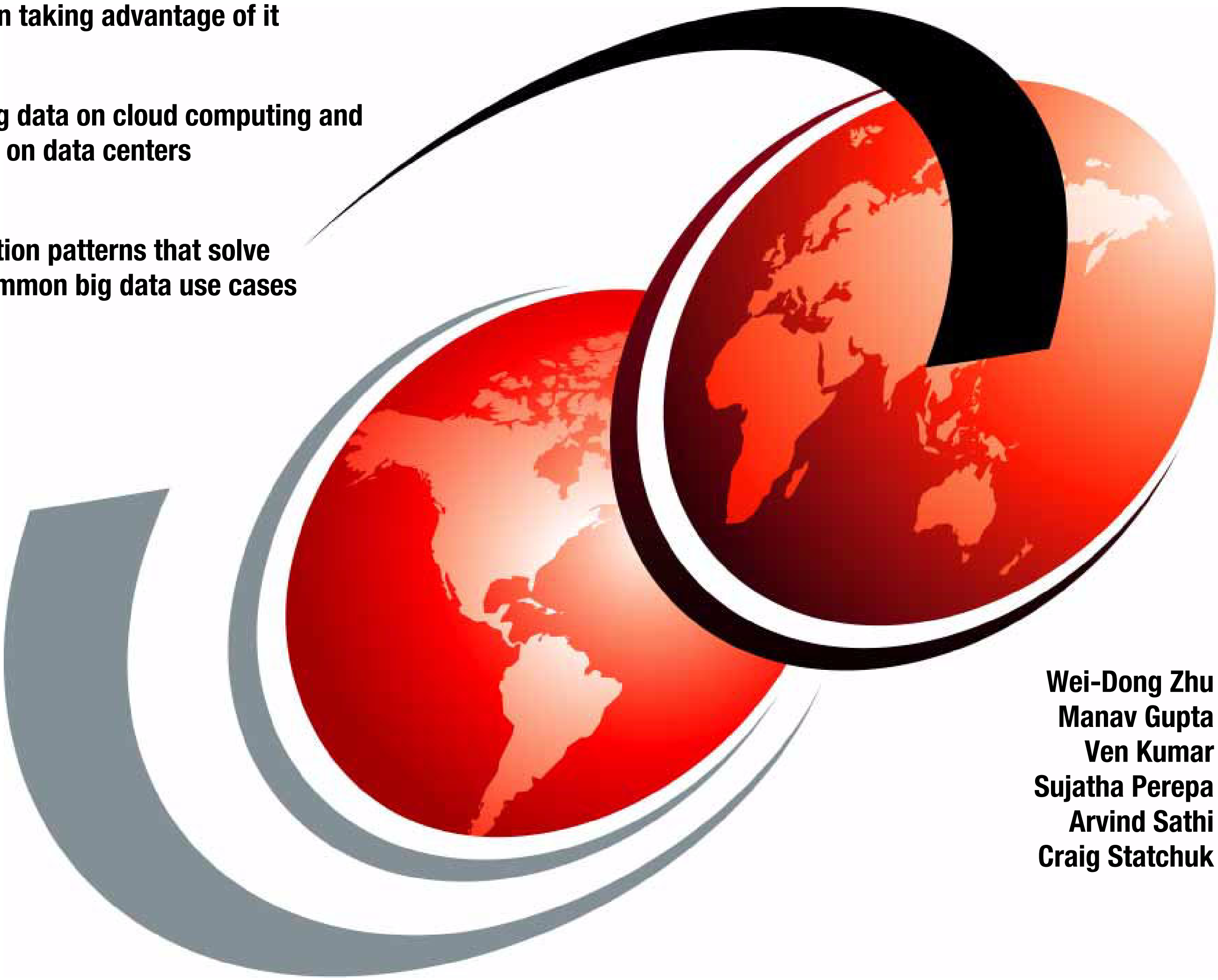
Building Big Data and Analytics Solutions in the Cloud

Characteristics of big data and key technical challenges in taking advantage of it

Impact of big data on cloud computing and implications on data centers

Implementation patterns that solve the most common big data use cases



Wei-Dong Zhu
Manav Gupta
Ven Kumar
Sujatha Perepa
Arvind Sathi
Craig Statchuk

ibm.com/redbooks

Redpaper

International Technical Support Organization

Building Big Data and Analytics Solutions in the Cloud

December 2014

REDP-5085-00

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

First Edition (December 2014)

This edition applies to:
IBM InfoSphere® BigInsights™
IBM PureData™ System for Hadoop
IBM PureData System for Analytics
IBM PureData System for Operational Analytics
IBM InfoSphere Warehouse
IBM InfoSphere Streams
IBM InfoSphere Data Explorer (Watson™ Explorer)
IBM InfoSphere Data Architect
IBM InfoSphere Information Analyzer
IBM InfoSphere Information Server
IBM InfoSphere Information Server for Data Quality
IBM InfoSphere Master Data Management Family
IBM InfoSphere Optim™ Family
IBM InfoSphere Guardium® Family

Contents

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 grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

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

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

Trademarks

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

Algorithmics®
BigInsights™
Cognos®
DB2®
developerWorks®
GPFS™
Guardium®
IBM PureData™
IBM Watson™
IBM®
IMS™
Informix®
InfoSphere®
Netcool®
OpenPages®
Optim™
PartnerWorld®
PureData™
Redbooks®
Redpaper™
Redbooks (logo) ®
SPSS®
Tivoli®
Watson™
z/OS®
z/VM®

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.

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.

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

Executive summary

Big data is currently one of the most critical emerging technologies. Organizations around the world are looking to exploit the explosive growth of data to unlock previously hidden insights in the hope of creating new revenue streams, gaining operational efficiencies, and obtaining greater understanding of customer needs.

It is important to think of big data and analytics together. *Big data* is the term used to describe the recent explosion of different types of data from disparate sources. *Analytics* is about examining data to derive interesting and relevant trends and patterns, which can be used to inform decisions, optimize processes, and even drive new business models.

With today's deluge of data comes the problems of processing that data, obtaining the correct skills to manage and analyze that data, and establishing rules to govern the data's use and distribution. The big data technology stack is ever growing and sometimes confusing, even more so when we add the complexities of setting up big data environments with large up-front investments.

Cloud computing seems to be a perfect vehicle for hosting big data workloads. However, working on big data in the cloud brings its own challenge of reconciling two contradictory design principles. Cloud computing is based on the concepts of *consolidation* and *resource pooling*, but big data systems (such as Hadoop) are built on the *shared nothing* principle, where each node is independent and self-sufficient. A solution architecture that can allow these mutually exclusive principles to coexist is required to truly exploit the elasticity and ease-of-use of cloud computing for big data environments.

This IBM® Redpaper™ publication is aimed at chief architects, line-of-business executives, and CIOs to provide an understanding of the cloud-related challenges they face and give prescriptive guidance for how to realize the benefits of big data solutions quickly and cost-effectively.

The paper covers these topics:

- The characteristics of big data
- The business drivers and key technical challenges in exploiting big data
- The impact of big data on cloud computing environments
- Functional and infrastructure architecture considerations for big data and data analytics in the cloud
- Implementation patterns to solve the most common big data use cases
- The implications of big data on data centers
- A roundup of available NoSQL technologies

In addition, this paper introduces a taxonomy-based tool that can quickly help business owners understand the type of big data problem at hand.

This paper is based on the knowledge IBM has gained in both cloud computing and big data, and builds upon the popular Cloud Computing Reference Architecture (CCRA) and its principles. The objective is not to provide an exhaustive list of every capability required for every big data solution hosted in a cloud environment. Rather, the paper aims to give insight into the most common use cases encountered across multiple industries and the common implementation patterns that can make those use cases succeed.

Authors

This paper was produced by a team of specialists from around the world working at the International Technical Support Organization (ITSO).

These dedicated IBM professionals collaborated to produce this publication:

Wei-Dong (Jackie) Zhu is an Enterprise Content Management, Risk, and Discovery Project Leader with the ITSO in San Jose, California. She has more than 10 years of software development experience in accounting, image workflow processing, and digital media distribution (DMD). Her development work with one of the DMD solutions contributed to a first time ever win for IBM of a technical Emmy award in 2005. Jackie holds a Master of Science degree in Computer Science from the University of the Southern California. Jackie joined IBM in 1996. She is a Certified Solution Designer for IBM Content Manager and has managed and lead the production of many Enterprise Content Management, Risk, and Discovery IBM Redbooks publications.

Manav Gupta is a Software Client Architect in Canada, bringing thought leadership across IBM Software Group brands to clients in the telecommunications industry. He has 15 years' experience in the telecommunications industry, with particular expertise in systems management, cloud computing, and big data. He has written extensively on fault and performance management (using products and technologies such as IBM Tivoli® Netcool®), cloud computing, and big data. Manav holds a degree in mathematics and a postgraduate diploma in software development.

Ven Kumar is a Client Technical Architect at IBM. He specializes in big data architecture and analytics, cloud computing, enterprise architecture, and Mobile and Web 2.0 technologies. Ven has over 15 years of industry experience in retail, consumer packaged goods, high-tech manufacturing, and ecommerce. Prior to joining IBM, Ven held positions at HP as the CTO for the Nike Account Team and practice director at Oracle. He holds a Bachelors degree in Electronics and Power Engineering from VNIT in India.

Sujatha (Suj) Perepa is an IBM Software Client Architect currently working on company projects for national security, justice, and law enforcement agencies of the US government. She pursues initiatives in integration, security, and information management technologies. Suj is currently engaged in cloud, big data, intelligence analytics, and cyber-security programs. She has developed multiple IBM Business Partner applications for IBM PartnerWorld®. Suj graduated from Stuart School of Business, Chicago, IL. She is an IBM Inventor and occasional contributor to *Thoughts on Cloud* and IBM developerWorks®.

Arvind Sathi is a Worldwide Communication Sector Architect for big data at IBM. Dr. Sathi received his Ph.D. in Business Administration from Carnegie Mellon University and worked under Nobel Prize winner Dr. Herbert A. Simon. Dr. Sathi has more than 20 years of leadership in Information Management architecture and delivery. His primary focus is creating visions and roadmaps for advanced analytics at leading IBM clients in the telecommunications, media and entertainment, and energy and utilities industries. He is the author of two books on big data, both published by MC Press.

Craig Statchuk is a Technical Architect in the IBM Business Analytics Office of the CTO. Craig has been a software developer, architect, and inventor for more than thirty years. He has worked in various disciplines, including development, consulting, and software architecture. Craig's research focus areas have included data integration, business process evolution, location intelligence, enterprise search, advanced analytics, and big data. He is currently responsible for application architecture and design in the Business Analytics division of IBM.

In addition, the authors want to express sincere thanks to Anshu Kak, Distinguished Engineer, WW SCA Leader-Tech Sales at IBM, for her support in the development of this document. The authors also want to state that the release of this IBM Redpaper publication is not possible without the commitment to IBM CCRA initiatives by the CCRA team.

The project was led by Wei-Dong Zhu, an IBM Redbooks® project leader. Shawn Tooley, a Redbooks technical writer, also contributed.

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

Big data and analytics overview

The amount of data being generated inside and outside each enterprise (see Figure 1-1) has exploded. The increasing volume and detail of information, the rise of multimedia and social media, and the *Internet of Things* are expected to fuel continued exponential data growth for the foreseeable future.

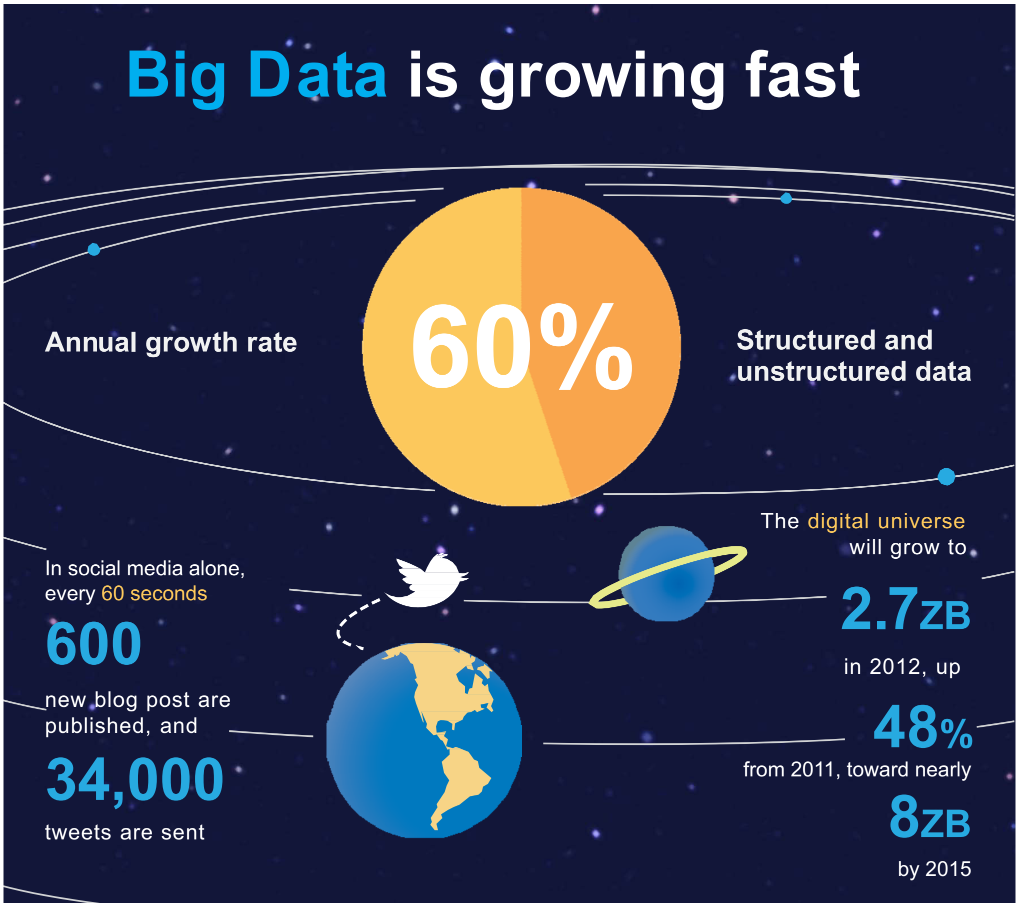


Figure 1-1 The explosion of data

Two common sources of big data exist.

First, there is organizational data, which, thanks to improved automation and broader access, is increasingly being shared. Organizational data includes emails, system logs, internal documents, business process events, and other structured, unstructured, and semi-structured data. It also includes social content, such as any blogs and wikis that are available within the organization.

Second, data comes from sources outside of the organization. Some of this extra-organizational data is available publicly and at no charge, some of it is obtained through paid subscription, and the rest is selectively made available by specific business partners or customers. This includes information from social media sites and various other sources, including product literature; corporate information; health, life, and consumer websites; helpful hints from third parties, and customer complaints, such as complaints posted on the websites of regulatory agencies.

