Data Integration in the Big Data World Using IBM InfoSphere Information Server

IBM Redbooks Solution Guide

An Apache Hadoop infrastructure can reduce the costs of storing and processing large data volumes. By investing in Hadoop, organizations can slash IT spending while creating significant return on investment (ROI) by using enhanced analytics and reporting that were not previously possible. However, the Hadoop infrastructure alone might not deliver the promised reduced cost or the increased ROI that flows from better reporting and analytics.

Achieving improved ROI is dependent upon how effectively data in the Hadoop environment is handled, prepared, and managed. Critical data-related tasks that must be managed effectively for success with Hadoop include data movement, data transformation and integration, data cleansing, data governance, data security, data privacy, and data analytics and reports.

Many organizations are considering implementing a *data* lake solution. This is a set of one or more data repositories that are created to support data discovery, analytics, ad hoc investigations, and reporting. Without proper management and governance, a data lake can quickly become a data swamp. The data swamp is overwhelming and unsafe to use because no one is sure where the data came from, how reliable it is, and how it should be protected. IBM® proposes an enhanced data lake solution that is built with management, affordability, and governance at its core. This solution is known as a *data reservoir* . A data reservoir provides the correct information about the sources of data that are available in the lake and their usefulness in supporting users who are investigating, reporting, and analyzing events and relationships in the reservoir.

IBM InfoSphere® Information Server provides an integrated set of tools that are built to handle the extreme throughput and governance that are required by today’s demanding business enterprises. These tools efficiently move data and keep track of the information about what was moved, when, and what was done to the data during the process. On top of this set of tools sits the business metadata, which is the entry point for the business to find things and the governance metadata that dictates who can use the data and controls its destiny. Using InfoSphere Information Server gives organization the freedom and the flexibility to run big data integration workloads when it makes sense for them to do so.

This IBM Redbooks® Solution Guide addresses the practical realities of managing the data integration tasks that are required for success with Apache Hadoop. Managing these tasks effectively in the Hadoop environment is one critical step in supporting a data reservoir rather than creating a data swamp (see Figure 1).

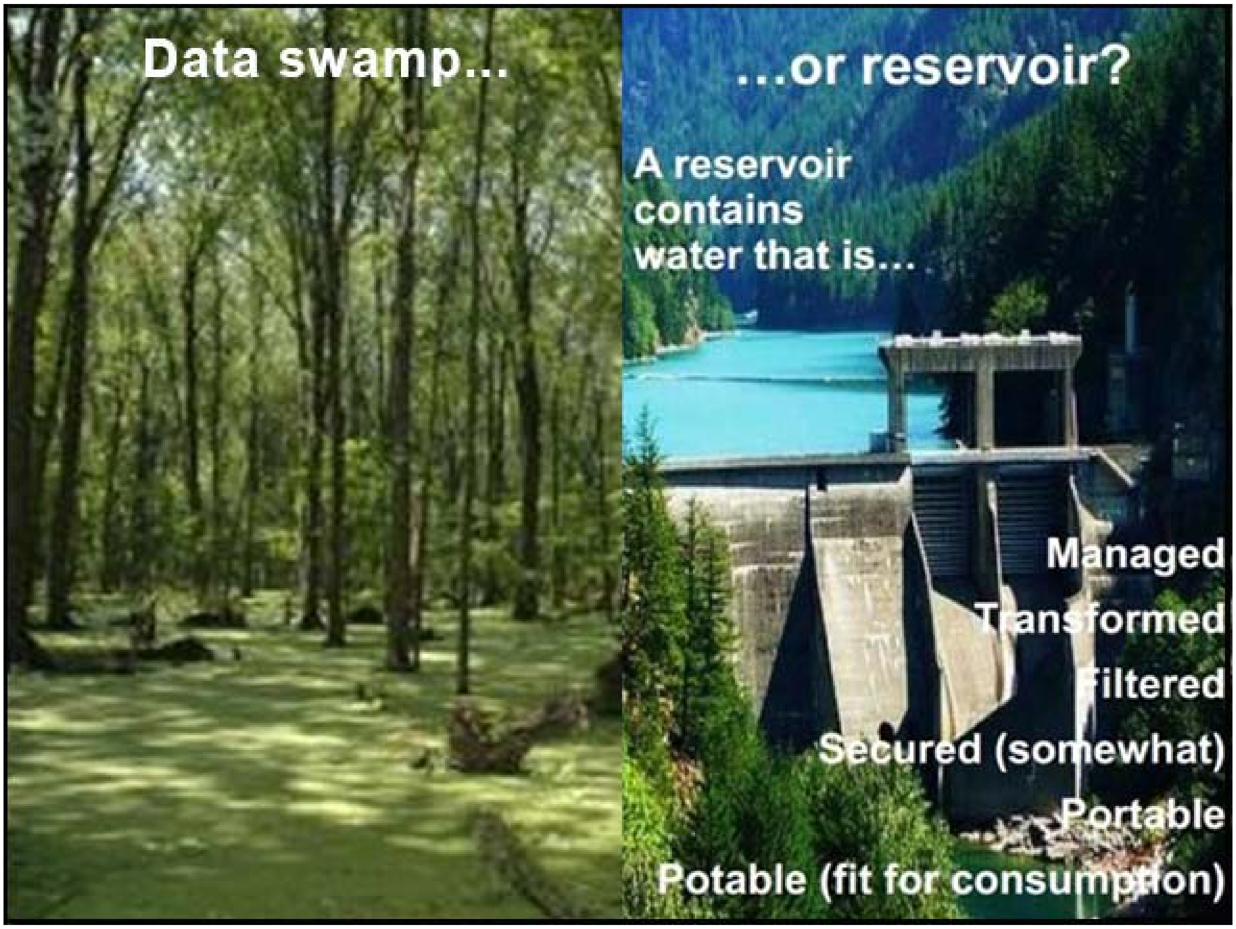


Figure 1. Data reservoirs are a critical step in effective use of Hadoop

Did you know?

In the past, there was a well-known marketing mantra called “The 3 Vs” that addressed the volume, velocity, and variety aspects of data in the new “Internet of Things” (see Figure 2).

