used in notebooks to create DataFrames and RDDs.

For more information, see IBM Knowledge Center.

4.6.5 IBM Data Science Experience Local On Power Systems

The recommended minimum production environment is a three-node cluster with S822LC for Big Data servers. Each node runs all three components: Control, Compute, and Storage for HA. This configuration can accommodate up to 10 concurrent users. If a graphical processing unit (GPU) is required, add more S822LC for HPC POWER8 servers as compute nodes.

Figure 4-9 shows the IBM DSX Local on Power Systems servers design and configuration. If more capacity is required, add more nodes based on the type of the capacity, such as compute nodes or storage nodes.

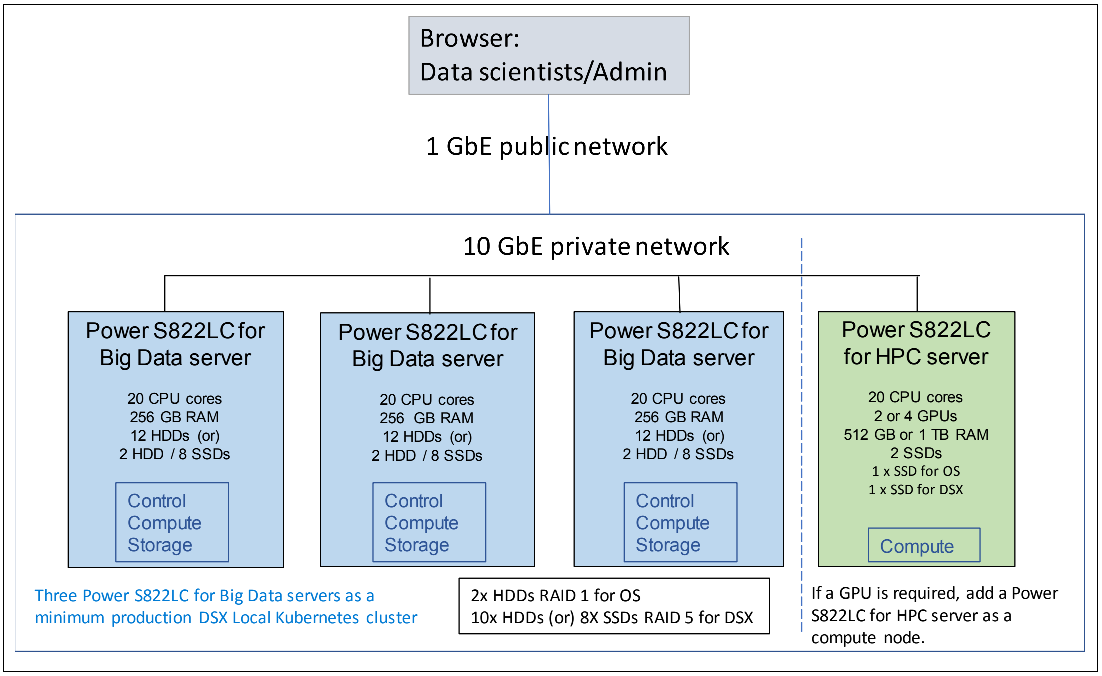


Figure 4-9 IBM Data Science Experience Local on Power Systems design

4.7 IBM Spectrum Conductor for Spark workload

IBM Spectrum Conductor is an on-premises platform for managing containerized applications that is based on the container orchestrator Kubernetes. Through the implementation of a robust resource manager (Apache Mesos), IBM Spectrum Conductor can manage resources for hybrid environments that contain both containerized and non-containerized applications, and a mixed hardware architecture.

IBM Spectrum Conductor is an integrated application and data-optimized platform that enables organizations to achieve up to 60% faster results with new generation, cloud-native applications and open source frameworks. The product achieves these results by efficiently analyzing, accessing, and protecting data.

Note: IBM Spectrum Conductor is compatible with IBM Spectrum Computing, complements IBM Spectrum Storage™ offerings, and is part of the software-defined infrastructure portfolio.

The STAC REPORT demonstrates the IBM Spectrum Conductor efficiency, and evaluates leading resource managers running Spark workloads in a multi-user environment.