Corporations have discovered that all data, whether internal or external, has nuggets of information from which they can benefit.

In addition, it is worth noting that big data does not simply mean massive quantities of data. Big data also does not equal Hadoop, although the two are often confused. Hadoop is a great new technology and has opened the eyes of many to the world of big data, but it is not the only option for handling the flood of multi-structured data sets and workloads coming from web-based applications, sensors, mobile devices, and social media.

The analyst community has traditionally used the *3 Vs* (volume, velocity, and variety) to define big data characteristics. Lately, the characteristics of veracity and visibility have been added, making it *5 Vs*. And while these 5 Vs are useful to define big data's characteristics, we must not forget that these characteristics have existed for many years. Consider these scenarios:

- A retailer that maintains a 6 TB data warehouse with over 400 dealers accessing summary reports, 800+ jobs in a dirty extract, transform, and load (ETL) window, and 25 concurrent users
- A telecom operator that collects 4 billion daily Call Detail Records (CDRs) that must be converted from native format to ASCII, rated and scored daily, and analyzed hourly for call quality

In both of these examples, the manipulation of the data requires non-traditional systems (in other words, systems such as relational databases) to meet the business objectives. Therefore, it is important for the reader to think beyond the 5 Vs when thinking of big data systems.

For this paper, we consider big data as *all* of the data that needs to be processed by an organization for a business purpose.

1.1 Examples of big data and analytics

Due to a combination of automation, consumer involvement, and market-based exchanges, big data is becoming readily available and is being deployed in a series of significant use cases that are radically disrupting traditional markets. Here are some examples of big data:

- *Social media content*: A wide variety of data, including unstructured data, text, and media content, can be found at various social media websites. This data contains valuable information that can be mined by an enterprise.
- *Cell phone details*: Currently, there are over 5 billion cell phones that can each provide useful information, such as the user's location, how the device is being used, and any functional problems that might be present in the device.
- *Channel-click information from set-top boxes*: Users' interactions with their television set-top boxes provide valuable information about the kinds of programs and topics that interest them.
- *Transactional data*: Today's enormous number of online transactions, each facilitated by various sources, such as mobile wallets and credit cards, are creating terabytes of data that can be mined and analyzed.
- *Documentation*: Documentation, such as financial statements, insurance forms, medical records, and customer correspondence, can be parsed to extract valuable information for further analysis.
- *Internet of Things*: The *Internet of Things* is generating large volumes and various data from sources as varied as ebooks, vehicles, video games, television set-top boxes, and household appliances. Capturing, correlating, and analyzing this data can produce valuable insights for a company.
- *Communications network events*: Communications networks are increasingly interconnected, resulting in the need to monitor large volumes of data and respond quickly to changes. For example, IP Multimedia Subsystem networks require the monitoring and dynamic configuration of various devices in the core and access networks to ensure real-time traffic routing while maintaining quality of service (QoS) levels.
- *Call Detail Records*: Analysis of Call Detail Records (CDRs) enables a company to better understand the habits of its customers and, potentially, of its customers' social networks.
- *Radio Frequency Identification (RFID) tags*: RFID tags are increasingly ubiquitous and the valuable data they contain is often ignored and not analyzed due to the sheer volume and variety of data obtained from these sensors.
- *Traffic patterns*: Today, traffic patterns can be studied based on data from on-road sensors, video cameras, and *floating-car data* (a method of determining current traffic-flow speed based on data from motorists' mobile phones). Rapid analysis of this data can be used to relieve traffic congestion.
- *Weather information*: Weather data is now being correlated to various other large sources of data, such as sales, marketing, and product information, enabling companies to market their products more effectively and cut costs.

1.2 IBM Big Data and Analytics platform

All big data use cases require an integrated set of technologies to fully address the business pain they aim to alleviate. Due to this complexity, enterprises need to start small, with a single project, before moving on to other issues and pursuing added value. IBM is unique in having developed an enterprise-class big data platform that allows you to address the full spectrum of related business challenges.

The IBM Big Data and Analytics platform gives organizations a solution stack that is designed specifically for enterprise use. The IBM Big Data and Analytics platform provides the ability to start small with one capability and easily add others over your big data journey because the pre-integration of its components reduces your implementation time and cost.

The IBM Big Data and Analytics platform addresses the following key enterprise requirements:

- The *5 Vs*: The platform includes functionality that is designed to help with each of the 5 Vs:
  - Variety: The platform supports wide variety of data and enables enterprises to manage this data *as is*, in its original format, and with extensive transformation tools to convert it to other desired formats.
  - Velocity: The platform can handle data at any velocity, either low-latency streams, such as sensor or stock data, or large volumes of batch data.
  - Volume: The platform can handle huge volumes of at-rest or streaming data.
  - Veracity: The platform includes various tools to remove uncertainty about the target data.
  - Visibility: The platform provides the ability to discover, navigate, and search over a broad range of data sources and types, both inside and outside your enterprise.
- Analytics:
  - The platform provides the ability to analyze data in its native format, such as text, binary, and rich multimedia content.
  - The platform can scale to analyze *all* of your data, not just a subset.
  - The platform enables *dynamic analytics,* such as automatic adjustments and actions. For example, streaming video service companies use users' past viewing behavior to generate new recommendations, and use this recommendation data in real time to provision greater capacity for improved viewing.
- Ease of use:
  - The platform includes a deep set of developer user interfaces (UIs), common languages, and management consoles. This Eclipse-based development environment enables faster adoption and reduces the time spent in coding and debugging.
  - The platform also provides user UIs and visualization capabilities, such as web-based analysis and visualization tools with familiar, spreadsheet-like interfaces.
- Enterprise-ready

  The platform has capabilities for fault tolerance across the solution stack, including enterprise-grade security and privacy features.
- Integration

  The platform provides the ability to integrate with a wide variety of data sources using industry-standard protocols, such as Open Database Connectivity (ODBC), Java Database Connectivity (JDBC), and Java Message Service (JMS).

Figure 1-2 shows an overview of the IBM Big Data and Analytics platform.

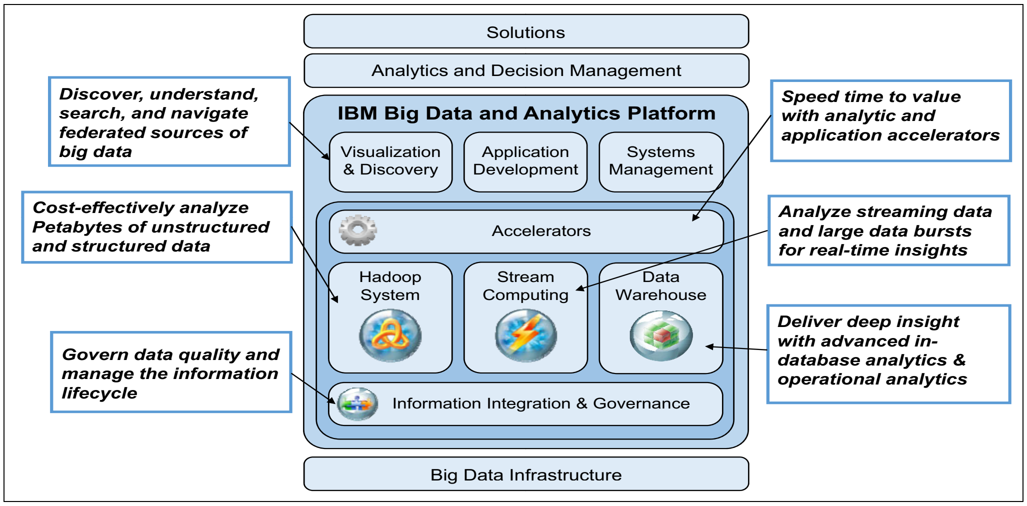

Solutions
Analytics and Decision Management
Discover, understand, search, and navigate federated sources of big data
IBM Big Data and Analytics Platform
Speed time to value with analytic and application accelerators
Visualization & Discovery
Application Development
Systems Management
Cost-effectively analyze Petabytes of unstructured and structured data
Accelerators
Analyze streaming data and large data bursts for real-time insights
Hadoop System
Stream Computing
Data Warehouse
Govern data quality and manage the information lifecycle
Deliver deep insight with advanced in-database analytics & operational analytics
Information Integration & Governance
Big Data Infrastructure

Figure 1-2 IBM Big Data and Analytics platform

As depicted in Figure 1-2, the IBM Big Data and Analytics platform uses the underlying big data infrastructure, which is typically either x86 or Power servers, for running the Hadoop system and Streams components, and data warehousing appliances.

The Hadoop system provides a cost-effective way to store large volumes of structured and unstructured data in one place for deep analysis. IBM provides a non-forked, open source Hadoop version and augments it with capabilities, such as enterprise-class storage (by using an IBM General Parallel File System (GPFS™)), security (by reducing the surface area and securing access to administrative interfaces and key Hadoop services), and workload optimization (using the Adaptive MapReduce algorithm that optimizes execution time of multiple small and large jobs).

The Stream Computing ability is designed to analyze data in motion while providing massive scalability and processing of multiple concurrent input streams. The IBM Streams platform can process and analyze a wide variety of structured and unstructured data and video and audio content.

The Data Warehousing component is provided by IBM workload-optimized systems that are delivered as deep analytical appliances, with a massive parallel processing engine and the ability to handle mixed operational and analytic workloads.

The Information Integration and Governance layer gives the IBM Big Data and Analytics platform the ability to integrate with any type of data. It also provides governance and trust for big data by using capabilities, such as security sensitive data, tracking data lineage, lifecycle management to control big data growth, and master data to establish a single source of truth.

The User Interfaces in the IBM Big Data and Analytics platform are tailored for three classes of users (business users, developers, and administrators), with different types of tooling for each class. Business users can analyze a wide variety of data in an ad hoc manner using a browser-based interface and spreadsheet-style interface for exploring and visualizing data.

Developers have access to various APIs and useful development environments, such as Eclipse. Developers also have access to many data-parallel algorithms, such as those algorithms for regression modeling and dimensionality modeling.

Administrative users have access to consoles to aid in monitoring and managing the systems and components of the IBM Big Data and Analytics platform.

The IBM Big Data and Analytics platform provides a number of accelerators, such as Analytics accelerators (to handle text data, mining data, and acoustic data) and Industry and Horizontal Application accelerators, such as pre-configured analytics for processing CDRs for telecom clients, and streaming options trading for financial clients.

Finally, the IBM Big Data and Analytics platform is designed for analytic application development and integration with a wide variety of third-party applications for business intelligence, predictive analytics, content analytics, and so on.

Figure 1-3 maps the platform's components to the 5 Vs of big data.

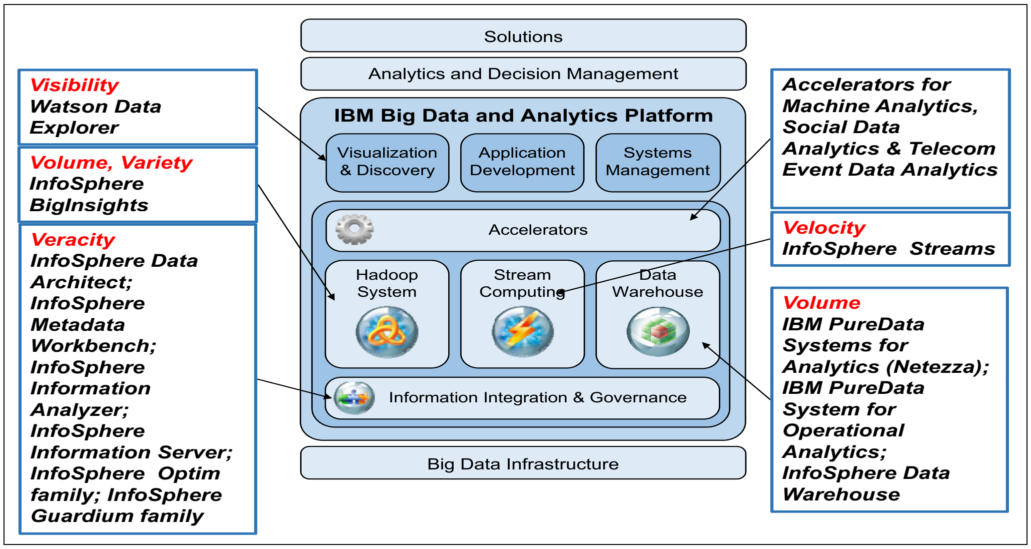

Solutions
Analytics and Decision Management
IBM Big Data and Analytics Platform
Visualization & Discovery
Application Development
Systems Management
Accelerators
Hadoop System
Stream Computing
Data Warehouse
Information Integration & Governance
Big Data Infrastructure
Visibility
Watson Data Explorer
Volume, Variety
InfoSphere BigInsights
Veracity
InfoSphere Data Architect; InfoSphere Metadata Workbench; InfoSphere Information Analyzer; InfoSphere Information Server; InfoSphere Optim family; InfoSphere Guardium family
Accelerators for Machine Analytics, Social Data Analytics & Telecom Event Data Analytics
Velocity
InfoSphere Streams
Volume
IBM PureData Systems for Analytics (Netezza); IBM PureData System for Operational Analytics; InfoSphere Data Warehouse

Figure 1-3 IBM Big Data and Analytics platform mapped to the 5 Vs of big data

Table 1-1 lists each of the IBM products associated with the IBM Big Data and Analytics platform and maps each product to its role in the process and the underlying big data characteristics that is being addressed.

Table 1-1 IBM Big Data and Analytics platform components and product functions

Product and supporting URL	Big data characteristics and product functions
IBM InfoSphere® BigInsights™ http://www.ibm.com/developerworks/bigdata/biginsights/index.html	Volume, Variety This mature, Hadoop-based solution for big data analytics aids in managing and analyzing massive volumes of structured and unstructured data at rest. InfoSphere BigInsights augments Hadoop with enterprise capabilities, including advanced analytics, application accelerators, multi-distribution support, performance optimization, enterprise integration, and more.
IBM PureData™ System for Hadoop http://www.ibm.com/software/data/puredata/hadoop	Volume PureData System for Hadoop gives you Hadoop data services optimized for big data analytics and online archiving with the simplicity of a custom appliance.