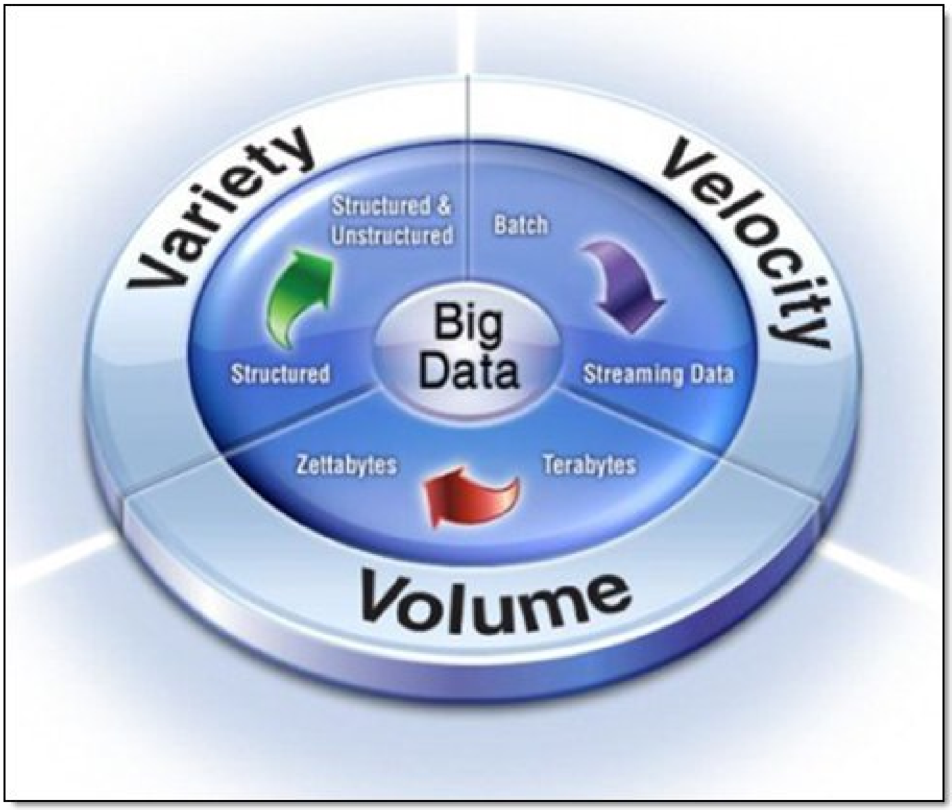


Figure 2. The 3 Vs of volume, velocity, and variety

Then, people realized that data quality is still relevant in this new world, so many articles and presentations introduced a fourth V, *veracity* . Hadoop can support all of these capabilities, but it requires a great deal of complex programming. There have also been alternative technologies available that have provided these capabilities without requiring a paradigm shift. Parallel processing technology has been available both inside and outside of databases for many years, and these technologies have handled terabytes of information. Handling the velocity or “data in motion” has also been available through streaming technology.

If the 4 Vs aren’t enough alliteration, you can also consider the 3 As and the 3 Ss. In fact, serious business organizations demand all of them in data processing environments, including Hadoop. The emergence of the 3 As of availability, accessibility, and accountability along with the 3 Ss of sovereignty (restricting data location), security, and service level agreement (SLA) for Hadoop environments is healthy. This demand demonstrates that Hadoop is being seriously considered for critical business workloads. It is these attributes that are driving the governance focus and the desire to transform the data swamp into the data reservoir.

Business value

The rapid emergence of Hadoop is revolutionizing how organizations take in, manage, transform, store, and analyze big data. Successful Hadoop projects can deliver business value and ROI through pure cost reduction. Also, increased revenue and profitability can be realized from deeper analysis of large data volumes that organizations simply could not afford to store and process in the past.

Effective big data integration and governance is critical for trusted big data. Without it, you get “garbage in, garbage out.” Unless organizations address these critical areas, they produce insights or transformative results that incomplete are significantly less accurate.

The emergence of the Hadoop infrastructure is making it possible for organizations to eliminate significant business and technical limitations of data integration processes and practices. Some of these processes and practices have accumulated over years, even decades. Over time, many organizations have continued to rely on hand coding of data integration logic rather than using commercial data integration software. In addition, most of the commercially available extraction, transformation, and loading (ETL) tools and data integration software platforms were never built on shared nothing, massively parallel software architectures. As storage costs have decreased steadily and data volumes have grown, organizations that rely on hand coding or non-scalable ETL tools have been forced to push 100% of their big data integration workloads into a parallel database (most often, the enterprise data warehouse).

Note: This guide refers to both ELT and ETL, based on their appropriate use.

Under certain conditions, running big extract, load, and transform (ELT) workloads in the parallel database is appropriate. However, when an organization is forced to run 100% of big ELT workloads in the parallel database without a distinct massively scalable data integration platform, many negative and unforeseen consequences emerge and build up over time:

- Data warehouse database computing and storage hardware is expensive, often costing 10X or more than commodity Linux or Intel systems.

- Hand coding is 10X more costly than using data integration software for building and maintaining ETL or ELT workloads. Hand coding projects also take longer to complete (adding new data sources to a warehouse can take months rather than days or hours). Finally, reliance on hand coding makes it very difficult, even nearly impossible, to establish data governance across the enterprise.

- Data integration workloads can consume a growing percentage of the database processing capacity. ELT workloads can consume 40 - 80% of the warehouse processing capacity, for example. This makes it difficult to meet SLAs for database queries. Rather than consuming excess warehouse processing capacity, ELT workloads can also influence capital investment decisions to add processing capacity.

- As data volumes increase, they add pressure on the ETL processing window. Any disruption or glitch in the process can encroach on the query processing window. That can affect throughput and wreak havoc on SLAs.

- It is often not possible to run more complex data integration logic in the database, such as customer name and address matching or consolidation. Consequently, data quality is often lower in the absence of more robust data integration processing.

- It becomes expensive and time-consuming to add new data sources to the enterprise data warehouse. Also, the warehouse does not grow as quickly as it is possible to do with massively scalable data integration software running on a commodity Linux or Intel grid.

100% reliance on running big ELT workloads in the parallel database can result in tens or even hundreds of millions of dollars of negative ROI over a period of years (higher costs, longer time to value, poor quality data, limited data governance). Data integration becomes a source of competitive disadvantage rather than competitive advantage under this scenario.

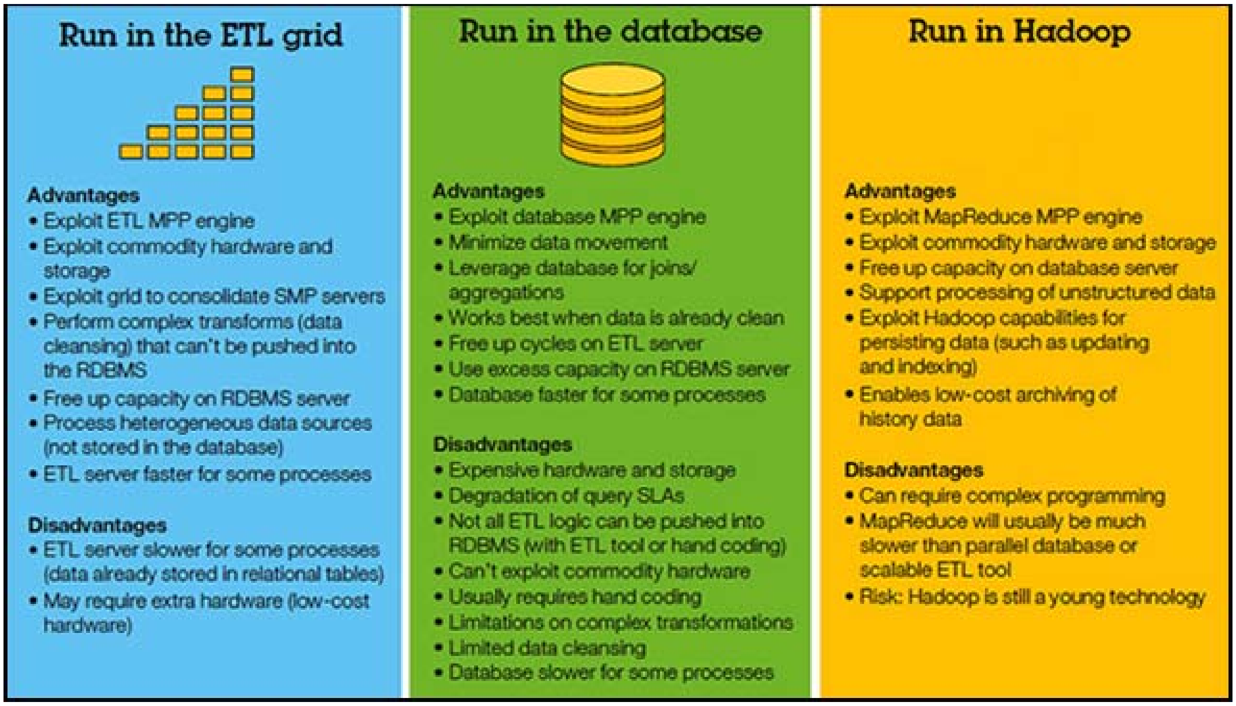
Historically, organizations that rely on a massively scalable data integration platform, such as the IBM InfoSphere Information Server, have not suffered the negative consequences that result from pushing 100% of big ELT workloads into the parallel database. These organizations can build a data integration job once and process massive data volumes wherever it makes the most sense (in the ETL grid, in the parallel database, or in Hadoop). They can also implement data governance effectively across the enterprise by using IBM data governance tools.