4.7.1 Why you should use IBM Spectrum Conductor

Organizations of all types are realizing that data is their most valuable business asset, and that extracting full value from that data provides a critical competitive advantage. However, with today's massive data volumes continuing to grow, organizations are looking for a complete solution that answers the following questions:

- How do I accelerate business insights from all data by using next generation applications and frameworks such as Spark and MongoDB?
- How do I store and access big data cost-effectively and not in disconnected silos?
- How do I best manage that data throughout its lifecycle while ensuring maximum security, reliability, and availability?

IBM Spectrum Conductor helps organizations answer these questions in the most cost-efficient way possible, by transforming their infrastructure into a tightly integrated data and application-optimized platform for next generation, cloud-native applications.

IBM Spectrum Conductor enables organizations to adopt new generation cloud-native applications and open source frameworks efficiently, effectively, and confidently. Resource and data sharing can be increased without compromising availability or security. Unlike open source HDFS, IBM Spectrum Conductor is POSIX-compliant and provides significant storage efficiencies compared with HDFS.

Remember: IBM Spectrum Conductor also supports HDFS for users who prefer that option.

Users can share and manage the data over many applications:

- Data volumes are created on demand and owned by lines of business (LOBs) or individuals.
- Data is shared between users applications and seen as a shared file system.
- The wanted data volumes are connected to Docker containers and explain what users can do with it.

When a cluster is running, it is important to be able to monitor and maintain the system. IBM Spectrum Conductor helps simplify management with integrated cluster and application management control tools. Administrators can see where the application services are running, monitor the status of services, and see how resources are being allocated to meet service-level objectives (SLOs).

IBM Spectrum Conductor achieves these goals in the most cost-efficient way possible through three core capabilities:

- Analyze: A workload and data-aware platform increases usage of existing hardware resources and speeds analysis.
- Access: Policy-driven data placement with global access enables organizations to tackle all facets of storing data and sharing resources for distributed teams or data centers.
- Protect: Enterprise-class features, such as data encryption, automatic failover, and seamless system recovery, provide enhanced resilience and protection.

4.7.2 IBM Spectrum Conductor with Spark

IBM Spectrum Conductor with Spark is a complete enterprise-grade, multi-tenant solution for Apache Spark. It implements the concept of IBM Spectrum Conductor to address the requirements of users who need to adopt the Apache Spark technology, and to integrate it into their environment. It enhances performance and scale by eliminating resource silos that are tied to separate Apache Spark implementations.

Apache Spark in the enterprise

Apache Spark is an open source cluster computing framework for large-scale data processing. Like Hadoop MapReduce, Spark provides parallel distributed processing, fault tolerance on commodity hardware, and scalability. With its in-memory computing capabilities, analytic applications can run up to 100 times faster on Apache Spark compared to other technologies on the market today.

Apache Spark is highly versatile, and known for its ease of use in creating algorithms that harness insight from complex data. In addition to its ease of use, this framework covers a wide range of workloads through its different modules:

- Interactive queries through Spark SQL
- Streaming data, with Spark Streaming
- IBM ML, with the MLib module
- Graph processing with GraphX

Applications can be built by using simple APIs for Scala, Python, and Java:

- Batch applications that use the Hadoop MapReduce compute model.
- Iterative algorithms that build upon each other.
- Interactive queries and data manipulation through *notebooks* (web-based interfaces).

Spark runs on Hadoop clusters, such as Hadoop YARN or Apache Mesos, or even as a stand-alone component with its own scheduler.

IBM Spectrum Conductor with Spark advantages

Apache Spark, thanks to its speed, ease of use, and rich environment, is increasingly being adopted in enterprises as the main platform for analytics workloads. More applications are being developed based on Apache Spark, but with different Spark versions, that use different tools and for different users. These heterogeneous environments become a challenge for the IT departments of today's enterprise.

The easiest way to handle these challenges is to set up isolated clusters for different LOBs, but this is expensive and inefficient in a modern IT infrastructure. IT demands a software product to provide Spark multitenancy on a shared physical cluster.