Product and supporting URL	Big data characteristics and product functions
IBM PureData System for Analytics http://www.ibm.com/software/data/puredata/analytics/system/	Volume PureData System for Analytics provides data warehouse services optimized for high-speed, Peta-scale analytics and simplicity.
IBM PureData System for Operational Analytics http://www.ibm.com/software/data/puredata/analytics/operational/	Volume With PureData System for Operational Analytics, you get operational data warehouse services optimized to balance high-performance analytics and real-time operational throughput.
IBM InfoSphere Warehouse http://www-01.ibm.com/software/data/db2/warehouse-editions/	Volume InfoSphere Warehouse provides a comprehensive data warehouse platform that delivers access to structured and unstructured information in real time.
IBM InfoSphere Streams http://www.ibm.com/developerworks/bigdata/streams/index.html	Velocity InfoSphere Streams helps you analyze many data types simultaneously and perform complex calculations in real time.
IBM InfoSphere Data Explorer (Watson™ Explorer) http://www.ibm.com/software/data/infosphere/data-explorer/	Visibility InfoSphere Data Explorer enables federated navigation and discovery across a broad range of big data sources.
IBM InfoSphere Data Architect http://www.ibm.com/software/data/optim/data-architect/	Visibility With InfoSphere Data Architect, you can discover, model, visualize, relate, and standardize diverse and distributed data assets across the enterprise.
IBM InfoSphere Information Analyzer http://www.ibm.com/software/data/infosphere/information-analyzer/	Veracity Profile the quality of data from source systems, and the systems themselves, using InfoSphere Information Analyzer.
IBM InfoSphere Information Server http://www.ibm.com/software/data/integration/info_server/	Veracity InfoSphere Information Server is a data integration platform that helps you understand, cleanse, transform, and deliver trusted information to your critical business initiatives, such as big data, master data management, and point-of-impact analytics.
IBM InfoSphere Information Server for Data Quality http://www.ibm.com/software/data/integration/info_server/data-quality/	Veracity With InfoSphere Information Server for Data Quality, you can cleanse data and monitor data quality, turning data into trusted information.

Product and supporting URL	Big data characteristics and product functions
IBM InfoSphere Master Data Management Family http://www.ibm.com/software/data/master-data-management/	Veracity The InfoSphere Master Data Management Family provides an operational system of record that creates and maintains a single version of the truth for key business entities, such as customers, patients, products, parts, suppliers, accounts, and assets, among others.
IBM InfoSphere Optim™ Family http://www-01.ibm.com/software/data/optim/	Veracity The InfoSphere Optim Family provides data growth management and archiving of complete business objects across heterogeneous environments, while also enabling easy retrieval of archived information.
IBM InfoSphere Guardium® Family http://www.ibm.com/software/data/guardium/	Veracity The InfoSphere Guardium family helps ensure the privacy and integrity of trusted information in your data center.

1.3 IBM Cloud Computing Reference Architecture

The IBM Cloud Computing Reference Architecture (CCRA) is a blueprint or guide for architecting cloud computing implementations. It is based on years of experience of IBM personnel working with customers on cloud computing solutions, and is driven by a broad set of functional and nonfunctional requirements collected during those engagements.

The IBM CCRA (see Figure 1-4) provides guidelines and technical work products, such as service and deployment models, and defines numerous overarching adoption patterns. An *adoption pattern* embodies the architecture patterns that represent the ways that organizations typically implement cloud computing solutions.

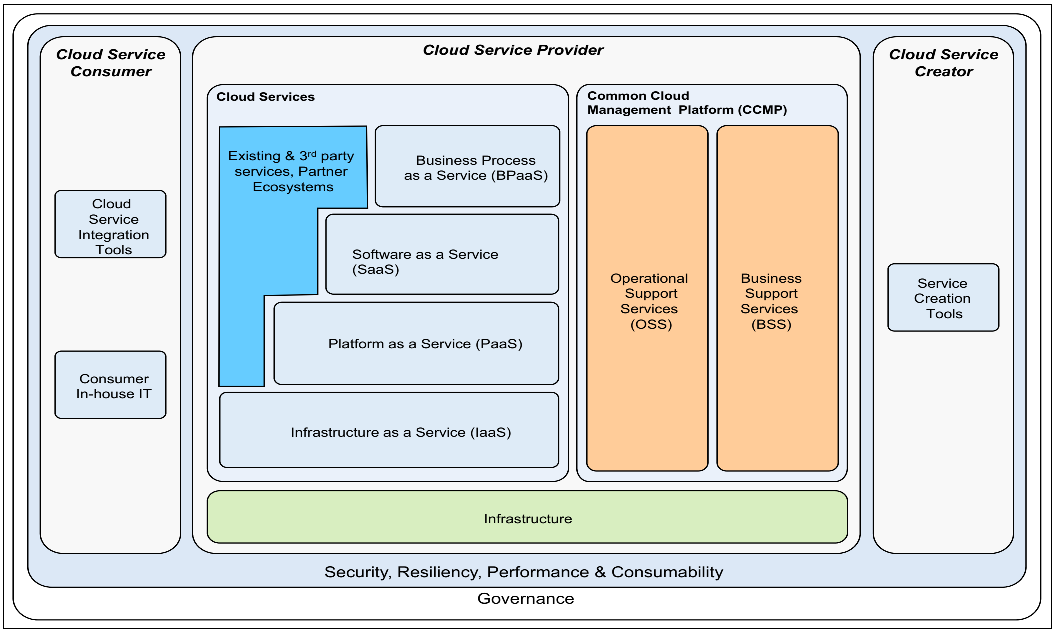

Cloud Service Consumer
Cloud Service Integration Tools
Consumer In-house IT
Cloud Service Provider
Cloud Services
Existing & 3rd party services, Partner Ecosystems
Business Process as a Service (BPaaS)
Software as a Service (SaaS)
Platform as a Service (PaaS)
Infrastructure as a Service (IaaS)
Common Cloud Management Platform (CCMP)
Operational Support Services (OSS)
Business Support Services (BSS)
Infrastructure
Cloud Service Creator
Service Creation Tools
Security, Resiliency, Performance & Consumability
Governance

Figure 1-4 High-level view of the IBM Cloud Computing Reference Architecture (CCRA)

This paper does not delve into the details of cloud computing. However, we can summarize a cloud computing environment as a data center orchestrator that provides these functions:

- Data storage (block or object storage)
- Data processing (compute resources)
- Data interchange (networking)

Cloud computing environments work on the principle of shared resource pools and exhibit these characteristics:

- Ability to handle failures, such as by moving a workload off a failed VM.
- Support for large workloads. Cloud computing environments are built to scale and new capacity can be provisioned quickly.
- Programmable. Cloud computing environments typically provide APIs to connect and orchestrate all aspects of the workloads running in the cloud.
- Utility computing, also called a *pay-as-you-go model*, with an illusion of infinite resources. Cloud computing environments require no up-front cost and enable fine-grained billing (such as hourly billing).

For this publication, the phrase *cloud computing* is used interchangeably with the phrase *infrastructure as a service (IaaS)*.

1.4 Big data and analytics on the cloud: Complementary technologies

Big data and analytics require large amounts of data storage, processing, and interchange. The traditional platforms for data analysis, such as data warehouses, cannot easily or inexpensively scale to meet big data demands. Furthermore, most of the data is unstructured and unsuitable for traditional relational databases and data warehouses.

Platforms to process big data require significant up-front investment. The methods for processing big data rely on parallel-processing models, such as MapReduce, in which the processing workload is spread across many CPUs on commodity compute nodes. The data is partitioned between the compute nodes at run time, and the management framework handles inter-machine communication and machine failures. The most famous embodiment of a MapReduce cluster, Hadoop, was designed to run on many machines that don’t share memory or disks (the *shared-nothing* model).

Alternatively, cloud computing is the perfect vehicle to scale to accommodate such large volumes of data. Cloud computing can *divide* and *conquer* large volumes of data by using *partitioning* (storing data in more than one region or availability zone). Furthermore, cloud computing can provide cost efficiencies by using commodity compute nodes and network infrastructure, and requiring fewer administrators (thanks to standardizing the available offerings through the Cloud Service catalog), and programmers (through the use of well-defined APIs). However, cloud computing environments are built for general-purpose workloads and use resource pooling to provide elasticity on demand.

So it seems that a cloud computing environment is well-suited for big data, provided the shared-nothing model can be honored. But there is another large difference, the acute volatility of big data workloads compared to typical workloads in a cloud computing environment.

1.5 A large difference

Figure 1-5 shows two graphs. The graph on the left represents a typical cloud workload. The graph on the right shows a big data workload. The typical cloud uses a few resources over a long period compared to the big data workload that uses many resources over a shorter period.

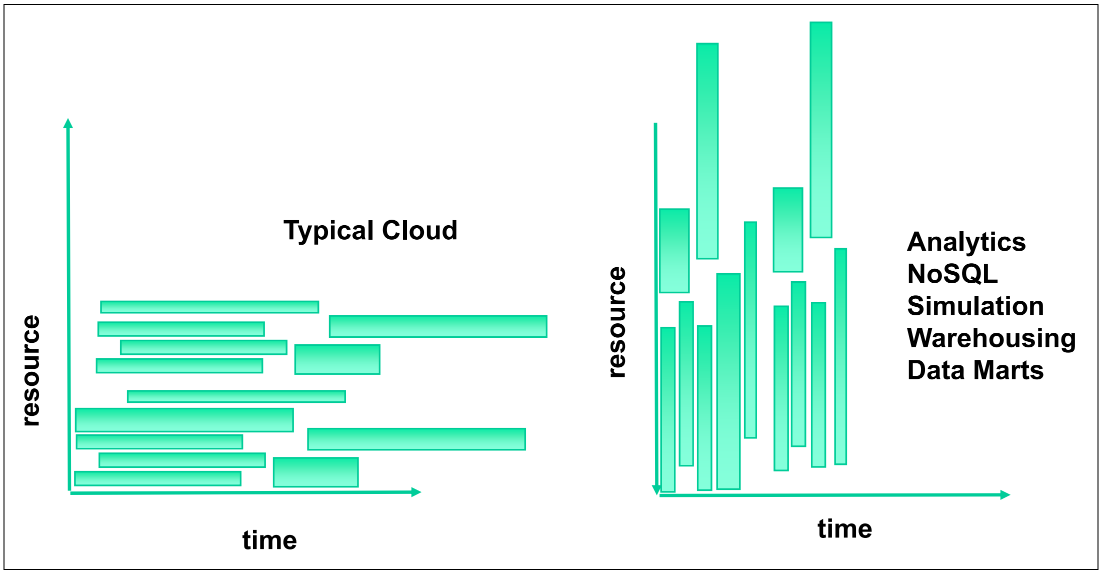


Figure 1-5 Typical cloud computing workloads versus big data workloads

Typical cloud computing workloads have volatility (certain workloads might be decommissioned after a time), but the duration of each workload is typically a few weeks to a few months. For example, communications providers often provision additional capacity to handle abnormally high workloads when new devices are launched. This new workload can require twice the provider's existing capacity, but the need might exist for just three to four weeks.

If you mentally rotate the figure counterclockwise by 90°, you get a representation of big data workloads. These workloads typically exist for less than a week (sometimes just a few hours), require massively large compute and storage capacity, and are decommissioned when the need is over. For example, an advertiser might analyze all micro-messages (such as Twitter messages) that are posted during a large public event to gauge public sentiment about its products and then change follow-up advertisements based on how the initial advertisements were received.

This differentiation has obvious implications. A big data cloud computing environment needs extreme elasticity to provision hundreds of virtual machines (VMs) in hours or minutes. Dedicated and isolated networks are required to ensure that data replication between nodes does not affect the ingestion of incoming data.

1.6 Implications for the cloud

Many aspects of big data require changes to the underlying IaaS cloud:

- Support for mixed workloads

  Traditionally, IaaS clouds are designed for general-purpose workloads (such as middleware, application servers, and databases) and work on the principle of consolidation and resource sharing. However, big data and analytics workloads require special hardware, especially with the shared-nothing architecture of big data systems. Certain big data systems rely on massively parallel processing and fast disk I/O. Other solutions rely on in-memory analytics to address requirements for real-time analysis. Both of these scenarios require specific hardware that is not found in typical IaaS clouds.

- Scalability

  The vast nature of big data is an easily recognized challenge. An existing IaaS cloud will almost certainly need to be modified to provide the level of performance and data durability that enterprises require when processing ever-increasing volumes of data. This requires a fundamental shift in the design of IaaS clouds. The implications range from the size of the cloud (number of compute nodes, number of disks per node, and so on) to the type of network interfaces used, the type of compute nodes used, and the separation of traffic between the compute nodes.

- Rapid elasticity

  Big data even stretches the limits of elasticity provided by a traditional cloud. It might not be sufficient to provision a VM in under an hour. Big data systems might require several hundred VMs to be provisioned in a matter of minutes.

- Networking

  An efficient network is crucial for a big data cluster. Networks in an IaaS cloud must be designed to provide resiliency with multiple paths between compute nodes for rapid data transfer, and they must be able to scale to handle the larger data volumes and throughput required to support big data. Dedicated network paths might be required for different types of data. For example, separate paths might be required for the interchange of user data, big data, and management data. In addition, IaaS cloud providers might need to invest in expanding access networks to allow faster rates of data ingestion, and new tooling to guarantee that they meet their QoS commitments.

- Multi-tenancy

  Isolation of workloads (or the VMs the workloads are running in) gains greater importance in big data systems, where performance is critical and the impact from a disruptive neighboring workload might be too costly. In addition to isolating VMs, big data solutions require isolation at the worker and data node levels, too. This is accomplished by using either a distributed resource broker, such as Apache Mesos, or a cluster resource manager framework, such as Apache YARN. This resource isolation also enables tiered service level agreements (SLAs) so that, for example, production workloads can be given higher priority for resources while development and test workloads get lower priority.

1.7 Big data and analytics in the cloud: Bringing the two together

So, if the cloud computing environment can be modified correctly, big data and cloud can come together in beneficial ways:

- The cloud engine can act as the orchestrator providing rapid elasticity.
- Big data solutions can serve as storage back ends for the cloud image catalog and large-scale instance storage.
- Big data solutions can be workloads running on the cloud.

Yet, for big data and the cloud to work together, many changes to the cloud are required:

- CPUs for big data processing

  A Graphics Processing Unit (GPU) is a highly parallel computing device originally designed for rendering graphics. GPUs have evolved to become general-purpose processors with hundreds of cores. They are considered more powerful than typical CPUs for executing arithmetic-intensive (versus memory-intensive) applications in which the same operations are carried out on many data elements in parallel fashion. Recent research has explored tightly integrating CPUs and GPUs on a single chip. So, one option is to create a resource pool with special compute chips for high performance computing for big data.

  Another option of boosting computing capacity for big data in the cloud is to create a resource pool with multi-core CPUs, which can achieve greater performance (in terms of calculations per second) for each unit of electrical power that is consumed than their single-core equivalents. With quad-core and hex-core CPUs now commonplace, this is the most attractive and cost-effective way to create dedicated resource pools for big data processing in a cloud.

- Networking for big data processing

  With a need to handle, potentially, petabytes of multi-structured data with unknown and complex relationships, the typical network design in a cloud infrastructure is no longer sufficient. Special considerations are required for ingesting the data into a Hadoop cluster, with a dedicated network to allow parallel processing algorithms, such as MapReduce, to shuffle data between the compute nodes.