In such a pattern, Apache Hadoop infrastructure can drastically reduce the overall costs of storing and processing large data volumes (including data integration processing). In addition, Hadoop provides a real opportunity to eliminate many of the significant negative consequences for organizations that run 100% of their big data integration workloads in the parallel database.

Solution overview

Perception is not always reality, and one size does not really fit all. The reality is that not all of an organization’s big data integration workloads can be effectively run in the Hadoop environment. Organizations will need to continue running certain big data integration workloads outside of Hadoop. The emerging preferred practice for big data integration and governance is to rely on a massively scalable data integration and governance platform, such as the IBM InfoSphere Information Server. With solutions such as this, you can build a data integration job once and then have the freedom and flexibility to run the big data integration workload wherever it makes the most sense (in the ETL grid, in the parallel database, or in Hadoop). You can also manage data governance across these three environments.

As Figure 3 shows, each of these three environments offers advantages and disadvantages for running big data integration workloads. Big data integration requires a balanced approach that supports all three environments.



Run in the ETL grid

Advantages
- Exploit ETL MPP engine
- Exploit commodity hardware and storage
- Exploit grid to consolidate SMP servers
- Perform complex transforms (data cleansing) that can’t be pushed into the RDBMS
- Free up capacity on RDBMS server
- Process heterogeneous data sources (not stored in the database)
- ETL server faster for some processes

Disadvantages
- ETL server slower for some processes (data already stored in relational tables)
- May require extra hardware (low-cost hardware)

Run in the database

Advantages
- Exploit database MPP engine
- Minimize data movement
- Leverage database for joins/ aggregations
- Works best when data is already clean
- Free up cycles on ETL server
- Use excess capacity on RDBMS server
- Database faster for some processes

Disadvantages
- Expensive hardware and storage
- Degradation of query SLAs
- Not all ETL logic can be pushed into RDBMS (with ETL tool or hand coding)
- Can’t exploit commodity hardware
- Usually requires hand coding
- Limitations on complex transformations
- Limited data cleansing
- Database slower for some processes

Run in Hadoop

Advantages
- Exploit MapReduce MPP engine
- Exploit commodity hardware and storage
- Free up capacity on database server
- Support processing of unstructured data
- Exploit Hadoop capabilities for persisting data (such as updating and indexing)
- Enables low-cost archiving of history data

Disadvantages
- Can require complex programming
- MapReduce will usually be much slower than parallel database or scalable ETL tool
- Risk: Hadoop is still a young technology

Figure 3. The advantages and disadvantages of the three environments for big data integration workloads

The two primary components of Hadoop infrastructure include the Hadoop distributed file system for storing large files and the Hadoop distributed parallel processing framework, known as MapReduce. One common fallacy about big data integration is that you can combine any non-scalable ETL tool with MapReduce to produce a highly scalable big data integration platform. In reality, MapReduce was not designed for high-performance processing of massive data volumes but, instead, for finely grained fault tolerance. The IBM InfoSphere Information Server data integration platform is capable of processing typical data integration workloads 10 to 15 times faster than MapReduce. A second shortcoming of MapReduce for big data integration is that not all complex data integration logic can be pushed into MapReduce. Forcing this situation makes it necessary to engage in sophisticated programming algorithms to handle more complex logic or to limit data integration processing to only simple logic. These limitations of MapReduce suggest that you need a platform such as the IBM InfoSphere Information Server running in Hadoop without MapReduce to overcome the performance and functional limitations of MapReduce.

Solution architecture

So how do you determine whether you should run your data integration workloads in the ETL grid, in the parallel database, or in the Hadoop environment? There is no simple answer. Each environment offers advantages and disadvantages. The following sections provide guidelines for understanding when each environment is appropriate:

- Hardware and storage costs
- Parallel processing software architecture
- Developer skills
- Handling unstructured data
- Offloading ELT from the EDW to Hadoop
- Joins, aggregations, and sorts in Hadoop
- Hadoop and data collocation
- Information governance

Hardware and storage costs

The ETL grid and Hadoop environments can maximize the same commodity computing and storage components, so neither environment offers any cost advantage over the other. In contrast, the parallel database used for running big ETL workloads is usually the enterprise data warehouse (EDW). Big EDW environments usually maximize computing and storage hardware that is much more expensive (10 - 50 times more) than commodity computing and storage hardware. The EDW offers a clear cost disadvantage.

Parallel processing software architecture

The most scalable software architecture for processing big data integration workloads is the shared nothing, data flow architecture with data pipelining, and data partitioning across nodes. Most parallel databases support this software architecture and can provide highly scalable massively parallel processing (MPP) for data integration workloads (although there might be some performance degradation in Hadoop environments caused by landing the data to support the fine-grained recovery function). Another limitation of the parallel database is that you cannot express more complex data integration logic in SQL (data cleansing, for example).

The InfoSphere Information Server is a scalable data integration platform built on this software architecture. It can provide highly scalable MPP processing for data integration workloads in all three environments and can run many types of complex data integration logic that cannot be pushed into the parallel database or Hadoop MapReduce. For those integration tasks that can be done effectively, InfoSphere Information Server also supports pushing data integration logic into the parallel database and into Hadoop MapReduce.

Hadoop MapReduce is the original parallel processing framework for Hadoop. It was designed for finely grained fault tolerance, which comes at the expense of high throughput and performance. MapReduce is not built on shared nothing, massively parallel data pipelining, and partitioning architecture. InfoSphere Information Server processes most data integration workloads 10 - 15X faster than MapReduce because of its superior architecture. Apache Spark is an emerging parallel processing framework for Hadoop. Spark, coupled with YARN or Apache Mesos, overcomes the architectural limitations of MapReduce, but be aware that these are emerging technologies that require significant amounts of hand-coded programming for running data integration workloads.