Organizations that are looking to deploy an Apache Spark environment should consider a purpose-built version, such as IBM Spectrum Conductor with Spark. It enables you to deploy Apache Spark efficiently and effectively, supporting multiple deployments of Spark, increasing performance and scalability, maximizing usage of resources, and eliminating silos of resources that otherwise are tied to separate Spark implementations.

As an end-to-end solution, IBM Spectrum Conductor with Spark provides the following benefits:

- Provides multitenancy through Spark instance groups, which are similar to the Spark notion of tenants.
- Improves performance and efficiency through granular and dynamic resource allocation for Spark instance groups sharing a resource pool.

- Simplifies management by providing a consolidated framework for Apache Spark deployment, monitoring, and reporting.
- Provides high-performance shared storage through IBM Spectrum Scale, a high-performance solution for storage management.
- Integrates with Docker containers, enabling you to run Spark instance groups in a Docker environment.

IBM Spectrum Conductor with Spark provides flexible Spark support by enabling multiple Spark versions to coexist in the cluster. You can run multiple instances of Spark, including different Spark versions, in a shared environment. It also enables browser-based data analytics through notebooks, either through the built-in Apache Zeppelin and Jupyter Notebooks or through other third-party notebooks that you can integrate.

Understanding the components of IBM Spectrum Conductor with Spark

The IBM Spectrum Conductor with Spark framework consists of the following key components: a Spark distribution, the resource orchestrator, and the application service controller daemon (ascd). It also integrates Elastic Stack.

- Spark distribution

  An IBM Spark distribution that contains the core components of Apache Spark: Spark Core, Spark SQL, Spark Streaming, IBM ML Library, and GraphX. It also prepackages extra files for IBM Spectrum Conductor with Spark capabilities.

- Notebooks

  An integrated notebook framework, wherein a notebook's graphical user interface is used for data manipulation and visualization. IBM Spectrum Conductor with Spark bundles the Apache Zeppelin and Jupyter notebooks; you can optionally integrate other third-party notebooks. For more information, see IBM Knowledge Center.

- Resource orchestrator

  The resource orchestrator, also known as the enterprise grid orchestrator (EGO), manages the supply and distribution of resources, making them available to applications. It interfaces with Spark Core through the Spark on EGO plug-in to provide resource provisioning, remote execution, HA, and business continuity.

  The resource orchestrator provides cluster management tools and is similar to a distributed operating system that pools a heterogeneous set of nodes into a large distributed virtual computer. It enables multiple types of workloads to share resources efficiently, driving up utilization while ensuring SLAs for individual applications.

- Application service controller daemon (ascd)

  The ascd manages the lifecycle of a Spark instance group. Registered as an EGO service, the ascd acts as a REST API layer and enables Spark instance groups to be described as a set of inter-related long-running services in a single specification. The services that are associated with the Spark instance groups can then be deployed and scaled.

  Internally, the Spark instance group is a special implementation of an ascd application instance. You can create application instances for long-running services that support your Spark applications, including cloud-native ones such as MongoDB and Cassandra.

- Elastic Stack

  The Elastic Stack (Elasticsearch, Logstash, and Beats) is integrated within IBM Spectrum Conductor with Spark for data collection and visualization. Registered as system services, the Elastic Stack integration enables you to search, analyze, and visualize Spark application data for efficient monitoring.

- IBM Spectrum Scale

  If you are licensed to use IBM Spectrum Scale, you can optionally use IBM Spectrum Scale capabilities. IBM Spectrum Scale is a cluster file system that provides concurrent access to a single file system or a set of file systems from multiple hosts. It enables high-performance access to this common set of data to support a scale-out solution and provide a HA platform.

  As storage technology, IBM Spectrum Scale is a space-efficient alternative to the HDFS.

Figure 4-10 illustrates an end-to-end enterprise class solution that uses Spark.