  A minimum of the following types of network segments are required:

  – Data: Dedicated to MapReduce applications with a bandwidth of 10 GB for lower latency and higher bandwidth

  – Admin: A separate and dedicated network for management of all compute nodes and traffic not related to MapReduce

  – Management: A platform for an Integrated Management Module (IMM) (can optionally share the VLAN subnet with the Admin segment)

- Storage for big data processing

  One of the biggest changes is to the storage subsystem. These changes can be addressed in two ways:

  – Disk Attached Storage (DAS): The compute nodes are designed with multi-core commodity hardware with a large array of local disks. The local disks do not employ RAID and are used as just a box of disks (JBOD). In this case, built-in redundancy of big data file systems, such as Hadoop Distributed File System (HDFS), is used because they are replicating blocks across multiple nodes.

  – A second option is to use a new type of storage architecture that allows storing and accessing data as objects instead of files. Rather than using traditional enterprise storage (such as storage area network (SAN) or network-attached storage (NAS), which is then pooled and provisioned dynamically), IaaS clouds are extended to provision rack-aware workloads and include support for object storage. Refer to Figure 1-6. A typical IaaS cloud is depicted on the left and an IaaS cloud that can support big data workloads is shown on the right. The object storage support of the expanded IaaS cloud enables big data workloads to scale horizontally and allows the stored objects to be accessed by any node in the server racks using a fully qualified Uniform Resource Identifier (URI). However, the primary intent of this storage is *not* to be used inside VMs for data processing.

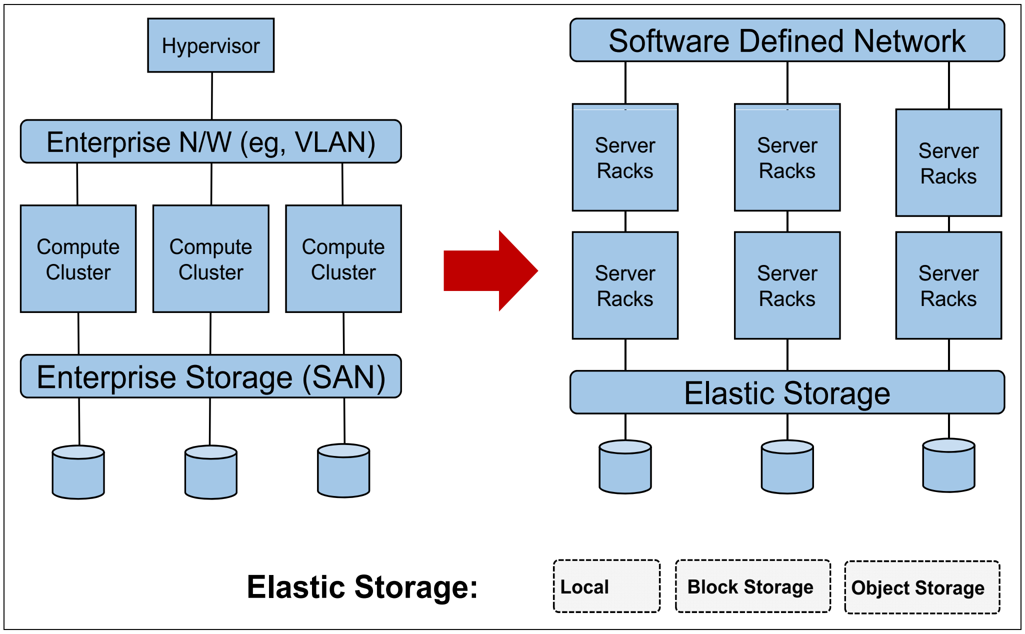


Figure 1-6 Traditional versus cloud deployments for big data

In this scenario, IaaS cloud storage can be thought of in terms of *primary* and *secondary* storage:

- Primary storage, also called *boot storage*, is any type of storage that can be mounted on the node of a cluster. It also holds the disk images of running VMs and user data.
- Secondary storage, or *object storage*, is a different way of storing, organizing, and accessing data on disk. An object storage platform provides a storage infrastructure to store files with significant metadata added to them (the files are then referred to as *objects*). The back-end architecture of an object storage platform is designed to present all of the storage nodes as a single pool. With object storage, there is no file system hierarchy. The cloud environments might implement object storage by adopting OpenStack Swift software. Object storage typically provides data protection by making multiple copies of the data (for example, the *three copies* of data paradigm promoted by public cloud vendors, such as Amazon and Rackspace).

With these changes to the cloud architecture, we can finally bring together big data and cloud computing. The following scenarios are now possible:

- Provision a Hadoop cluster on bare-metal hardware
- Operate a hybrid cloud (part hypervisor for VM provisioning, part bare metal for the data store), which is the most common cloud configuration today
- Reconfigure the entire cloud on demand

2

Big data and analytics business drivers and technical challenges

This section describes the business drivers that are leading more enterprises to manage big data in the cloud. The challenges such organizations face in harnessing the full value of the technology are also described.

2.1 Big data business drivers

Enterprises are keen on developing new business models to more efficiently manage, discover, navigate, and analyze mission-critical big data. They want to harness their organizational big data to improve employee morale and productivity; understand customer preferences and behaviors (to improve the services they sell and retain customer loyalty), and improve their own organizational performance to keep up with market trends.

The big data goals of these organizations fall into several business categories:

- Revenue:
  - Monetize big data: Design and execute big data analytics use cases that increase revenue, lower costs, or reduce risk.

    Critical success factors: Metrics for revenue (value), cost savings, and risk reduction.
  - Manage big data at a low cost: Demonstrate cost savings of big data analytics styles for both MapReduce clusters and real-time analytics.

    Critical success factor: Understand the cost savings that can be achieved with MapReduce clusters, such as Hadoop, and real-time analytics when compared to traditional IT solutions, such as data warehouses and storage area network (SAN) storage.
  - Improve efficiency in business operations: Develop insight about the value of specific business processes, such as enterprise resource planning (ERP), supply chain management (SCM), and customer relationship management (CRM).

    Critical success factor: Define and categorize the big data types that are available for mining to improve ERP, SCM, and CRM.
- Customer services:
  - Improve customer understanding (360-degree view of the customer): Mine all sources of client experience and interaction from additional unstructured and semi-structured data types using real-time and batch (Hadoop) analytics.

    Critical success factor: Metrics for sentiment analysis, client satisfaction, and client experience improvement.
  - Obtain behavioral insight into client transactions: What led to a certain business transaction? Why did the client choose us? What else can we deduce about a client’s buying behavior?

    Critical success factor: Obtain client information within the parameters of privacy for the client.
  - Attract and retain customers: Mine and apply insight toward marketing and sales effectiveness with clients, customers, and customer support personnel.

    Critical success factor: Capture broader information about customer buying behavior and preferences.
  - Fraud detection and claims processing: Derive and exploit additional insight from data types not previously analyzed for anti-fraud and claims processing.

    Critical success factor: Mine fraudulent activity patterns from batch and real-time data.

- Business development:
  - Introduce new products or services: Thanks to your new insight about target market preferences, new products and services will have higher adoption rates by the target clientele.

    Critical success factor: Collect feedback from new product and service introductions.
  - Outsource non-core functions: Decide what to outsource without affecting the customer experience.

    Critical success factor: Design and implement analytics for core versus non-core functions using big data analytics techniques.
  - Pursue mergers, acquisitions, and divestitures: Gather and consider marketplace insights about the potential impact of mergers, acquisitions, and divestitures.

    Critical success factor: Identify your target marketplace and perform sentiment analysis of what is working and what is not.
  - Gain new competitive insights: Mine all sources of information, even non-traditional sources of information, to learn about the brand perception of the company by its customers, its reputation, and its industry ranking.

    Critical success factor: Define metrics for improvement that are achievable if based on better insight.
- Business agility and governance:
  - Increase business agility: Mine real-time events for trends and apply the insight to transactions and interactions with customers.

    Critical success factor: Apply what is learned in an iterative manner to steadily improve the client experience.
  - Plan with greater confidence: Build better scenario-based analysis models.

    Critical success factor: Provide big data analytics capabilities to business analysts and solution designers.
  - Make better decisions faster: Harvest better insights from both batch (Hadoop) and real-time events and rapidly make them available to decision makers.

    Critical success factor: Grow big data analytic skills in the functional areas dedicated to decision management.
  - Ensure regulatory compliance: Improve your understanding of the current regulatory climate and expectations of auditors.

    Critical success factor: Improved traceability of all relevant records.
  - Lower risk: Improve the cost-benefit analysis of various risks (regulatory, market, credit, counter-party operational, and so on).

    Critical success factor: Reduction in penalties and outages, and organization risk rating by external agencies.

Another key business driver for big data is IT and operational optimization. To optimize these areas, develop a big data strategy that uses existing enterprise investments, including data and applications.

A well-integrated data architecture is needed to support an advanced analytics platform. Although it is a huge challenge to integrate different types of data sources across an organization, it is a necessity. But this integration *can* be accomplished, often in phases, with the help of available tools and software. A good strategy, combined with a well-integrated collection of data, will provide a strong foundation for big data analytics.

2.2 Technical challenges

Although many technological innovations are making big data a popular topic, it is still difficult to fully exploit and benefit from it. The *5 Vs* that define the benefits of big data also bring inherent challenges:

- The ability to store and manage large volumes of data
- The ability to handle the expanding data and assuring its trustworthiness
- Adopting methods to integrate, federate, and correlate different types of data and use them for enterprise applications
- Adopting massive parallel processing technologies to rapidly process the data for timely results
- Applying industry-standard extract, transform, and load (ETL), data quality, security, Master Data Management (MDM), and lifecycle management methods to big data

Table 2-1 summarizes the aspects of big data lifecycle management that remain a challenge for most companies.

Table 2-1 Big data technical challenges

Challenge	Details	Critical success factors
Large volumes of data	Storing data at a petabyte scale is expensive.	Communicate costs accurately and benchmark the expected results.
Development of new applications is too difficult and time-consuming in Hadoop, NoSQL, and other new technologies	Data scientist skills are rare and expensive. Open source and Java skills might not be available for Hadoop, NoSQL, Cassandra, and so on.	Match big data visualization tools to the needs of the analyst community.
High infrastructure and maintenance costs	At some point, a big data IT infrastructure must be put in place (with associated production costs).	Develop a cost-benefit analysis of the trade-offs of big data versus traditional data IT infrastructures and communicate this information to stakeholders.
Creating another data silo	Big data storage is sometimes viewed as cheap storage dumps and might create another data repository in the organization.	Define and communicate a vision of the big data ecosystem that integrates with the enterprise data architecture.
Data quality and security	The lifecycle of big data, from raw input sources to valuable insights, requires rigorous data quality and security standards.	Ensure data privacy and confirm that an audit trail exists for all transactions.
Fulfilling service level agreements (SLAs) for performance and availability	Processing petabyte-scale data to provide reports in acceptable time frames requires greater investment in big data storage architecture.	Define and communicate SLAs in a realistic manner and establish accountability for them throughout the development-test-production lifecycle.
Lack of clarity on which types of analytics give the needed insights	Involved personnel need to understand statistical algorithms, how to apply them, and how they are improved by additional data.	Assess and improve the needed skills for big data success.

Challenge	Details	Critical success factors
Lack of integration with data sources	A meta model is often needed to provide a composite view of multiple, disparate big data sources.	Design and implement data integration to provide necessary viewpoints, such as dashboards, enterprise service bus (ESB) for big data, indexes, and data dictionaries.

The significance of these big data technical challenges will vary depending on where on the adoption path the organization currently is, such as exploration, adoption, or implementation.

Many data stakeholders believe that big data holds information that is more insightful than what they can extract from their traditional analytical systems. So, they are eager to embrace the concepts and initiatives of big data. Yet these organizations cannot simply apply the traditional approaches of managing and extracting information to big data. They have to understand that the 5 Vs of big data pose the kinds of technology challenges that were just stated. And they have to adopt the methods, models, processes, and technologies needed to tame and harness big data.

The big data skill gap can be a hindrance as well. These projects must be managed and performed by big data subject matter experts and highly skilled data experts. The involved personnel must have prior Information Management expertise and a deep understanding of the current big data technologies.

3

Big data and analytics taxonomy

Here we introduce the concept of a big data *taxonomy*, or classification system, to more easily distinguish among the various big data-related challenges and enable clients to quickly understand the types of problems they have. The technology stack required to overcome one big data challenge can vary greatly from what is needed to face a different challenge, even within the same industry.

We need a big data taxonomy for several reasons:

- Disambiguation of big data (Is it high volume, high velocity, or something else?)
- Easy identification of required tooling (for example, Hadoop or Netezza®)
- Classification of the problem domain based on usage, such as entity analytics or semantic analytics

For the authors' findings about various combinations of big data characteristics, see Table 3-1 on page 22.

Each row in the table indicates a different use case. The use cases are classified by the type of big data involved, and the different types are summarized after the table.

The first four columns of the table indicate the applicability of the 5 Vs of big data (Visibility is omitted from this analysis because it does not have a direct impact on the type of big data).

The Horizontal Scalability column denotes whether processing the use case data requires the solution to scale horizontally. This decision is often dictated by the type of data. For example, if volume is low, horizontal scalability is not required.

The SQL limitation column denotes whether there is a limitation in using Structured Query Language (SQL) for accessing the data. Although you might think that SQL limitations will only exist when there is significant variety in the data, data volumes in the multi-petabytes can introduce SQL performance limitations.

The Big data column reveals our conclusion about whether each particular use case can be classified as big data.

The Comments column provides a brief summary of the type of data problem involved in each use case and how it can be resolved.

Certain rows in the table are highlighted because they are used to illustrate additional points that are discussed later.

Table 3-1 Big data taxonomy

Volume	Velocity	Variety	Veracity	Horizontal scalability?	SQL limitation?	Big data?	Comments
No	No	No	Yes	No	No	No	Data validation is required.
No	No	No	No	No	No	No	Data validity, lineage, and provenance need to be established.
No	No	Yes	No	May	May	May	Data validity and non-relational data store are required.
No	Yes	No	No	May	No	May	Continuous ingest capability is required.
Yes	No	No	No	Yes	May	Type 1	Traditional data warehouse.
No	No	Yes	Yes	May	Yes	Type 3	Non-relational data repository is required.
No	Yes	Yes	Yes	Yes	Yes	Types 3 and 4	Non-relational data repository with continuous data ingestion.
Yes	Yes	No	No	Yes	May	Types 1 and 4	Continuous ingest capability with horizontal scalability of relational data repository with data validation.
Yes	Yes	Yes	No	Yes	Yes	Types 1, 3, and 4	Continuous ingest with horizontal scalability of non-relational data repository.
Yes	Yes	No	Yes	Yes	May	Types 1 and 4	Continuous ingest with horizontal scalability of relational data repository.