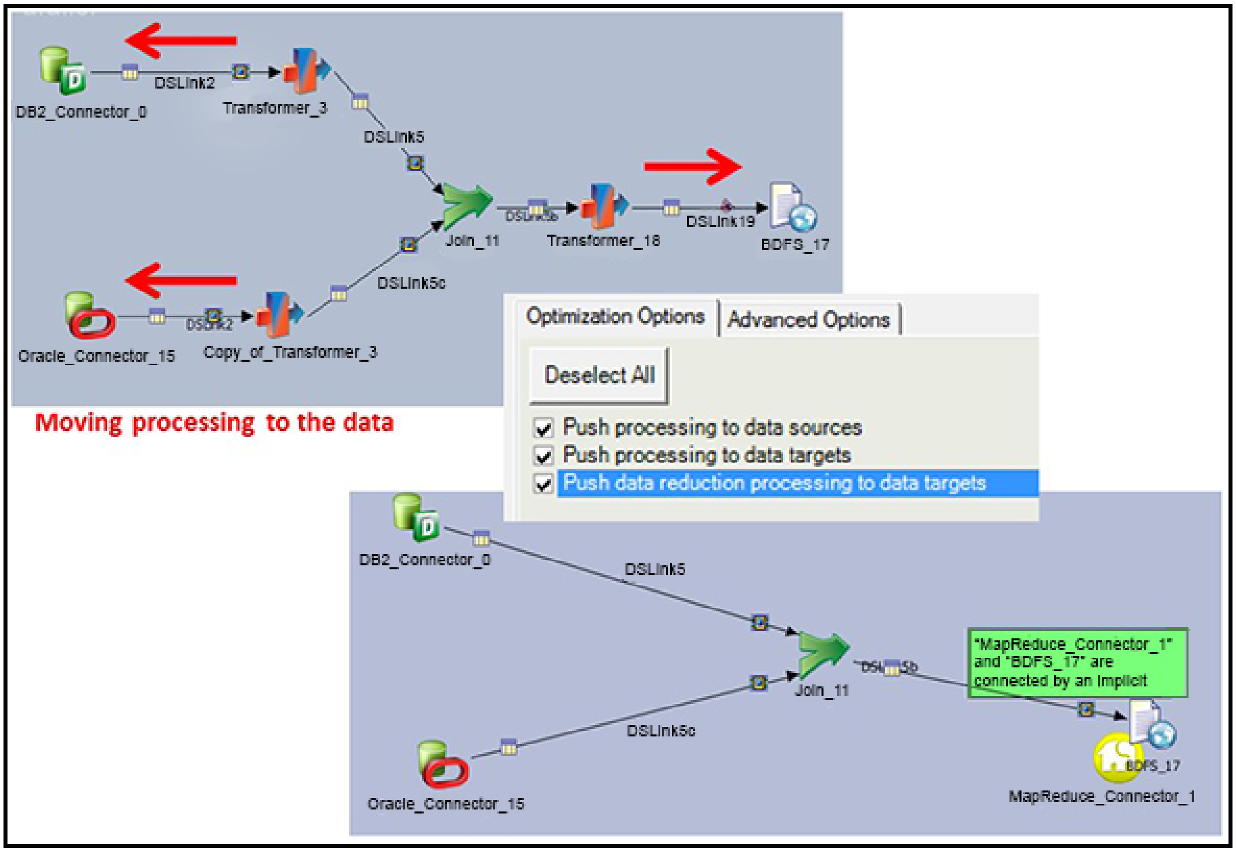
Developer skills

Looking at the question from a skill set point of view is a very site-specific situation. Large numbers of developers know SQL, which can be used for building data integration workloads. However, also consider that increased data volumes or complexity of operations demand more skills of the DBA team to continually adjust and tune the database environment. One limitation of SQL is that you cannot express some types of complex data integration logic. Because SQL is a set-based programming language, developer productivity for complex logic is lower (generally by a factor of 10X) when compared to using a data integration platform, such as the IBM InfoSphere Information Server. InfoSphere Information Server developers build data integration workloads by using a graphical user interface, which is generally associated with a rapid learning curve and yields significant developer productivity. Another advantage of this approach is that you can implement more complex data integration logic than can be expressed in SQL.

An ETL grid might require a more experienced infrastructure team, but most shops have significant Linux or UNIX experience and capabilities already. .In moderate-sized production environments, Hadoop infrastructure teams also demand a skilled infrastructure team. Unless you have a graphical tool to design the processing done in Hadoop (such as InfoSphere Information Server), you will also need to have staff trained in hand-coded languages (such as Java, Pig, Python, and so on).

Based on the concepts presented, the following list notes the standards for data integration (what to run, when, and where):

- **Database (conventional EDW).** Push ELT into the database only when it makes sense. This is usually when the data is already stored in tables and you have excess capacity on the MPP database engine. The days of pushing all ELT workloads into the data warehouse appliance because it is the only scalable engine (and dumping lots of data into this appliance) are coming to an end.
- **Hadoop.** Run data integration in Hadoop if that is where your deep data stores over time are persisted. Also, run data integration in Hadoop when you need to handle unstructured data.
- **ETL grid.** Over time, you can expect to see some ETL grid workloads migrate to Hadoop. However, there are some use cases where Hadoop might not provide many advantages. This includes situations in which you do not need to retain deep stores of data, such as using an InfoSphere Information Server grid to produce forecasts that get loaded into the ERP system. A second example is a large corporate billing system that is running on an InfoSphere Information Server grid. The grid does a lot of processing to construct the bills. (See Figure 4.)



Moving processing to the data

☑ Push processing to data sources
☑ Push processing to data targets
☑ Push data reduction processing to data targets

"MapReduce_Connector_1" and "BDFS_17" are connected by an implicit

Figure 4. Moving the processing to the data

Many of the MPP databases and ETL grid solutions, such as InfoSphere Information Server, implement both pipeline and *dynamic partition parallelism* (the ability to change the data collocation properties on the fly without landing the data). This implementation has shown their flexibility and throughput characteristics to be significantly better than Hadoop MapReduce when given similar resources. Enhancing a database according to the resources available in an ETL grid or Hadoop cluster can also deliver high throughout. However, this approach typically proves to be more costly, especially for ETL and ELT processing.

Given that grids and databases are typically more efficient than Hadoop and grids, and Hadoop is more cost-efficient than databases, logic currently tends to favor an ETL grid as the most cost-effective solution. This logic is further supported by the ability to "tune" the grid hardware configuration to the specific workload requirements. For example, some data integration processing can be extremely CPU-intensive, while other workloads can be extremely I/O-intensive. The Hadoop cluster is normally constructed by using a common building block approach. In this approach, to add CPU capacity, you need to add both CPU and storage, even if you do not need the storage or I/O bandwidth. Likewise, to add I/O bandwidth or storage, you need to add CPU capacity whether you need it or not.