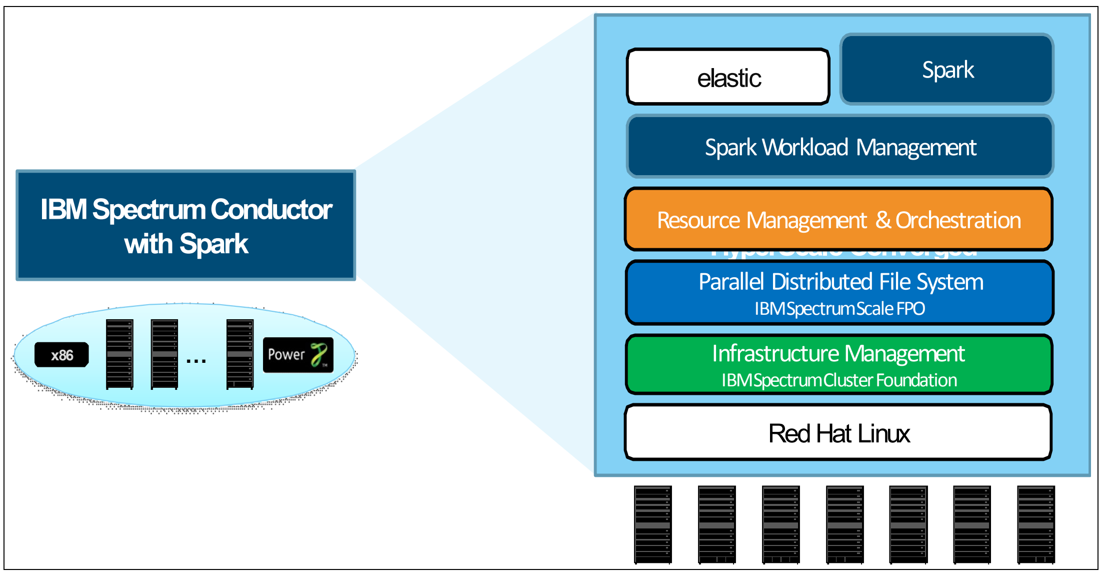


Figure 4-10 IBM Spectrum Conductor with Spark: End-to-end enterprise class solution

4.7.3 Integration with Apache Spark

Apache Spark applications run as independent sets of processes on a cluster that is coordinated by the Spark Context object in the driver program. The Spark Context object can connect to the cluster manager (either the Spark stand-alone cluster manager, Apache Mesos, Hadoop YARN, or EGO in IBM Spectrum Conductor with Spark) through the Session Scheduler, which allocates resources across applications, as shown in Figure 4-11.

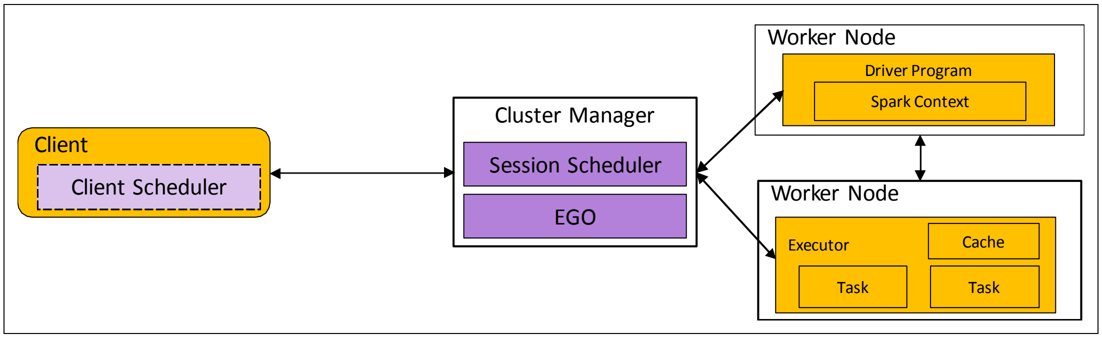


Figure 4-11 Spark integration with IBM Spectrum Computing resource manager

In IBM Spectrum Conductor with Spark, the resource orchestrator (EGO) acts as the cluster manager with Apache Spark, enabling Spark applications to benefit from resource sharing across the cluster, which provides the following benefits:

- Fine-grained scheduling: Apache Spark uses a coarse-grained or *fine-grained* resource scheduling policy. The Spark application requests a static number of resources and holds them in its lifecycle, which means each application gets more or fewer resources as it scales up and down. Based on the fine-grained scheduling policy, applications share resources at a fine granularity, especially when many applications are running in a cluster concurrently.
- Resource sharing and reclaim: Tenants (known as a *Spark instance group*) share resources that are managed in consumers and resource groups by the resource orchestrator. Therefore, some tenants can acquire more resources than the resource distribution definition by borrowing them from other applications. When the lender needs more resources, it reclaims these borrowed resources from the borrower. This can keep the cluster in high usage status, and maintain consistency between all tenants.
- Multiple resource scheduling policies (first-in first-out (FIFO)/Fairshare) for each tenant: All tenants partition the resources based on the resource allocation plan.
- Multi-tenant scheduling with the Session scheduler: A consumer can run many applications concurrently, and the session scheduler can schedule resources between applications that belong to one consumer.

References

Here are some useful references about the topics in this section:

- IBM developerWorks community about IBM Spectrum Conductor with Spark
- Available samples and tutorials to help you get started with IBM Spectrum Conductor with Spark

4.8 Tools that are available for data integration

An important step in building EDW on Hadoop is to extract, transform, and move data from various disparate sources and load it into Hadoop data storage; in the traditional EDW world, this is called ETL or ELT. For traditional EDW offloading, it involves extracting and moving data from OLTP systems to Hadoop; after data is processed (cleansed and transformed) in Hadoop, data is then loaded into existing EDW. There are different kinds of technologies and tools that are available that can help the development of this data integration process.

We categorize these data integration tools into two categories: Apache open source technology tools and vendor tools. Many vendor tools are based on Apache open source technologies and provide added functions.

4.8.1 Apache open source technology tools

Apache has many projects, and the projects are evolving with both technology and business requirements. Each project serves its own purpose. Because of the number of tools and overlapping of some of the functions among the tools, it can be confusing when choosing the correct tool. The technologies that are listed in the following sections are commonly used for building EDW on Hadoop in the ETL (data movement and data flow) process. A brief description of each component is provided as a quick reference.

For more information, see the relevant Apache websites. These components are included in either HDP or the Hortonworks DataFlow (HDF) package for Power Systems.

Apache Flume

Apache Flume is a distributed system for collecting, aggregating, and moving large amounts of data, such as log data or stream events (simple event processing) from multiple sources into HDFS.

Apache Sqoop

Apache Sqoop is an effective tool for transferring bulk data between Hadoop and relational databases.

Apache NiFi

Apache NiFi is an integrated data logistic management tool. It is a powerful and reliable system to process and distribute data that provides data flow automation between systems, such as moving data from OLTP systems to Hadoop. It provides a web-based user interface for designing and monitoring data routing, transformation, and system mediation logic.

MiNiFi

MiNiFi is a complementary data collection technology that supplements the core tenets of NiFi in data flow management. It focuses on the collection of data at the source of its creation.

Apache Kafka

Apache Kafka is a distributed publish/subscribe, real-time messaging platform. It provides a unified, high-throughput, low-latency, durable, and fault-tolerant platform for handling real-time data flow. Its storage layer provides a scalable publish/subscribe message queue architecture.

4.8.2 Vendor tools

Vendor tools usually provide functions for a more focused area, either targeting certain vendor platforms from the source or target, or focusing on certain areas, such as real-time data flow management, data connectivity, or data replication. They are supported by client implementation experiences. This section provides a quick overview of the tools that are available on a Power platform or that can be run on a supported platform and connect to HDP on Power Systems as needed.

Attunity

Attunity is a data integration provider that provides a family of products and solutions, such as Attunity Visibility, Replicate, and Compose:

- Visibility provides intelligent data management with data usage and workload analytics, and it enables in-depth, multi-dimensional analysis of data warehouse and Hadoop environments. Use Visibility to collect and analyze existing data warehouse data usage and workload, to identify hot data and cold data, and to move cold data off an existing data warehouse to a Hadoop environment.
- Replicate provides data replication, synchronization, ingestion, and streaming across a wide range of heterogeneous databases, data warehouses, and big data platforms. The intuitive GUI provides a quick and easy way to set up data replication between a source and a target. This tool is helpful in setting up the continuous replication and synchronization between OLTP data sources and data ware house.
- Compose provides a guided process for data warehouse design, modeling, development, and testing. It generates ETL code, and applies updates to the data warehouse quickly.