Volume	Velocity	Variety	Veracity	Horizontal scalability?	SQL limitation?	Big data?	Comments
Yes	Yes	Yes	Yes	Yes	Yes	Types 1, 3, and 4	Continuous ingest with horizontal scalability of non-relational data repository with data validation.
Yes	Yes	Yes	No	Yes	Yes	Types 2 and 4	Highly recursive social data with continuous ingest, non-relational data store, and data validation.

Based on the use cases in Table 3-1 on page 22, we can establish four major categories or types of big data:

- Type 1: Structured data, and lots of it. Today, this data set is being addressed by horizontally scaling data warehouses. Yet in some cases, the data volumes and usage can be so great that the traditional warehouses might be unable to cope with it.
- Type 2: Big graphs (hierarchical data). This is the most complex big data type and is typically found while analyzing data with nested interrelationships (such as in social media connection data, or intersecting geo-location data).
- Type 3: Semi-structured data, which, as the name suggests, cannot be easily converted to a relational format. This data type accounts for the bulk of data today. It includes anything from audio and video streams to binary data from sensors.
- Type 4: Transactional streams. This is streaming data that requires continuous ingest. If this data is only relational, some of it can be addressed by databases that are available today. For non-relational streams of data, a new mechanism is required.

Scanning Table 3-1 on page 22 by row, you can quickly identify the type of problem. We highlighted two rows in the table to illustrate how to interpret the information provided:

- Type 1: In this scenario, the traditional Relational Database Management System (RDBMS) or data warehouse might suffice, provided it scales horizontally and cost-effectively. This is the typical data warehouse of today.
- Type 3 and type 4: This type of big data involves semi-structured data that is coming in continuously. In this scenario, a typical data warehouse will not suffice and a NoSQL store will be required to store the data (including continuous ingest capability).

Table 3-1 on page 22 is a handy guide for understanding the various big data problems you are facing and selecting the best tools to develop a solution.

4

Big data and analytics architecture

There are many Information Management (IM) architectures available in the public domain, including the IBM IM Reference Architecture (RA). These RAs and similar frameworks are the best guide for building any data processing system.

Yet we have noticed that a common set of functions is required within the context of big data. When organizations embark on the big data journey, they already have many of these tools in their business intelligence (BI) and data warehouse environments, so the real need is to complement these tools with important additional tools. Another requirement is for the new tooling to coexist with the existing BI environment, so that the data can flow seamlessly. This is also where *maturity models* are used.

4.1 Maturity models

Big data is a journey that requires new technologies, governance, analytics, and organizational components to successfully harness hidden insights. Many maturity models that are specifically designed for big data describe the stages of maturity of the processes that most organizations follow when they embark on big data initiatives. The following model is one of the most popular maturity models:

- The Data Warehousing Institute specifies a maturity model with five stages: Nascent, Pre-adoption, Early Adoption, Corporate Adoption, and Mature (or Visionary). It also provides an assessment tool at this website:

  http://tdwi.org/bdmm

However, except for several new skills and hardware requirements, big data competency is no different from competency in traditional BI and data warehouse environments. So, here we introduce a simpler, staged maturity model that can be used to measure an organization's big data processes by capability levels.

Figure 4-1 illustrates our five-level maturity model for big data environments.

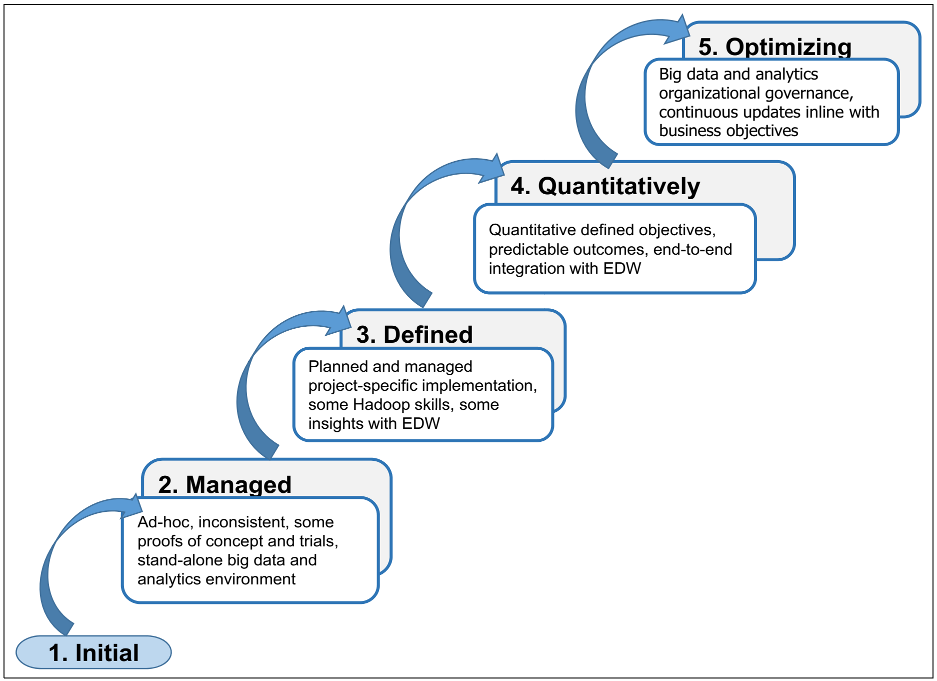

5. Optimizing
Big data and analytics organizational governance, continuous updates inline with business objectives
4. Quantitatively
Quantitative defined objectives, predictable outcomes, end-to-end integration with EDW
3. Defined
Planned and managed project-specific implementation, some Hadoop skills, some insights with EDW
2. Managed
Ad-hoc, inconsistent, some proofs of concept and trials, stand-alone big data and analytics environment
1. Initial

Figure 4-1 Big data and analytics maturity model

This new maturity model consists of the following levels:

- Level 1 (Initial)

  All organizations are at this level. There are traditional BI and data warehouse systems processing data, with relational databases potentially reaching their limits. At this stage, the business is looking for new, faster ways to gain insight about its data.

- Level 2 (Managed)

  In this stage, processes exist for big data systems, but they are ad hoc, inconsistent, and not repeatable. Several proofs of concept (POC) or trials are in place. But such systems are stand-alone and not integrated with the rest of the information management or data warehouse environments, and existing tools might or might not be reused.

- Level 3 (Defined)

  Businesses at this stage have a standardized set of processes and procedures to undertake big data projects within the organization. Big data initiatives are tied to a business objective, with integration planned with the information management and data warehouse environments. Existing tools are reused where possible.

- Level 4 (Quantitatively managed)

  Here, quantitatively measurable objectives are defined for the business's big data environments. The processes and procedures are repeatable and ensure the successful implementation of big data initiatives. End-to-end integration exists with the rest of the data warehouse and information management environments. Existing tools are reused in every possible instance.

- Level 5 (Optimizing)

  In this final stage of evolution, big data governance exists at the corporate level and is tied to other data and information management governance processes. The big data technologies are continuously updated inline to extract maximum analytical value.

We have deliberately stayed away from tying this new maturity model to any specific technology or tooling because we view this model as a complement to the existing information management and data warehouse environments. This alternate maturity model can easily be mapped to existing data management maturity models, such as the Data Management Book of Knowledge (DMBOK), the Method for an Integrated Knowledge Environment (MIKE) 2.0 Information Model, and the IBM Data Governance Council Maturity Model.

4.2 Functional architecture

In this section, we introduce the most common set of functional capabilities required for processing big data. We define a *capability* as a collection of related functions required to complete a task. These capabilities can be viewed as a combination of the capabilities of the IBM Big Data Platform and the IBM Information Management Reference Architecture.

We collected examples of typical big data environments around the world and identified the most common capabilities in each, discarding the outliers. The objective was to describe 80% of the functionality required in most big data environments. Table 4-1 summarizes these key capabilities.

Table 4-1 Key functional capabilities for big data processing

Capability	**Description**
Data ingestion	Optimize the process of loading data in the data store to support time-sensitive analytic goals.
Search and survey	Secure federated navigation and discovery across *all* enterprise content.
Data transformation	Convert data values from source system and format to destination system and format.
Analytics	Discover and communicate meaningful patterns in data.
Actionable decisions	Make repeatable, real-time decisions about organizational policies and business rules.
Discover and explore	Discover, navigate, and visualize vast amounts of structured and unstructured information across many enterprise systems and data repositories.
Reporting, dashboards, and visualizations	Provide reports, analysis, dashboards, and scorecards to help support the way that people think and work.
Provisioning	Deploy and orchestrate on-premises and off-premises components of a big data ecosystem.
Monitoring and service management	Conduct end-to-end monitoring of services in the data center and the underlying infrastructure.
Security and trust	Detect, prevent, and otherwise address system breaches in the big data ecosystem.

These same capabilities are required in any data warehouse or information management environment.

Figure 4-2 illustrates how the functional capabilities described in Table 4-1 on page 28 are organized within a big data architecture.

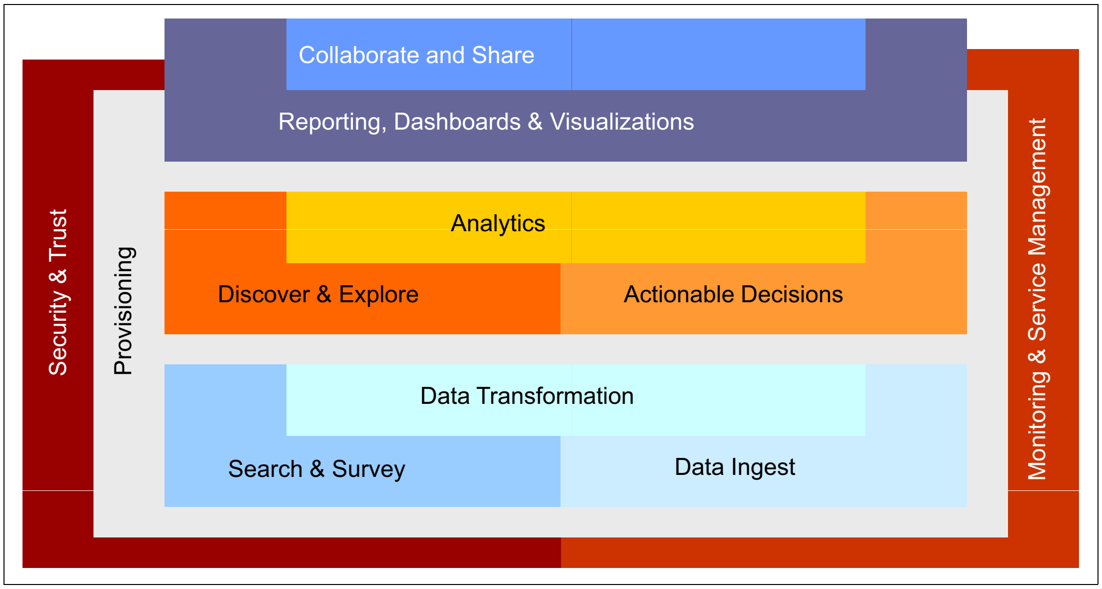


Figure 4-2 Big data functional overview

Figure 4-3 maps available IBM software products to each of the functional areas listed in Table 4-1 on page 28 and illustrated in Figure 4-2.

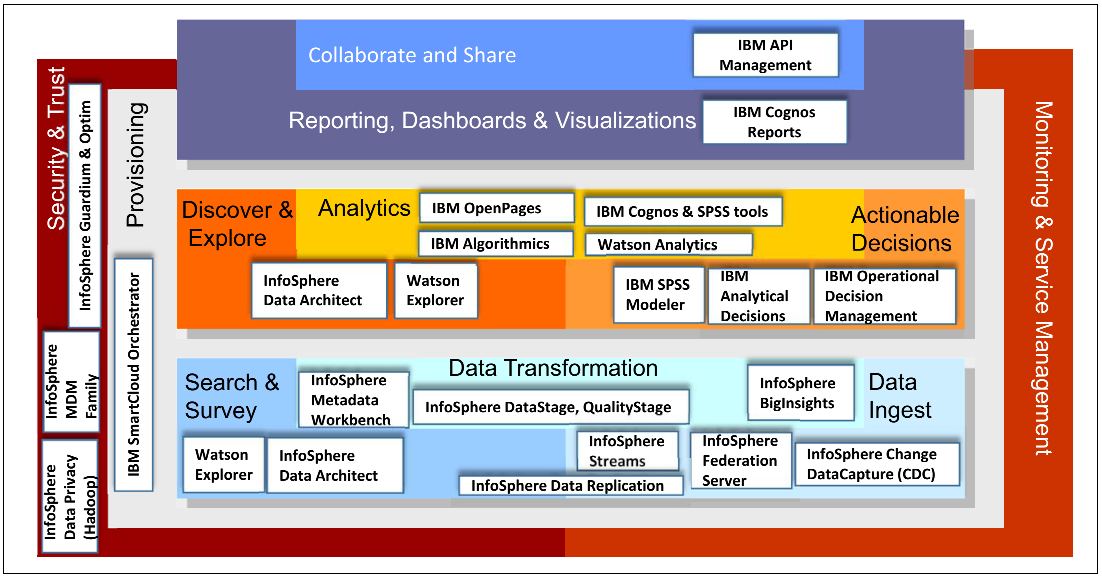

Collaborate and Share
IBM API Management
Reporting, Dashboards & Visualizations
IBM Cognos Reports
Discover & Explore
Analytics
IBM OpenPages
IBM Algorithmics
IBM Cognos & SPSS tools
Watson Analytics
Actionable Decisions
InfoSphere Data Architect
Watson Explorer
IBM SPSS Modeler
IBM Analytical Decisions
IBM Operational Decision Management
Search & Survey
InfoSphere Metadata Workbench
Data Transformation
InfoSphere DataStage, QualityStage
InfoSphere BigInsights
Data Ingest
Watson Explorer
InfoSphere Data Architect
InfoSphere Streams
InfoSphere Federation Server
InfoSphere Change DataCapture (CDC)
InfoSphere Data Replication
Security & Trust
InfoSphere Guardium & Optim
InfoSphere MDM Family
InfoSphere Data Privacy (Hadoop)
Provisioning
IBM SmartCloud Orchestrator
Monitoring & Service Management

Figure 4-3 Big data functional overview with product mapping

Table 4-1 on page 28 provided a brief synopsis for most of the IBM products mentioned in Figure 4-3 on page 29. Table 4-2 describes the rest of the products shown.