ETL grids have historically addressed expansion with their configuration flexibility. Grid scheduling software has also matured to support nodes with special hardware configurations. This ETL grid scenario works well if the data being processed is structured. If unstructured and semi-structured data is introduced, the advantage changes to Hadoop. Hadoop's *schema on read* philosophy provides significantly more flexibility for these types of data. IBM experts see the ability to bring this advanced grid type flexibility and scheduling to the Hadoop world as the next advancement in the Hadoop architecture. With that advancement comes the ability to run native ETL grid-style workloads (with their structured data throughput requirements) alongside Hadoop workloads (with their unstructured requirements).

Handling unstructured data

Certainly there are some defined unstructured data types, such as video and audio feeds, but even these sources have video recognition and audio extraction software available to render them programmatically useful. *Schema on read* is a valuable feature that provides maximum flexibility for exploratory analysis. However, for Hadoop to emerge into the mainstream enterprise processing environment, a judicious balance must be found regarding when to apply structure to the data. Although everyone talks about processing unstructured data, data is not useful until a structure is assigned to it. Although this seems contradictory of the Hadoop philosophy of unstructured data (which is schema on read), the key to useful data structuring is applying either schema on read or schema on write, whichever is appropriate. For example, while taking in the data, you might want to assign part of the data to have a structure and part to remain unstructured. The portion of the data that you assign structure to would be the attributes necessary for data partitioning, as well as data that is already structured.

The reasons that the data might be structured e early in the process (while taking in the data) are to reduce redundant downstream processing and to simplify the data processing that is required by the end user. This approach saves resources, because the structuring needs to be done only once. It also provides a more standardized (quality) data foundation for the users. If every user is required to provide the entire structure at read time, individual users might include differences in the way the data is interpreted, and this can lead to inconsistent results.

Although our data explorers and data scientists might prefer raw data for their analysis, efficiently leveraging big data for business use in the enterprise requires the implementation of these structures before their use (to insulate the business users from interpretation errors). Although, on the surface, Hadoop seems to be the only option for processing unstructured data, the reality is that these techniques have been used in complex data processing for years. In the past, they were called CLOBs (character large objects) and BLOBs (binary large objects) and coded specifically for those conditions. What Hadoop offers are powerful tools to apply structure to unstructured data and, especially, semi-structured data.

Offloading ELT from the EDW to Hadoop

There are two dimensions to consider when talking about moving ETL from inside to outside of a database. Offloading ELT has been a common focus as the business demands on the database increase and the query workloads begin to collide with the ELT workloads. Continuing to grow the database has often been a harsh experience for organizations, and more businesses are moving to a hybrid approach for processing where that is logical. For example, that might mean processing transactions outside the database in some cases but inside the database as the data volumes involved become large. When decision-makers consider using a tool to generate the transformation logic and pushing that logic into the database, what they are realizing is really the birth of moving processing closer to the data as volumes increase. This is the Hadoop philosophy. After choosing to offload some portion of the ELT workload, the question becomes *where* to move it. This reverts somewhat to our earlier grid versus database versus Hadoop discussion. There are tools available, such as InfoSphere Information Server, that allow code to be generated and deployed in both the grid and the Hadoop environment. In fact, with Balanced Optimization, you can also build SQL, which can be pushed into the database.

One of the main enabling factors in the ability to take advantage of Hadoop has been the expanding support for a SQL interface. SQL is, by far, the most common interface used by business users, and there are many IT tools that can interact with SQL. Being able to use a known high-level user interface has been recognized as the most prominent hurdle to overcome to legitimize Hadoop in the business world. Although all dialects of SQL involve some differences, most relational databases have progressed to supporting most functions at the SQL 2011 level. Unfortunately, most of the Hadoop SQL distributions are stuck at supporting only subsets (at best) of the SQL 2003 standard. This limitation means that for an item currently running in a database to be moved into Hadoop, some of the SQL will need to be rewritten or user-defined functions (UDFs) will need to be created to fill the gaps.

The one product that has been available to address this need is IBM Big SQL, which also works in today’s Hadoop world by providing true SQL 2011 compatibility. SQL running in a database today, which does not use vendor-specific SQL functions, can be ported to Hadoop with minimal or no change. In fact, IBM has developed conversion techniques that move ELT functionality from both IBM Netezza® and Teradata into Hadoop. Moving Oracle ELT into Hadoop is also a straightforward process. This approach allows transformation processing to be done on more cost-effective hardware. It also enables the operations to be moved to the data, whether it is structured or unstructured, because assigning structure to the data can be done before SQL processing (similar to a UDF in a traditional relational database).

Joins, aggregations, and sorts in Hadoop

Joins, aggregations, and sorts are historically heavy memory consumers, and many Hadoop SQL solutions have had a history of aborting the process when memory requirements surpass the available memory. For this reason, many organizations have been cautious about moving these operations to Hadoop. Unfortunately, most complex ETL processing is highly dependent on these design patterns, so they typically make up a significant portion of the resource usage and elapsed time. Later releases of these SQL processing engines appear to manage these memory intensive operations better, but more time and examples of success are needed to resolve current concerns. One exception to this limitation is Big SQL, which has several features to optimize how memory-based operations work and can offload data to disk during periods of high memory use.

Hadoop and data collocation

One of the initial challenges of working with data in Hadoop is that data is typically stored in a random fashion across the nodes. A join operation often requires moving large amounts of data between nodes to bring related data together for processing. Partitioning advances in Hive and Parquet can help to some degree, but these are mainly aimed at improving query performance on a single table, not join performance. Techniques such as data collocation and partition elimination can drastically reduce run times and resource requirements. Although partition elimination can be available in Hive and Impala, data collocation is not available, because there is currently no way to assign partitions in different tables to the same node. Also, with Hive and Parquet, all of the data in a partition typically needs to be scanned to find a specific record. This makes operations such as lookups very expensive from a run-time and resource use viewpoint.

The ability to perform a direct lookup is important for operations such as code replacement and standardizing fields, key lookups, and so on during data integration operations. Many organizations are looking to Apache HBase or other NoSQL databases to provide this lookup capability in the Hadoop environment. Data structures (at least those involved in data location) can be implemented before the data is landed to help generate table statistics. It is ideal if the data is already structured appropriately when it arrives, but introducing structure while taking in the data can create complexity and impedance (possibly requiring error handling and data rejection algorithms). This is often undesirable if the goal is to quickly load an exact copy of the data into the landing area (which is common in data lake architecture).

The following paragraphs briefly describe the join techniques in both Cloudera Impala on Parquet files and Apache Tez or Hortonworks Stinger on Parquet or HDFS files. In both cases, data is required, by default, to be broadcast across the nodes to provide data collocation.

The Impala query planner decides between two techniques for performing join queries (broadcast or partitioned). This decision is dependent on the absolute and relative sizes of the tables. If the right table is considered to be smaller than the left table, the default is a broadcast join. In a *broadcast* join, the contents are sent to all of the other nodes involved in the query. A *partitioned* join (not related to a partitioned table) is used for large tables of roughly equal size. In a partition join, portions of each table are sent to appropriate other nodes. In doing so, subsets of rows can be processed in parallel. Another factor in the choice between a broadcast and portioned join depends on statistics being available for all tables in the join. These statistics are collected by the COMPUTE STATS statement.