Hortonworks DataFlow

HDF provides an end-to-end platform that collects, curates, analyzes, and acts on data in real time, with a drag-and-drop GUI. HDF is an integrated solution that includes Apache NiFi/MiniFi, Apache Kafka, Apache Storm, and Druid. It has the following features:

- HDF Data Flow Management integrates with NiFi/MiniFi, collects and ingests data, and manages data in full flight through a visual control pane, where you can adjust sources, join and split streams, and prioritize data flow.
- HDF Stream Processing is integrated with Apache Storm and Kafka. It provides immediate and continuous insight for data. Hortonworks Streaming Analytics Manager is an open source tool that is used to design, develop, deploy, and manage streaming analytics applications by using a drag GUI.
- HDF Enterprise Services are based on Apache Ambari, Ranger, and Hortonworks Schema Registry. They provide security and data governance capabilities to manage the HDF and HDP infrastructures through a comprehensive management GUI for provisioning, monitoring, and governance.

Oracle Instant Client

Oracle Instant Client is a data connectivity tool that enables applications to connect to a local or remote Oracle Database for development and production deployment. The Instant Client libraries provide the necessary network connectivity, and basic and high-end data features to make full use of Oracle Database.

Paxata

Paxata is in the business of empowering all business consumers to intelligently transform raw data into ready information, instantly, with an enterprise-grade, scalable, self-service, and intelligent platform. Paxata Adaptive Information Platform triangulates data into an information fabric from any source, any cloud or enterprise, for any business to create trusted information. With Paxata, business consumers use clicks, not code, to achieve results in minutes, not months.

By using Paxata, a business' data sources can be assembled into an information fabric by using the Adaptive Information Platform to deliver information at the speed of thought to all of their business consumers. The ability to do this task gives these businesses a unique competitive advantage in their markets.

Paxata Self-Service Data Prep Application provides an intuitive GUI for business users to rapidly turn raw data into data worthy of analytics. After data is moved into the Hadoop landing zone or data lake, business users can discover, explore, clean, blend, enrich, and shape data through an intuitive user experience to create analytics ready data sets (AnswerSets). The entire lifecycle is fully governed, secure, and easily automated, which improves productivity and aids policy conformance and compliance.

Progress DataDirect

Progress DataDirect is a data connectivity tool that connects data between any data sources to better integrate data across relational, big data and cloud databases. It provides high-performing ODBC, JDBC, ADO.NET, and OData connectivity for any data source on any platform.

SAP

If your existing data warehouse is on SAP HANA Business Warehouse, then the following products are worth of looking at. These products are all supported on PowerLinux.

- Smart Data Access is a native component of the SAP HANA database. It is a data connection tool that can access data that is stored in remote data sources. Using SDA, SAP HANA can create "virtual table" mapping to tables located in remote data sources, enabling SAP HANA to access the remote data directly by accessing the "virtual table".
- Smart Data Integration is also a native component of the SAP HANA database. It provides the capability to connect to other data sources, transforming the data, and replicating the remote data into SAP HANA.
- Spark Controller enables SAP HANA to access Hadoop data through the SQL interface. It also enables HANA to read data from Hadoop/Spark, and write data to Hadoop/Spark. Spark Controller primarily works with Spark SQL and connects to an existing Hive metastore.

Veristorm vStorm Enterprise

Veristorm vStorm Enterprise moves data from many heterogeneous data sources, including mainframe data sources to Hadoop platforms such as HDP. For mainframe data sources such as VSAM and QSAM, data is extracted from the mainframe in binary form and streamed to the target system, such as HDP. vStorm Enterprise uses streaming technology, which enables near real-time access to data while eliminating staging.

Related publications

The publications that are listed in this section are considered suitable for a more detailed description of the topics that are covered in this paper.