Table 4-2 IBM Products for big data processing

IBM product	**Purpose**
IBM Cognos® http://ibm.co/1aIz3Ea	IBM Cognos Business Intelligence and Cognos Financial Performance Management are business intelligence tools used for generating data, reports, plans, and scorecards to support business decisions.
IBM SPSS® http://ibm.co/1aBdScr	SPSS is predictive analytics software used to determine what will happen next, so that you can make smarter decisions, solve problems, and improve outcomes.
IBM InfoSphere Change Data Capture http://ibm.co/1nYyQFF	IBM InfoSphere Change Data Capture helps you replicate your heterogeneous data in near real time to support data migrations, application consolidation, data synchronization, dynamic warehousing, master data management (MDM), business analytics, and data quality processes.
IBM Watson™ Content Analytics http://ibmurl.hursley.ibm.com/MKZI	IBM Content Analytics software is used to discover, organize, and analyze enterprise content to derive insightful and actionable information.
IBM OpenPages® http://ibm.co/1iza0wY	IBM OpenPages helps you manage risk and compliance initiatives across the enterprise.
IBM Algorithmics® http://ibm.co/1e7UP4s	IBM Algorithmics software helps you make risk-aware business decisions.
IBM API Management http://ibm.co/NbAIxg	IBM API Management software provides tools for creating, proxying, assembling, securing, scaling, and socializing web APIs on premises.

4.3 Big data and analytics infrastructure architecture

Figure 4-4 depicts the infrastructure architecture for a big data and analytics solution in a cloud computing environment.

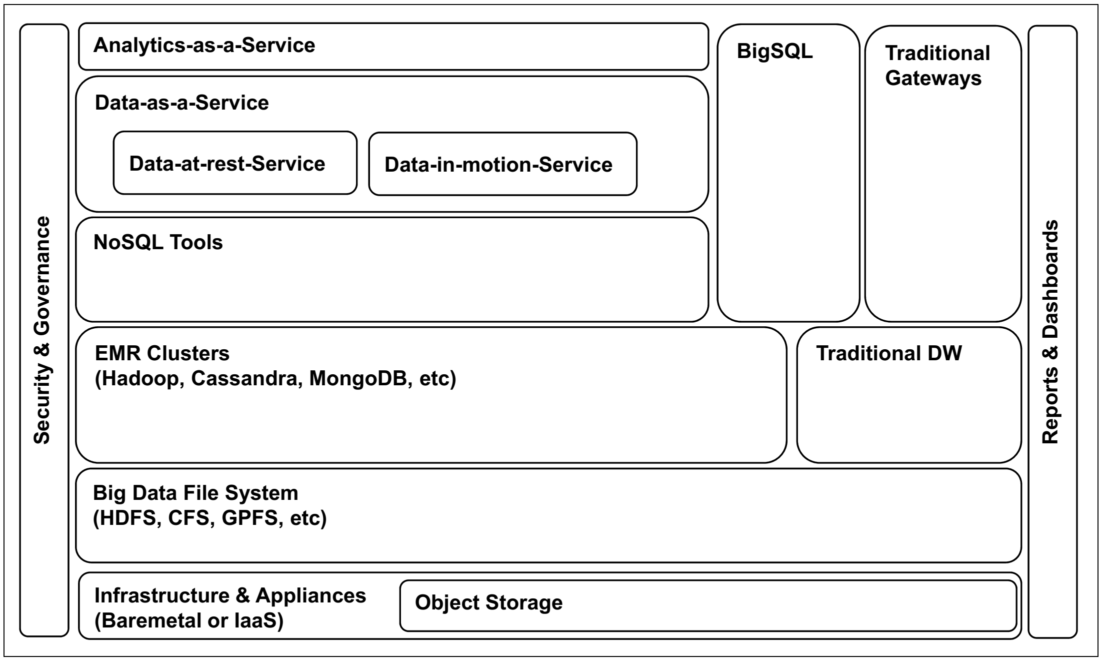


Figure 4-4 Big data and analytics architecture on cloud

This section describes the components in Figure 4-4:

- The infrastructure for running big data workloads can consist of physical hardware or it can be part of an infrastructure as a service (IaaS) cloud. In either case, specialized resource pools are required for specific workloads, for example, Hadoop, online transaction processing (OLTP), or streaming analytics workloads.
- Above the bare-metal server and storage infrastructure is the big data file system, which is also dependent on the workloads. This is where the data resides. In the case of Hadoop, the file system of choice is Hadoop Distributed File System (HDFS) (or General Parallel File System (GPFS)). HDFS runs on top of the resource pool or cluster. In the case of a traditional RDBMS, the operating system's file system (managed by the database) stores the data.
- A MapReduce cluster operates on the data stored in the cluster. In the case of Hadoop, the MapReduce cluster uses the data in an HDFS or GPFS. Amazon offers an Elastic MapReduce (EMR) cloud service that provides MapReduce function but it uses data stores in the Amazon S3 storage. The Amazon S3 storage is more like traditional storage than HDFS distributed storage in that you can have dedicated clusters for different workloads (such as production or development). Because the clusters might be running on top of an IaaS service, you do not need to worry about data loss. Clusters can be expanded or shrunk dynamically. In certain cases, the IaaS layer can be used simply as a data store that provides file systems that are mounted on bare-metal hardware. Traditional databases provide all of the necessary data management functionality for data stored in them.

- Various NoSQL tools provide an abstraction layer on top of MapReduce. Tools, such as Hive, can be used for ad hoc queries. Or, Apache Pig (a programming tool for data operations) can be used for extract, transform, and load (ETL) and for running ad hoc algorithms against a Hadoop cluster. Hive and Pig generate back-end MapReduce jobs that get executed on Hadoop. A new set of tools is now available that provides SQL-based access to EMR clusters, such as IBM BigSQL. BigSQL combines an SQL interface (for ad hoc processing) with parallel processing for handling large quantities of data.
- The Data-as-a-Service layer provides API access to data that is stored in the underlying big data infrastructure. This layer can be broken into two subsets:
  - Data-at-rest-Service: Provides a view into the data stored in underlying layers by using predefined transformations of the data set by data scientists. The view into the data can typically be used by others in IT.
  - Data-in-motion-Service: Provides a continuous stream of transformed, and possibly processed, data for consumption downstream.
- The Analytics-as-a-Service layer receives both data-at-rest and data-in-motion, applies analytical algorithms that are specific to any data-consuming applications that are involved, and provides dashboards, reports, visualizations, and predictive modeling related to the data. This layer hides all of the complexity of data collection, storage, and cleansing and provides an easy mechanism for data scientists to explore large volumes of data to extract insights.

Figure 4-5 takes the concepts shown in Figure 4-4 on page 31 and overlays them with IBM products mapped to each capability.

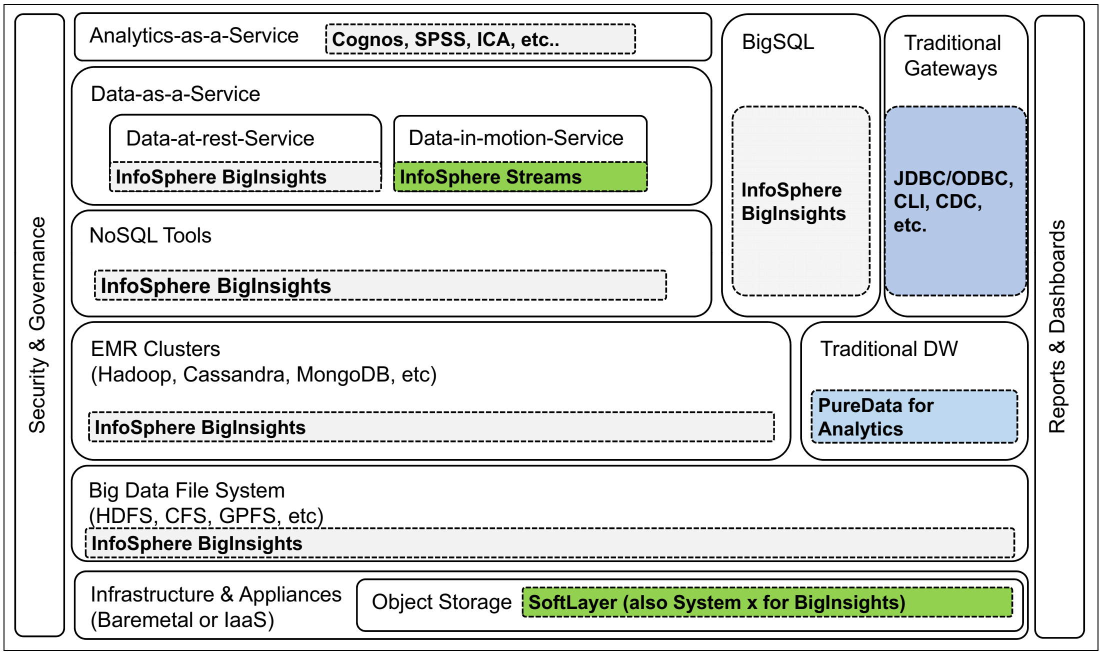


Figure 4-5 Big data analytics infrastructure with IBM products

5

Key big data and analytics use cases

IBM has identified the following high-value use cases based on numerous customer engagements across a wide variety of industries. These use cases can be your first step into big data:

- Big data exploration

  Find, visualize, and understand big data to improve decision making. Big data exploration addresses the challenge faced by every large organization: Information is stored in many different systems and silos and people need access to that data to do their day-to-day work and make important decisions.

- Enhanced 360-degree view of the customer

  Extend existing customer views by incorporating additional internal and external information sources. Gain a fuller understanding of customers, including what things are important to them, why they buy certain products, how they prefer to shop, why they switch products or stores, what they plan to buy next, and what factors lead them to recommend a company to others.

- Operations analysis

  Analyze various machine and operational data for improved business results. The abundance and growth of machine data, which can include anything from IT components to sensors and meters to GPS devices, requires complex analysis and correlation across different types of data sets. By using big data technologies for operations analysis, organizations can gain real-time visibility into operations and the customer experience.

- Data warehouse augmentation

  Integrate big data and data warehouse capabilities to increase operational efficiency. Optimize your data warehouse to enable new types of analysis. Use big data technologies to set up a staging area or landing zone for your new data before determining what parts of it need to be moved to the data warehouse. Use information integration software and related tools to offload infrequently accessed or aged data from warehouse and application databases.

- Security intelligence (security, fraud, and risk analysis)

  Reduce risk, detect fraud, and monitor cyber security in real time. Augment and enhance cyber security platforms and related analysis systems with big data technologies that process and analyze new types and sources of underutilized data (for example, social media posts, emails, remote sensor readings, and Telco signals). The results can be improved insight and understanding in applications, such as intelligence gathering, public safety, and law enforcement.

5.1 Use case details

This section describes each use case in greater detail. In each example, the main functions are separated into layers, as shown in Figure 5-1 and the ensuing figures. In addition, the following high-level layers are useful for conceptualizing how things are organized:

- The lower layers of the use case transform data into common processing formats.
- The middle layers of the use case move data to appropriate storage containers to optimize the aggregation and grouping of data subsets, while making it possible to perform iterative operations on the data millions or even billions of times, if necessary.
- The higher layers provide the applications that ultimately implement the use case solutions.

5.1.1 Big data and analytics exploration

In the big data and analytics exploration scenario, users want to discover more information about specific topics, using as much relevant information as possible. The following steps in this scenario are important:

- Exploring new data sources
- Mining existing data for relevant associations
- Creating new business value from unstructured content
- Understanding large data patterns with visualization and algorithms

Figure 5-1 shows a sample configuration for big data exploration.

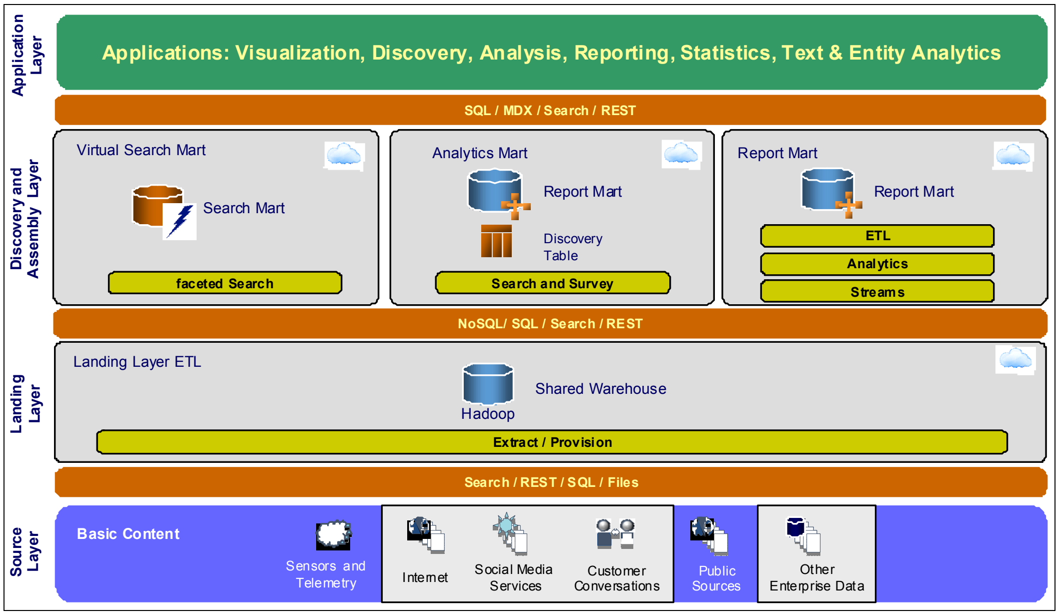


Figure 5-1 Use case pattern: Big data exploration

Data is collected in the Source Layer. Sources include Internet and social media websites, operational data such as support call logs containing customer conversations, and unmodeled data from other parts of the enterprise. Search, Representational State Transfer (REST), SQL, and other file sharing protocols are used to extract copies of the data into shared warehouses that act as staging or *landing* areas for further information discovery.

Applications in the Application Layer interact with shared warehouses in the Discovery and Assembly Layer, for example:

- Search: Virtual Search Marts offering dimensionally aligned search results using facets or similar taxonomy-aware grouping mechanisms. This type of organization allows applications to show new information across familiar reporting dimensions without modeling.
- Analytics Marts: Discovery Tables created using tools, such as InfoSphere BigSheets. An advanced analyst experiments with different Landing Layer sources looking for relevant data that is not already exposed in existing data marts. Analytics and statistical functions are applied to create data correlations that are exported to a report mart for use by higher-level applications.
- Dedicated warehouses: Traditional report marts are constructed as always. A combination of streaming, analytics, and extract, transform, and load (ETL) allows more data to be extracted faster than before. The result is timely, high-quality reporting data with greater domain coverage.

Exploration applications are then free to iterate over a complete range of sources to provide unique solutions that combine existing and new data.