With the delivery of Hive on Tez, users have the option of executing queries on Tez. Using Tez facilitates simpler, more efficient query plans because of its data flow model on a directed acyclic graph (DAG) of nodes. This model results in significant performance improvements and interactive query on Hive and Hadoop. The following list notes some of the techniques that account for the enhanced speed:

- **Broadcast joins.** Eliminates the need to build a hash table on the client when using a product such as Hive's MapJoin
- **Dynamic partitioned hash joins.** Distributes a small table based on the Big Table bucketing trait
- **Cardinality estimation.** Bases decision on join algorithm and parallelism

Information governance

Information governance is the orchestration of people, process, and technology to enable an organization to leverage data as an enterprise asset. It is the high level planning and control of information management activities that enable an organization to leverage data as a shared asset, "fit for purpose" by users with different expectations. Information Governance must include the specification of decision rights and an accountability framework to ensure appropriate behavior in the valuation, creation, storage, use, archiving and deletion of information. It includes the processes, roles and policies, standards and metrics that ensure the effective and efficient use of information in enabling an organization to achieve its goals.

Information governance requires organization, process, and enabling technology changes that span both information technology and business in the management of data. Keep in mind the following key objectives for building information governance:

- Establishing a culture that recognizes the value of data as an enterprise asset
- Building governance infrastructure, technology, and a supporting organization
- Defining processes and business rules for ongoing governance
- Developing common and standard data domain definitions
- Developing architecture practices and standards
- Measuring, monitoring, and managing data quality
- Understanding data lineage and impact analysis
- Understanding metadata and the accuracy of the metadata
- Advancing toward self-service data provisioning
- Connecting operational activities with corporate business performance
- Ensuring effective compliance and regulatory support infrastructure

The following components are also critical in building information governance:

- Business intelligence and performance management

  Planning, budgeting, and forecasting uses analytics to align financial and operational plans, understand target values for key categories of revenue and expenditure, and evaluate expected business outcomes. It measures progress against leading industry preferred practices for the purpose of identifying opportunities to better link strategy to action, optimize budget allocations, and perform what-if analysis.

- Information lifecycle management (ILM)

  ILM includes the policies, processes, practices, and tools that are used to align the business value of information effective IT infrastructure from the time information that is conceived through its final disposition. Information is aligned with business processes through management of service levels that are associated with applications, metadata, information, and data.

- Data warehousing

  The Data Repository layer contains the databases, data stores, and related components that provide most of the storage for the data that supports a Business Analytics and Optimization (BAO) environment. The Data Repository layer repositories are not a replacement or replica of operational databases that reside on the Data Source layer, but are a complementary set of data repositories that reshape data into formats that are necessary for making decisions and managing a business. These database structures are represented by conceptual, logical, and physical data models and data model types, for example 3NF, star and snowflake schemas, unstructured, and so on.

- Data integration

  The Data Integration architectural layer focuses on the processes and environments that deal with the capture, qualification, processing, and movement of data to prepare it for storage in the data repository layer, which is subsequently shared with the analytical and access applications and systems. This layer can process data in scheduled batch intervals or in near real-time and just-in-time intervals, depending on the nature of the data and the business purpose for its use.

- Master data management (MDM)

  MDM is a set of disciplines, technologies, and solutions to create and maintain consistent, complete, contextual, and accurate business data for all stakeholders across and beyond the enterprise.

- Data lineage and impact analysis

  These topics are well-known for their vital relationship to streamlining maintenance activities, but a more significant business requirement is taking center stage as regulatory controls are spreading. The ability to demonstrate mastery of the processing that has occurred to data as it moves through the organization and validate its lineage can have a significant financial impact on an organization. This is an area where hand coding poses a challenge to collecting and maintaining this metadata.

- Metadata

  Even if metadata is accurate when the application is deployed, unless a rigorous manual application lifecycle maintains this metadata (and is verifiable), it will not hold up to the necessary rigor. This is an area where ETL tools are a priority. The ability to maintain the metadata regardless of where the process is actually occurring (in the grid, in the database, or in Hadoop) can be worth its weight in gold when audit time strikes. As technologies advance, it is also important to retain the flexibility to move the processing to the most appropriate location. The ability to graphically create the transformation specifications and determine the appropriate location for them to reside is essential. Prototyping and iterative development is a critical requirement for the future.

- Self-service data provisioning

  Facilitating this provisioning (using metadata or other methods) is a key need as the industry moves toward allowing business users more timely access to data that they are authorized to see. The metadata can provide the foundation that is necessary to find the data and understand its heritage. InfoSphere Information Server also provides a provisioning layer, called *Data Click* , that is easy for business users to access data while maintaining the data governance facilities to track and manage the provisioned data.

Business needs today extend beyond simple data processing. The InfoSphere Information Server suite is more than capable of handling those extended business needs through, for example, the Information Governance Catalog (IGC). The IGC provides comprehensive information integration capabilities to help you understand and govern your information and encourages a standardized approach to discovering your IT assets and defining a common business language. See the Integration section of this guide and this web page for more information:

http://www.ibm.com/software/products/en/infosphere-information-governance-catalog

IBM has produced a wealth of material regarding strategies and techniques for implementing data governance. For more information about this topic, explore the IBM Redbooks website:

http://www.redbooks.ibm.com

Usage scenarios

Organizations need to have the freedom and flexibility to build a data integration application once and then run it wherever it makes the most appropriate sense, whether in the ETL grid, in the parallel database, or in Hadoop. The decision regarding which environment you choose depends on what the organization is trying to accomplish.

The scenarios that follow show examples of the use of each example.

- ETL grid example

  A major logistics company has migrated its mission-critical billing system from the mainframe and reimplemented it as an InfoSphere Information Server application running in a low-cost Linux grid. The choice of the Linux grid is appropriate because the primary requirement is for low-cost, high-throughput processing of complex application logic. This particular application does not require storage of large volumes of operational data over time, which lessens the need for the Hadoop Distributed File System to store large data volumes. Another important point is that, if necessary, the ETL grid can be constructed by using the same commodity processing and storage hardware that is typically found in Hadoop environments. Businesses do not need to implement a Hadoop infrastructure just to use low-cost hardware. IBM InfoSphere Information Server users have been exploiting commodity grid hardware and storage for processing massive data volumes for more than a decade.