IBM Redbooks

The following IBM Redbooks publications provide more information about the topics in this document. Some publications that are referenced in this list might be available in softcopy only.

- *Hortonworks Data Platform with IBM Spectrum Scale: Reference Guide for Building an Integrated Solution*, REDP-5448
- *IBM Power System S822LC for Big Data: Technical Overview and Introduction*, REDP-5407
- *IBM Spectrum Conductor and IBM Spectrum Conductor with Spark*, REDP-5379
- *IBM Spectrum Scale: Big Data and Analytics Solution*, REDP-5397
- *Implementing an Optimized Analytics Solution on IBM Power Systems*, SG24-8291
- *Introduction Guide to the IBM Elastic Storage Server*, REDP-5253
- *Performance Optimization and Tuning Techniques for IBM Power Systems Processors Including IBM POWER8*, SG24-8171

  Chapter 6, “Linux”, focuses on gathering the correct technical information, and laying out simple guidance for optimizing code performance on IBM POWER8 processor-based systems that run the Linux operating systems.

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

Hadoop resources

- Apache Hadoop projects:

  http://hadoop.apache.org/
- Cassandra: A scalable multi-master database with no single points of failure:

  http://cassandra.apache.org/
- Hadoop and IBM projects

  https://www.ibm.com/analytics/hadoop
- Hadoop trials:

  https://www.ibm.com/analytics/us/en/technology/hadoop/hadoop-trials.html
- *HDP on IBM Power - Reference Architecture and Design (online)*:

  https://www.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=POL03270USEN&
- *HDP on IBM Power Systems with IBM Elastic Storage Server - Reference Architecture and Design (online)*:

  https://www.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=POL03281USEN&

- Hive: A data warehouse infrastructure that provides data summarization and ad hoc querying:

  http://hive.apache.org/
- Hortonworks:

  http://www.hortonworks.com
- Installation of HDP 2.6 on IBM Power Systems (step-by-step):

  https://community.hortonworks.com/articles/101185/installation-of-hdp-26-on-ibm-power-systems.html
- ISV solution infrastructure for Hortonworks on IBM Power Systems:

  https://www.ibm.com/developerworks/library/l-isv-solution-hortonworks/
- Manual for HDP installation:

  https://docs.hortonworks.com/HDPDocuments/HDP2/HDP-2.6.2/bk_command-line-installation/bk_command-line-installation.pdf
- Pig: A high-level data flow language and execution framework for parallel computation:

  http://pig.apache.org/
- Spark: A fast and general compute engine for Hadoop data:

  http://spark.apache.org/

IBM Elastic Storage Server and IBM Spectrum Scale resources

- Deploy a Big Data Solution on IBM Spectrum Scale:

  https://ibm.biz/BdiVKs
- IBM Elastic Storage Server:

  https://www.ibm.com/us-en/marketplace/ibm-elastic-storage-server
- IBM Spectrum Scale HDFS Transparency main page:

  https://ibm.biz/BdrY82
- *IBM Spectrum Scale FPO Deployment And Maintain Guide*:

  https://ibm.biz/BdHKQF
- *HDP on IBM Power with IBM Elastic Storage Server - Reference Architecture and Design* (online):

  https://www.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=POL03281USEN&
- *HDP on IBM Power with IBM Elastic Storage Server Reference Architecture and Design* (download):

  https://www.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=POL03281USEN&dd=yes&
- Overview of IBM Spectrum Scale:

  https://www.ibm.com/support/knowledgecenter/en/STXKQY_4.2.3/com.ibm.spectrum.scale.v4r23.doc/bl1in_IntroducingIBMSpectrumScale.htm

IBM Power Systems resources

- *CAPI for POWER8 Systems* white paper:

  https://www-304.ibm.com/webapp/set2/sas/f/capi/CAPI_POWER8.pdf

- IBM Linux on Power online resources:
  - IBM Knowledge Center: Technical information about Linux on Power Systems servers:

    https://www.ibm.com/support/knowledgecenter/linuxonibm/liabx/lcon_System_p5.htm

  - The Developer portal for Linux on Power is geared more towards developers. The portal is an excellent place to start because it has pointers to all sorts of information.