Big data exploration: Case study example

A global aerospace manufacturer increased efficiency among its knowledge workers and saved $36 million annually by deploying a big data exploration solution:

- Requirements:
  - Reduce costly maintenance delays and avoid possible financial penalties for out-of-service equipment
  - Increase the efficiency of maintenance and support technicians, support staff, and engineers
- Results:
  - Supported 5,000 service representatives
  - Eliminated the use of paper manuals that were previously used for research
  - Placed more than 40 additional airplanes into service without adding more support staff
  - Reduced call time by 70% (from 50 minutes to 15 minutes)

5.1.2 Enhanced 360-degree view of the customer

This use case addresses the need to optimize every customer interaction by gathering comprehensive sets of data about those customers. The following functions are typical:

- Building a connected picture of the typical customer
- Mining all existing and new sources of information about the customer
- Analyzing social media to uncover customer sentiments about the company’s products
- Optimizing every customer interaction based on the newly mined information

Figure 5-2 shows a typical deployment.

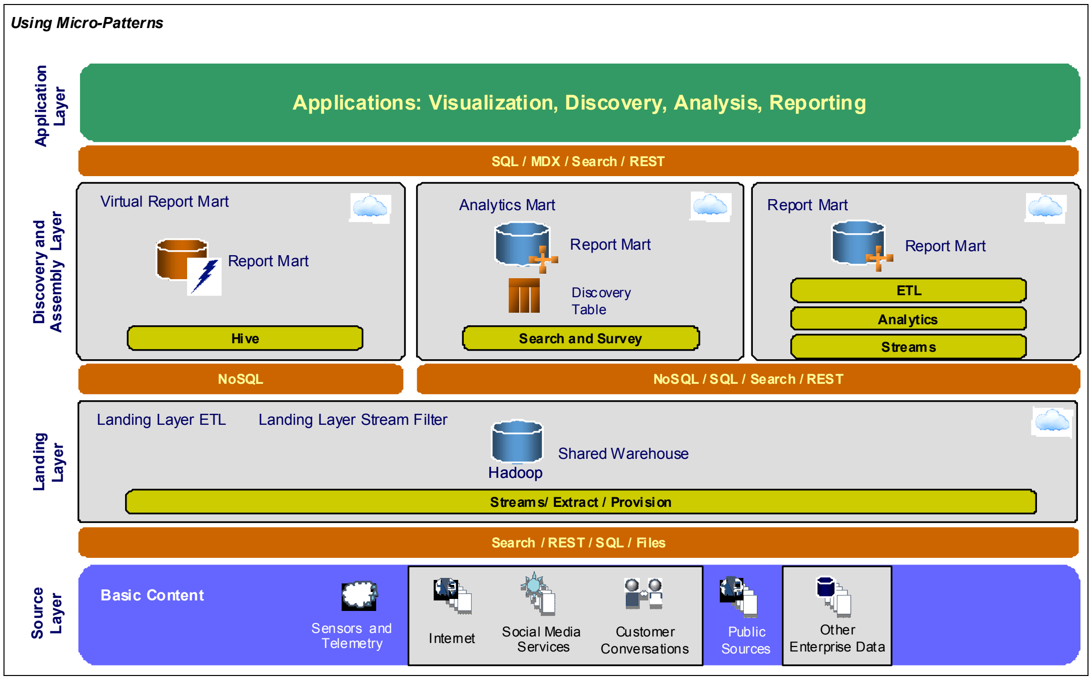


Figure 5-2 Enhanced 360º view of the customer

Data is collected in the Source Layer in the same fashion described in the previous use case. Additional information is collected from sources, such as public and government websites. Streams are added to the extraction process to provide faster access to selected data.

Applications interact with a shared warehouse. Virtual report marts are assembled using the Hive SQL protocol. This allows larger Data Marts to be offered to applications without needing to replicate data. Other sources are enhanced by faster data stream-based data collection.

Customer reporting and analytics use the Discovery and Assembly Layer. Applications have the option of accessing public and social media data in near real time to create timely alerts about customer trends and behavior.

Enhanced 360-degree view of the customer: Case study example

An IBM customer, a product company, improved access across 30 different information repositories containing customer interaction data. Here are the details:

- Requirements:
  - Offer an intuitive user interface for data exploration and discovery across all repositories.
  - Provide access to all offices around the globe, and deploy the solution quickly for a lower total cost of ownership.
  - Provide secure search capabilities across Microsoft SharePoint sites, intranet pages, wikis, blogs, and databases.

- Results:
  - Gained new ability to identify experts across all global offices and 125,000 users worldwide
  - Eliminated duplicate work effort by employees
  - Improved discovery (*find-ability*) across the entire organization
  - Provided a federated, integrated system of knowledge and information that improved internal decision making

5.1.3 Operations analysis

This use case allows users to analyze various machine and operational data sources for improved business results. The abundance and growth of machine data, which can include anything from IT machines to sensors and meters to GPS devices, requires complex analysis and correlation across unfamiliar domains. By using big data for operations analysis, organizations can gain real-time visibility into operations, customer experience, transactions, and behavior.

Analytics applied to machine data helps an organization achieve greater operational efficiency. Here are some typical examples:

- Analyzing machine data to identify events of interest
- Applying predictive models to identify potential anomalies
- Combining information to understand service levels
- Monitoring systems to avoid service degradation or outages

Figure 5-3 shows a typical implementation.

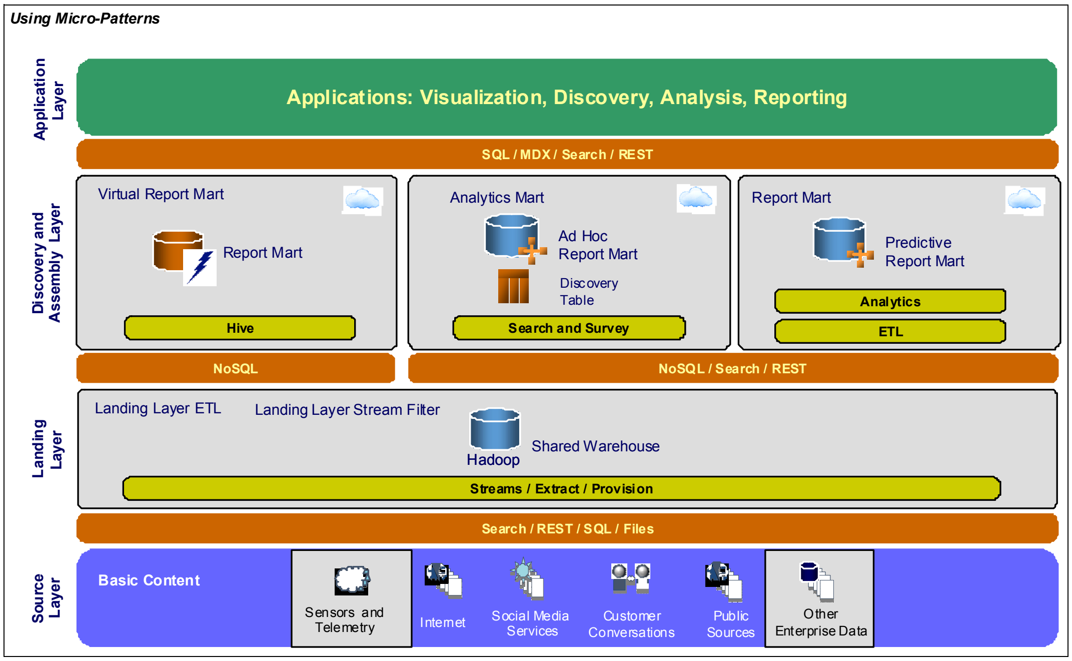


Figure 5-3 Operational analysis use case pattern

Data is collected in the Source Layer. It includes Internet, social media, public, and enterprise sources.

Applications interact with report marts that are built using a combination of streams, ETL, and analytics. The focus of many operations is data augmentation. Existing tables and databases are expanded with new columns and attributes as information is integrated.

Reporting and analytics typically follow existing dimensional patterns. Augmented data offers new associations that are easier to assess and use.

Operational analysis: Case study example

A large phone company reduced churn and improved customer satisfaction, helping to ensure that campaigns are highly effective and timely. Here are the details:

- Requirements:
  - Ensure that marketing campaigns target the right customers, especially if they are considering leaving the network.
  - Keep the company's high-usage customers happy by using campaigns offering services and plans that are right for them.
- Results:
  - Achieved a projected 100% increase in campaign response rates (from 25% - 50%) due to improved predictive analytics
  - Reduced the time needed to analyze Call Detail Records (CDRs) from a full day to as little as 30 seconds
  - Trimmed churn by an expected 15% - 20%

5.1.4 Data warehouse augmentation

This use case uses big data to increase operational efficiency. Landing Zone warehouses facilitate the analysis of data that is in constant demand. Infrequently accessed or aged data is moved away from warehouse and application databases as part of a continuous delivery workflow.

The goal is to help organizations get more value from an existing data warehouse investment while reducing overall costs, for example:

- Adding new sources of data to existing data warehouse investments
- Optimizing storage by providing a queryable archive
- Rationalizing data for greater simplicity and reduced expense
- Speeding data queries to enable more complex analytical applications
- Improving the ability to scale predictive analytics and business intelligence operations

Figure 5-4 shows a sample implementation.

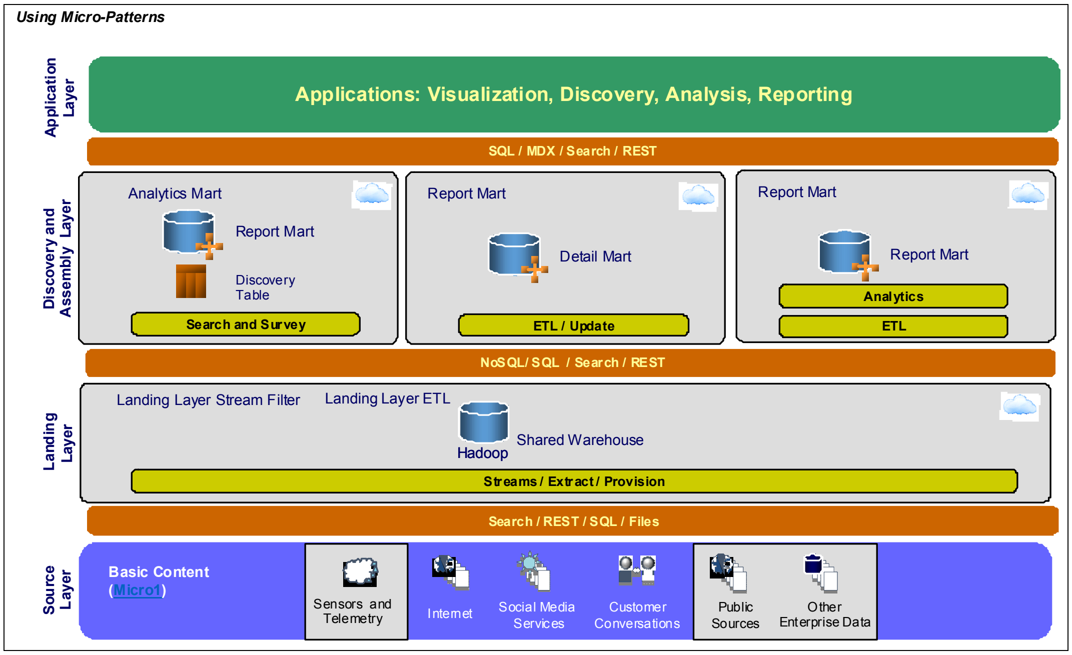


Figure 5-4 Data warehouse augmentation use case pattern

The Source Layer follows a pattern similar to use cases already described. Typical information sources include telemetry and enterprise data.

The focus of many operations continues to combining old data with new, which is a process called *data augmentation*. The ability to explore and find new associations becomes an essential requirement. Virtual Report Marts over Hive are used to take advantage of SQL-based analytics. Discovery tables with faceted search capabilities are also introduced to facilitate ad hoc data collections.

Reporting and analytics follow existing dimensional patterns. Augmented data offers additional information that is generally compatible with existing applications.

Data warehouse augmentation: Case study example

An automobile manufacturer needed to build a global data warehouse to consolidate its global data-related projects; deliver real-time operational reporting; create a single infrastructure to handle structured, semi-structured, and unstructured data; and deploy enterprise-class capabilities to gain big data insights. Here are the details:

- Requirements:
  - Improved call center and dealer service performance
  - More efficient regulatory compliance
  - Faster, more accurate financial performance data
  - Improved new vehicle designs based on root cause analyses of defects
- Results

  Existing data marts showed only small slices of the overall picture with low-quality results in some key areas. By switching to a comprehensive Landing Layer solution, the number of applications using new data sources increased while the amount of ETL and related activities dropped. There was a vast improvement in the range of questions that were answerable, and the total cost of ownership decreased.

5.1.5 Security intelligence (security, fraud, and risk analysis)

This use case shows how organizations can lower risk, detect fraud, and monitor cyber security in real time. Objectives include enhancing cyber security and intelligence analysis platforms to process and analyze new data types, and identifying sources of underutilized data that can improve intelligence, security, and law enforcement insight.

This use case has the following focus areas:

- Intelligence and surveillance insight: Analyze data in motion and data at rest to find associations, uncover patterns and facts, and maintain currency of the information.
- Real-time cyber attack prediction and mitigation: Analyze network traffic to discover new threats sooner, detect known complex threats, and take action in real time.
- Crime prediction and protection: Analyze Telco and social media data to gather criminal evidence, prevent criminal activities, and proactively apprehend criminals.

Figure 5-5 shows a typical deployment.

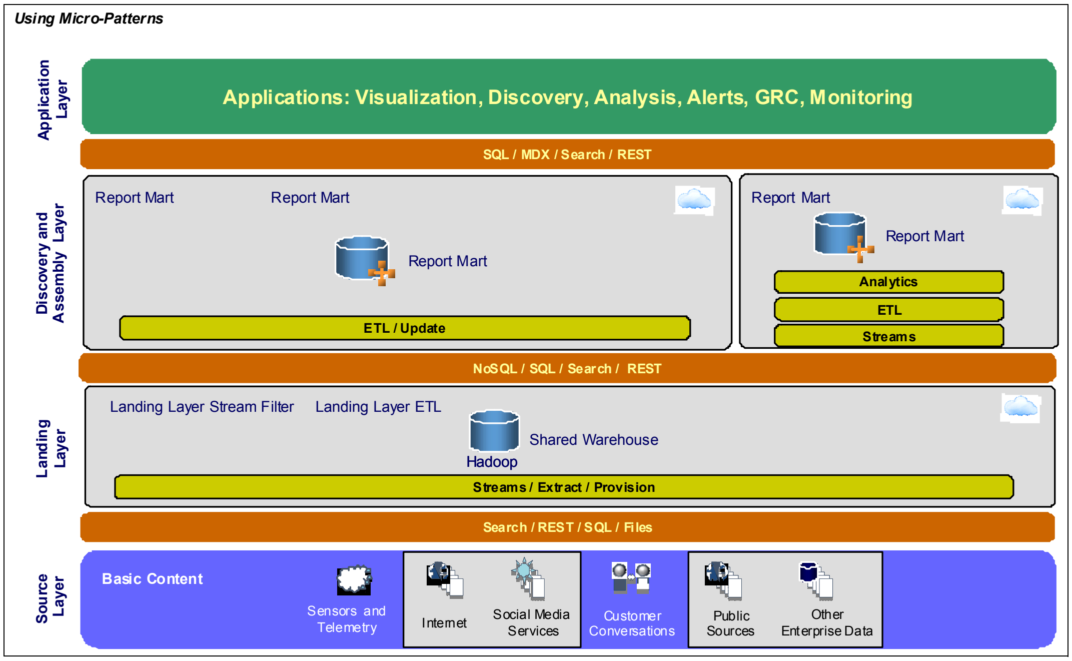


Figure 5-5 Security intelligence (security, fraud, and risk analysis) use case pattern

Source Layer data follows the usual big data ingestion pattern. Real-time sensor and data acquisition is added to facilitate more timely analytics and decision making.