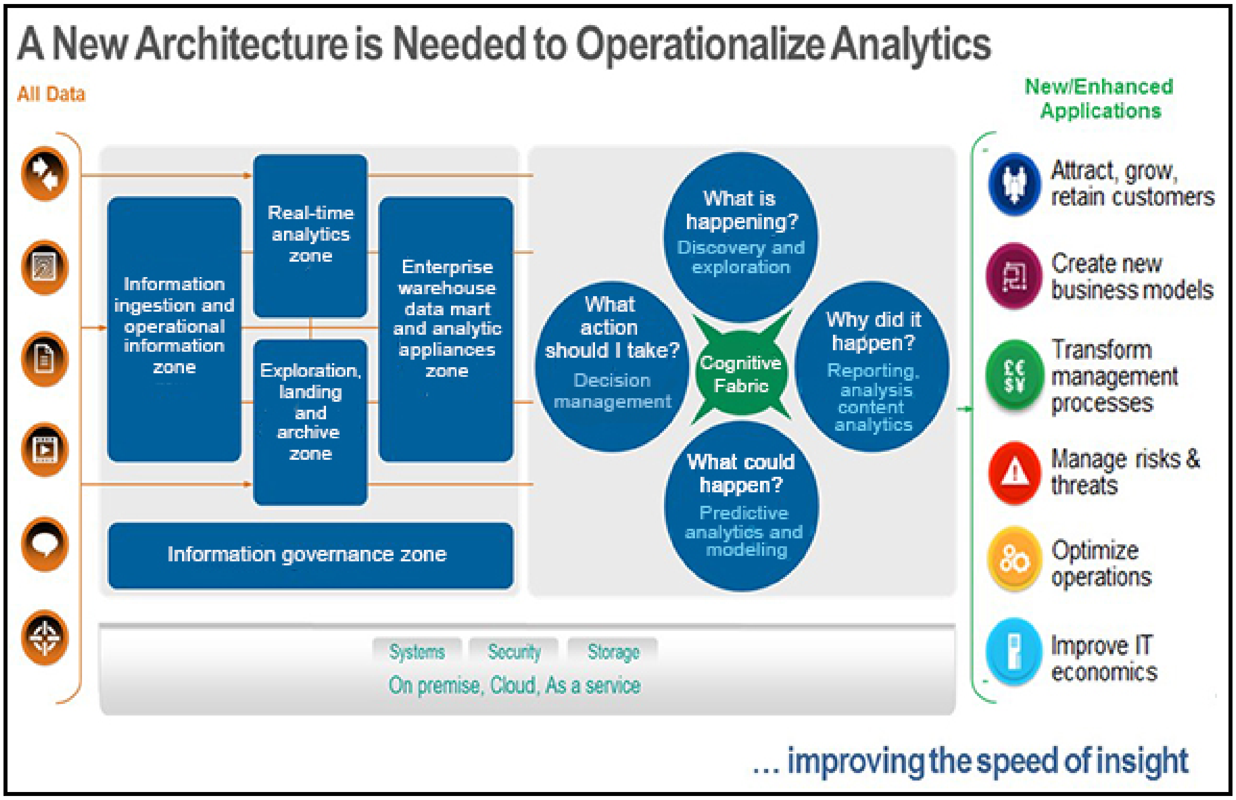
- Parallel database example

  A major healthcare provider wants to maximize Hadoop for offloading the massive ELT workloads that are running in the enterprise data warehouse. In the process of off-loading ELT workloads (which now consume more than 50% of the database processing capacity), the provider wants to eliminate many of the negative consequences associated with 100% reliance on ELT pushdown into the database. This includes replacing hand-coded data integration logic (which requires support by a team of hundreds of individuals) with data integration tools to implement data quality processing for a single, consolidated view of data and implementing data governance. After offloading the majority of the ELT workload from the data warehouse to Hadoop, the provider will continue to push limited data integration processing into the database when it is appropriate (when the data is already stored in database tables and operations such as sorts, joins, and aggregations can take advantage of the MPP database engine). However, the majority of the ELT workloads will migrate to Hadoop to save money and to improve data quality and governance.

- Hadoop example

  A global bank wants to store all operational data coming from critical enterprise applications in Hadoop. The data stored in Hadoop will include many data sources that could not be stored previously because of storage costs. Hadoop is the appropriate environment because of the HDFS and low-cost storage and processing. The bank wants to perform all data integration processing in Hadoop before sending data to downstream data warehousing, analytical, and reporting systems.

These examples illustrate the range of problems that IBM software can accommodate without isolating the solution to a particular technology. IBM is not only a distributor of all these technologies but also as a partner with vendors across all these technologies, so you do not need to “force fit” one specific architecture to all problems. Figure 5 illustrates an IBM reference architecture. This reference architecture is a logical description and is flexible enough to fit all of these use cases. We encourage you to contact us to discuss this open approach to data integration.



A New Architecture is Needed to Operationalize Analytics

All Data

Information ingestion and operational information zone

Real-time analytics zone

Exploration, landing and archive zone

Enterprise warehouse data mart and analytic appliances zone

Information governance zone

What is happening? Discovery and exploration

What action should I take? Decision management

Cognitive Fabric

Why did it happen? Reporting, analysis content analytics

What could happen? Predictive analytics and modeling

Systems

Security

Storage

On premise, Cloud, As a service

New/Enhanced Applications

Attract, grow, retain customers

Create new business models

Transform management processes

Manage risks & threats

Optimize operations

Improve IT economics

... improving the speed of insight

Figure 5. Designing a big data architecture

Integration

The following sections provide an overview of the components of the IBM InfoSphere Information Server suite of tools.

InfoSphere Information Server Software Components

- InfoSphere Information Governance Catalog

  An interactive, web-based tool that enables users to create, manage, and share an enterprise vocabulary and classification system in a central catalog, which helps an organization to understand the business meaning of their assets and provide search, browse, and query capabilities (Users can establish asset collections and run lineage reports to examine data flow between assets.)

- InfoSphere DataStage® and QualityStage®

  Form the foundation for data transformation and collection of relevant design and operational metadata

- InfoSphere Information Analyzer

  Provides a sophisticated data profiling capability that also integrates with IGC to provide this information to the business users

- InfoSphere Data Click

  Provides a simple graphical interface for data provisioning and maintains governance controls within IGC

- InfoSphere FastTrack

  Accelerates the design time to create source-to-target mappings and to automatically generate jobs

- InfoSphere Blueprint Director

  Extends the vision of the projects to all members of your team, fostering collaboration, best practices, and connectedness

- InfoSphere Information Services Director

  Provides a unified and consistent way to publish and manage shared information services

- InfoSphere Metadata Asset Manager

  Imports and exports metadata from design tools, business intelligence tools, databases, and files into and out of the metadata repository of InfoSphere Information Server

Data connectivity support

IBM InfoSphere has a wide range of capabilities to connect to data sources and targets. It supports the following major databases natively:

- Greenplum
- Hadoop Distributed File System (HDFS), including balanced optimization, ELT
- IBM Cognos® TM1®
- IBM DB2®, including balanced optimization, ELT
- IBM Informix®
- IBM® PureData™ for Analytics (PDA), powered by Netezza, including balanced optimization, ELT
- IBM Red Brick® Warehouse
- iWay Integration Suite
- Java Database Connectivity (JDBC) and Open Database Connectivity (ODBC)
- Microsoft SQL Server and Microsoft OLE
- Oracle, including balanced optimization, ELT
- Sybase
- Teradata, including balanced optimization, ELT
- UniVerse and UniData

Application connectivity support

- IBM InfoSphere Master Data Management
- IBM InfoSphere Streams
- IBM WebSphere® ILOG® JRules
- Java
- Salesforce
- SAP and SAP Business Warehouse (SAP BW)

Low latency support

- Distributed transactions
- IBM InfoSphere Change Data Capture
- IBM® MQ

For details about these topics, see “New features and changes for IBM InfoSphere Information Server, Version 11.3” in the IBM Knowledge Center:

http://ibm.co/1N1XUHl

Supported software

IBM InfoSphere Information Server V11.3.1 is compatible with the products listed in this section.