    https://developer.ibm.com/linuxonpower/

  - IBM developerWorks Technical resources for Linux developers and IT professionals is a good repository of articles and tutorials:

    https://www.ibm.com/developerworks/learn/linux/index.html

  - The IBM Linux on Power Tools repository enables the use of standard Linux package management tools (such as yum and zypper) to provide easy access to IBM recommended tools:
    - IBM Linux on Power hardware diagnostic aids and productivity tools
    - IBM Software Development Toolkit for Linux on Power servers
    - IBM Advance Toolchain for Linux on Power Systems servers

    The IBM Linux on Power Tools repository is found at the following website:

    http://www.ibm.com/support/customercare/sas/f/lopdiags/yum.html

  - IBM Marketplace - Linux on IBM Power Systems:

    https://www.ibm.com/power/operating-systems/linux

- IBM Portal for OpenPOWER:

  https://www.ibm.com/systems/power/openpower/welcome.xhtml

- IBM Power Systems:

  https://www.ibm.com/power

- OpenCAPI consortium:

  http://opencapi.org/

- Power Systems reliability:

  http://itic-corp.com/blog/2017/06/ibm-lenovo-servers-deliver-top-reliability-cisco-ucs-hpe-integrity-gain/

- Supported features for Linux on Power Systems:
  - Supported features for Linux on Power Systems with Red Hat Enterprise Linux and PowerVM virtualization. Use this link to find the Linux distribution version that supports each system feature on Linux on Power Systems. Feature support is shown for Red Hat Enterprise Linux.

    https://www.ibm.com/support/knowledgecenter/linuxonibm/liaam/supportedfeatures_rhel.htm
  - Supported features for Linux on Power Systems with SUSE Linux Enterprise Server and PowerVM virtualization. Use this link to find the Linux distribution version that supports each system feature on Linux on Power Systems running PowerVM virtualization. Feature support is shown for SUSE Linux Enterprise Server.

    https://www.ibm.com/support/knowledgecenter/linuxonibm/liaam/supportedfeatures_sles.htm
  - Supported features for Linux on Power Systems with Ubuntu and PowerVM virtualization. Use these charts to find the Linux distribution version that supports each system feature on Linux on Power Systems. Feature support is shown for the Ubuntu Linux distribution.

    https://www.ibm.com/support/knowledgecenter/linuxonibm/liaam/supportedfeatures_ubuntu.htm
- Supported Linux distributions for POWER8

  The recommended Linux distribution for a particular server is always the current level distribution that is optimized for the server. The listed distributions are the operating system versions that are supported for the specific hardware. For more information about the product lifecycles for the Linux distributions, see the distribution's support site.
  - SUSE Linux Enterprise Server: SUSE Product Support Lifecycle:

    https://www.suse.com/lifecycle/
  - Red Hat Enterprise Linux: Red Hat Enterprise Linux Life Cycle:

    https://access.redhat.com/site/support/policy/updates/errata/
  - Ubuntu: Ubuntu Release Life Cycle:

    http://www.ubuntu.com/info/release-end-of-life
  - Up-to-date Linux distributions that are optimized for POWER8 processor-based servers running Linux:

    https://www.ibm.com/support/knowledgecenter/linuxonibm/liaam/liaamdistros.htm

SQL engine on Hadoop: IBM Big SQL resources

- Best practices in IBM Hadoop Big SQL developerWorks website:

  https://developer.ibm.com/hadoop/category/bigsql/
- Customer recommendations for using IBM Big SQL:

  https://youtu.be/_cwYGK_oPe8
- IBM Big SQL V5.0.1 IBM Knowledge Center:

  https://www.ibm.com/support/knowledgecenter/SSCRJT_5.0.1/com.ibm.swg.im.bigsql.welcome.doc/doc/welcome.html
- IBM Open Platform with Apache Hadoop:

  https://www.ibm.com/analytics/us/en/technology/hadoop/

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

REDP-5476-00

ISBN 0738456608

Printed in U.S.A.