Operating systems:

- IBM® AIX® version 6.1
- IBM Linux on z Systems™
- Microsoft Windows 7 and 8 and Windows Server 2008 and 2012
- Oracle Solaris version 10
- Red Hat Enterprise Linux (RHEL) version 6
- SUSE Linux Enterprise Server (SLES) version 11

Databases:

- IBM DB2
- Microsoft SQL Server
- Netezza, an IBM company
- Oracle
- Sybase
- Teradata

Hadoop distributions:

- IBM BigInsights™
- Cloudera
- Hortonworks

Note: The DataStage® component of IBM InfoSphere Information Server can be installed in a multitier environment. For details about which components configuration and support options for tiered installations, see the IBM Knowledge Center:

- IBM InfoSphere Information Server Welcome page

  http://www.ibm.com/support/knowledgecenter/SSZJPZ/welcome

- IBM InfoSphere Information Server Planning, Installation, and Configuration Guide

  http://ibm.co/1CxZldz

Ordering information

To order InfoSphere or try BigInsights, go to the web pages cited in this section.

The IBM InfoSphere Platform

The InfoSphere Platform provides all the foundational building blocks of trusted information, including data integration, data warehousing, master data management, big data and information governance. The platform provides an enterprise-class foundation for information-intensive projects. It provides the performance, scalability, reliability, and acceleration that are needed to simplify difficult challenges and deliver trusted information to your business faster.

For your business requirements and specific needs, review the capabilities you need, and then select: **Request a quote** on the following web page:

http://www.ibm.com/software/data/infosphere/

IBM BigInsights

IBM BigInsights on Cloud trial is available on IBM Bluemix™. Bluemix is the IBM open standards, cloud-based platform for building, managing, and running apps of all types. When you have a Bluemix account, it takes only minutes to get the IBM BigInsights on Cloud trial up and running. Go to the following website to try IBM BigInsights:

http://www.ibm.com/software/data/infosphere/hadoop/trials.html

Related information

- IBM Offering Information page (to search on announcement letters, sales manuals, or both):

  http://www.ibm.com/common/ssi/index.wss?request_locale=en

On this page, enter <solution name> (remove angle brackets), select the information type, and then click **Search**. On the next page, narrow your search results by geography and language.

- IBM InfoSphere Information Server

  http://www.ibm.com/software/products/en/infosphere-information-server

- Apache Hadoop

  http://hadoop.apache.org/

- IBM BigInsights for Apache Hadoop

  http://www.ibm.com/software/data/infosphere/hadoop/enterprise.html

- InfoSphere DataStage for Enterprise XML Data Integration, SG24-7987

  http://www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/sg247987.html

- IBM InfoSphere Information Server Deployment Architectures, SG24-8028

  http://www.redbooks.ibm.com/abstracts/sg248028.html

- Metadata Management with IBM InfoSphere Information Server, SG24-7939

  http://www.redbooks.ibm.com/abstracts/sg247939.html

- IBM InfoSphere Information Server Installation and Configuration Guide, REDP-4596

  http://www.redbooks.ibm.com/abstracts/redp4596.html

- Smarter Business: Dynamic Information with IBM InfoSphere Data Replication CDC, SG24-7941

  http://www.redbooks.ibm.com/abstracts/sg247941.html

- Optimizing Data Integration Solutions by Customizing the IBM InfoSphere Information Server Deployment Architecture, TIPS-0964

  http://www.redbooks.ibm.com/abstracts/tips0964.html

- Implementing IBM InfoSphere Change Data Capture for DB/2 z/OS V6.5, REDP-4726

  http://www.redbooks.ibm.com/abstracts/redp4726.html

- The Value and Benefits of IBM InfoSphere BigInsights Running on IBM System z, TIPS-1215 http://www.redbooks.ibm.com/abstracts/tips1215.html

- Big Data Networked Storage Solution for Hadoop, REDP-5010

  http://www.redbooks.ibm.com/abstracts/redp5010.html

- Hadoop and System z, REDP-5142

  http://www.redbooks.ibm.com/abstracts/redp5142.html

- IBM System x Reference Architecture for Hadoop: IBM InfoSphere BigInsights Reference Architecture, REDP-5009

  http://www.redbooks.ibm.com/abstracts/redp5009.html

- Better Business Decisions at a Lower Cost with IBM InfoSphere BigInsights, TIPS-0934

  http://www.redbooks.ibm.com/abstracts/tips0934.html

- Building Big Data and Analytics Solutions in the Cloud, REDP-5085

  http://www.redbooks.ibm.com/abstracts/redp5085.html

- IBM InfoSphere Streams: Accelerating Deployments with Analytic Accelerators, SG24-8139

  http://www.redbooks.ibm.com/abstracts/sg248139.html

- Using the IBM Big Data and Analytics Platform to Gain Operational Efficiency, TIPS-1170

  http://www.redbooks.ibm.com/abstracts/tips1170.html

- IBM PureData System for Analytics Architecture: A Platform for High Performance Data Warehousing and Analytics, REDP-4725

  http://www.redbooks.ibm.com/abstracts/redp4725.html

- Turning Big Data into Actionable Information with IBM InfoSphere Streams, TIPS-0948

  http://www.redbooks.ibm.com/abstracts/tips0948.html

- InfoSphere DataStage Parallel Framework Standard Practices, SG24-7830

  http://www.redbooks.ibm.com/abstracts/sg247830.html

- IBM Information Server: Integration and Governance for Emerging Data Warehouse Demands, SG24-8126

  http://www.redbooks.ibm.com/abstracts/sg248126.html

- Governing and Managing Big Data for Analytics and Decision Makers, REDP-5120

  http://www.redbooks.ibm.com/abstracts/redp5120.html

- IBM Information Governance Solutions, SG24-8164

  http://www.redbooks.ibm.com/abstracts/sg248164.html

- Information Governance Principles and Practices for a Big Data Landscape, SG24-8165

  http://www.redbooks.ibm.com/abstracts/sg248165.html

Notices

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

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information 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 give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you. This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

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

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.

This document was created or updated on March 25, 2015.

Send us your comments in one of the following ways:

- Use the online **Contact us** review form found at:
  ibm.com/redbooks
- Send your comments in an e-mail 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 U.S.A.

This document is available online at http://www.ibm.com/redbooks/abstracts/tips1265.html .

Trademarks

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

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

AIX®
BigInsights™
Bluemix™
Cognos®
DataStage®
DB2®
IBM®
IBM PureData™
ILOG®
Informix®
InfoSphere®
Insights™
PureData®
QualityStage®
Red Brick®
Redbooks®
Redpaper™
Redbooks (logo)®
System z®
TM1®
WebSphere®
z Systems™
z/OS®

The following terms are trademarks of other companies:

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

Intel, 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.

Microsoft, 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.