A virtual Search Mart offers dimensionally aligned search results using facets or similar taxonomy-aware grouping mechanisms. This type of organization allows applications to show new information across familiar reporting dimensions without modeling.

Analytics Marts are created from Discovery Tables. Statistical functions are applied to create data correlations that are exported to a report mart for use by higher-level applications.

Traditional report marts are constructed to meet defined business requirements. A combination of streaming, analytics, and ETL allows more data to be extracted in less time. Detail Marts are created to house summary and transaction data.

Reporting and analytics follow traditional dimensional paths. Augmented data is often correlated with statistical models to provide the basis for further exploration and analytics. Results are then typically integrated into a comprehensive governance policy and compliance model.

Security intelligence (security, fraud, and risk analysis): Case study example

A security company uses streaming data technology to support covert intelligence and surveillance sensor systems. Here are the details:

- Requirements

  Deploy a security surveillance system to detect, classify, locate, and track potential threats at a highly sensitive national laboratory.

- Results:

  - Reduced the time required to capture and analyze 275 MB of acoustic data from hours to just 1/14 of a second
  - Enabled analysis of real-time data from different types of sensors and 1,024 individual channels to support extended perimeter security
  - Enabled a faster, more intelligent response to any threat

6

Big data and analytics patterns

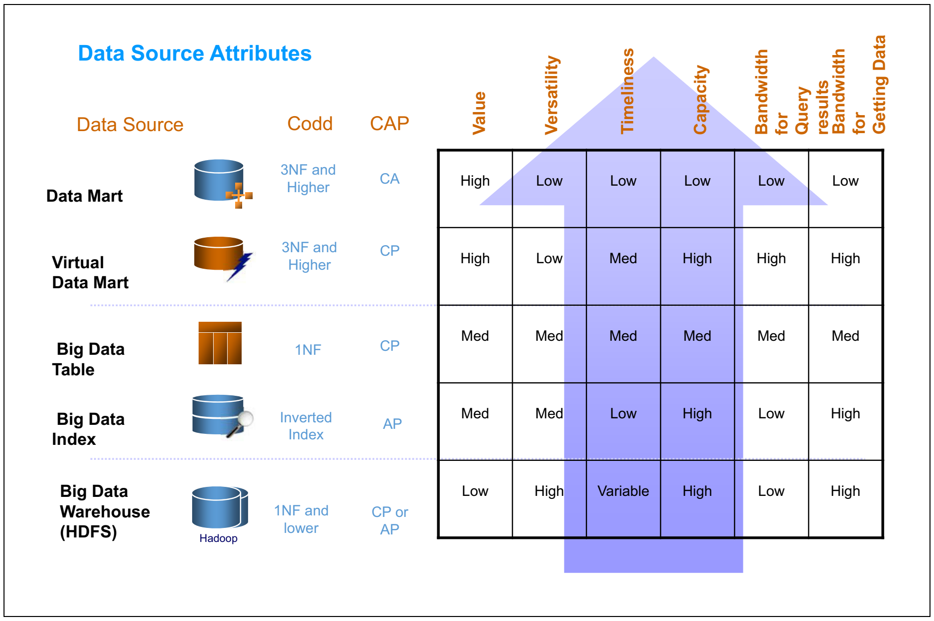
This chapter uses familiar processing components and database configurations to define a set of reusable design patterns. Solutions can be assembled from the top down to address primary user requirements. Bottom-up design is also encouraged, allowing technology choices to be made as early as possible.

6.1 Zones

Big data analysis often calls for a flexible environment, allowing for rapid, high-volume data collection and processing. Traditional extract, transfer, and load (ETL) provides good results when data is understood and correctly segmented. Yet when new and unfamiliar information is added to the mix, ETL infrastructures can have difficulty cleaning and aligning incoming data to predefined dimensions and categories.

Overlapping processing *zones* are offered as a solution. Data is provisioned in multiple formats throughout its lifecycle. This staging of data allows for more flexible query options and easier alignment with new or unexpected types of information. The use of zones differs from ETL, where only the final warehouse or data mart is ultimately saved.

Figure 6-1 shows how data typically moves between components in a Shared Analytics Information Zone.



Data Source Attributes

Data Source	Codd	CAP	Value	Versatility	Timeliness	Capacity	Bandwidth for Query results	Bandwidth for Getting Data
Data Mart	3NF and Higher	CA	High	Low	Low	Low	Low	Low
Virtual Data Mart	3NF and Higher	CP	High	Low	Med	High	High	High
Big Data Table	1NF	CP	Med	Med	Med	Med	Med	Med
Big Data Index	Inverted Index	AP	Med	Med	Low	High	Low	High
Big Data Warehouse (HDFS)	1NF and lower	CP or AP	Low	High	Variable	High	Low	High

Figure 6-1 Big data zones

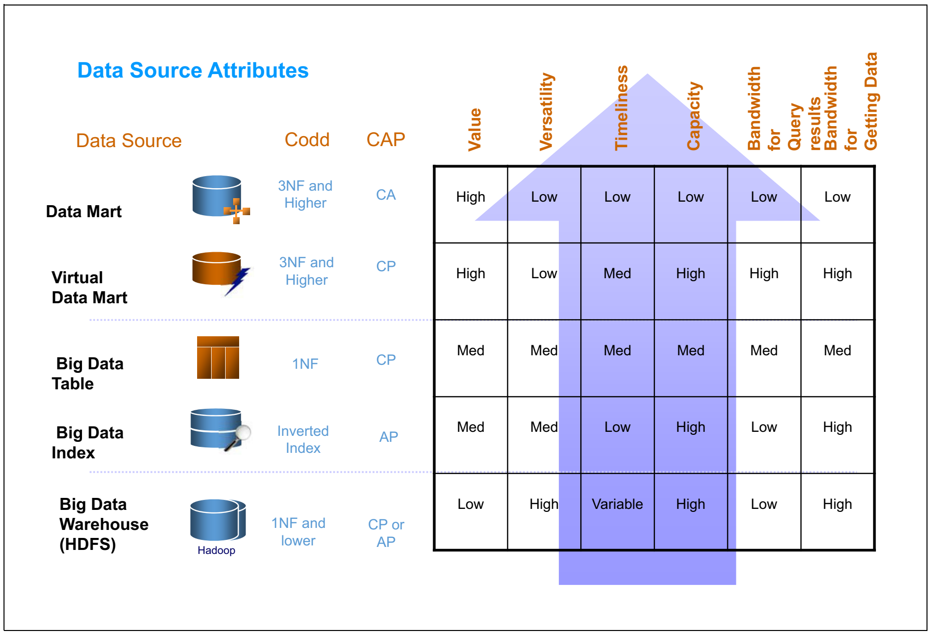
The goal is to accommodate the diverse and distributed information needed for modern data analysis. Some data can be consumed successfully by using an application programming interface (API). But many sources, such as websites or online files, lack the bandwidth and availability needed to support iterative content analysis. Therefore, unprocessed data is typically transferred, either in whole or in part, to a landing area in the Shared Analytics Information Zone. This approach facilitates rapid retrieval, interrogation, and aggregation of elements using Hadoop Distributed File System (HDFS) and related columnar technologies.

Similar to ETL, analytics are used to transform unprocessed data into more useful collections of information that better correlate with the data that is already available. The number of steps required to transform this data can vary. Elapsed time and processing cycles, both of which relate to cost, are often balanced against result quality. Although further iterations can improve an answer, the amount of improvement might be small enough to warrant using fewer cycles.

Results are collected, interpreted, and presented in the Information Interaction Zone. Applications take data from the Shared Analytics Information Zone and add value by further identifying trends, correlations, and connections between data elements.

6.1.1 Data refinement value

Data is transformed, augmented, and normalized as it moves through different analytic phases. The value of this data tends to increase as it is cleaned and further normalized in each processing step. Figure 6-2 shows common structures and schemas.



Data Source	Codd	CAP	Value	Versatility	Timeliness	Capacity	Bandwidth for Query results	Bandwidth for Getting Data
Data Mart	3NF and Higher	CA	High	Low	Low	Low	Low	Low
Virtual Data Mart	3NF and Higher	CP	High	Low	Med	High	High	High
Big Data Table	1NF	CP	Med	Med	Med	Med	Med	Med
Big Data Index	Inverted Index	AP	Med	Med	Low	High	Low	High
Big Data Warehouse (HDFS)	1NF and lower	CP or AP	Low	High	Variable	High	Low	High

Figure 6-2 Big data schemas (data structures)

Enterprise data value generally increases as it becomes more closely aligned with the metadata used in reporting and analytics.

Additional attributes come into play here, for example:

- High volume data is commonly stored in a big data warehouse (HDFS) for fast access during the multiple iterations involved in even trivial analyses.
- Indexed data provides fast access to large amounts of data, particularly with structured data and otherwise unfamiliar data.
- Big data tables allow users to organize ad hoc data and ultimately build dimensional views of data sets.
- Virtual data marts provide access to in-place HDFS data while offering access methods that are familiar to the business.
- Data marts provide data snapshots in easily consumable formats for efficient reporting and analysis.

6.1.2 Normalization

Data normalization is familiar to database administrators who are comfortable with the process of organizing the fields and tables of a relational database to minimize redundancy and dependency. First Normal Form (1NF) describes data that is generally stored in rows and columns and meets minimum requirements for discriminating data by key. Third Normal Form (3NF) meets the normalization requirements for easier data access and repurposing of data into online analytical processing (OLAP) cubes and other common representations. Achieving 1NF and 3NF normalization is a goal in most big data projects, even though it is rarely stated. Higher levels of normalization generally result in higher value information that is easier to aggregate and combine with existing reporting data.

6.1.3 Maximizing the number of correlations

Organizations *leave information on the table* when correlations are missed and connections are not as flexible as they can be. Using more data from a wider variety of sources becomes a key element in successful deployments.

Here are four strategies that help:

- Seek to consolidate the best information from operational and analytical systems. Expect to see a mix of both structured and unstructured data.
- Connect to new sources of information, including websites, social media, and other parts of the organization. Correlations and associations will use new tools that extend beyond existing SQL table structures.
- Reuse and collaboration need to be a priority. Sharing results empowers wider enterprise understanding.
- Data security needs to continue to be a focus, not only for source data but for analytic results and associated conclusions, too.

6.1.4 Quality, relevance, and flexibility

New types of data introduce different measures of *uncertainty* into each of the areas depicted in Figure 6-3 on page 47 and listed here:

- *Quality* is a predominant attribute in mature reporting systems. Data is cleaned and normalized to provide the highest-quality data aligned with the reporting requirements.
- *Relevance* is an attribute common in Business Intelligence applications. Data is accurately aligned with the dimensions that correspond to primary business drivers.
- *Flexibility* is often associated with search systems. Results are based on keyword queries that answer an extremely wide range of questions.

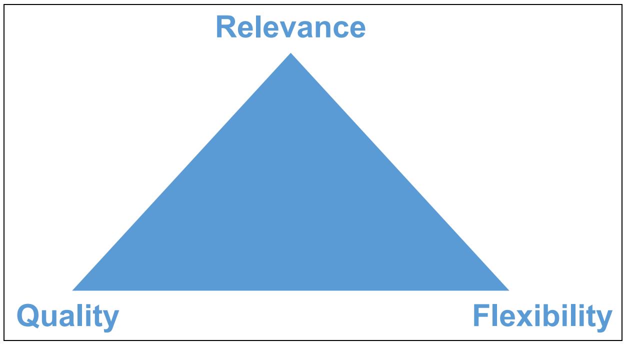


Figure 6-3 Quality, relevance, and flexibility

The challenge with large data sets is that the attributes of quality, relevance, and flexibility are often in opposition to each other. A successful solution needs to adapt. In one part of an application, quality might need to be increased at the expense of relevance or flexibility. Different parts of an application might need to favor flexibility over the other attributes.

6.1.5 Solution pattern descriptions

This section introduces simplified patterns that encapsulate data, processing, and analytics. The patterns can be combined and reused as required to build a wide range of big data solutions.

Solution pattern 1: Landing Zone Warehouse

This pattern illustrates a simple method for delivering moderate amounts of data to applications. Information is extracted and transformed first to a reusable warehouse and then to a solution-aware report mart, as shown in Figure 6-4.

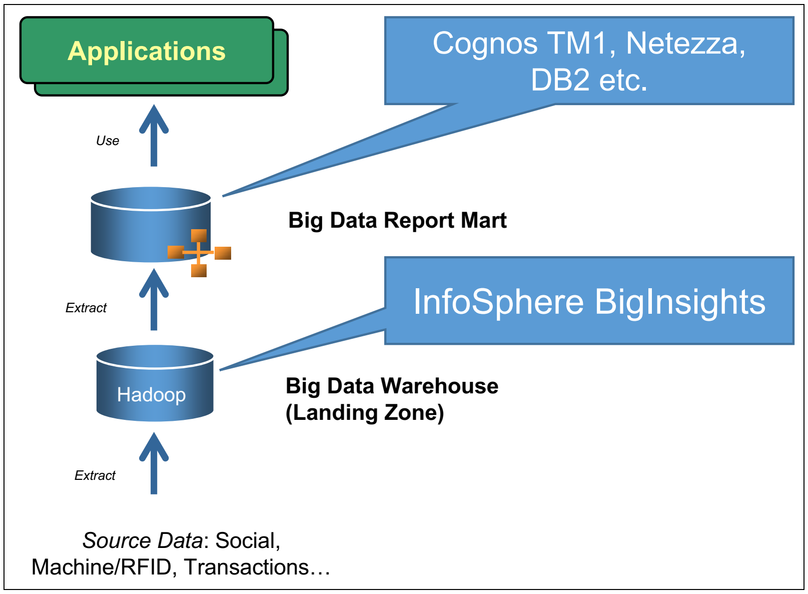

Applications
Cognos TM1, Netezza, DB2 etc.
Use
Big Data Report Mart
InfoSphere BigInsights
Extract
Hadoop
Big Data Warehouse (Landing Zone)
Extract
Source Data: Social, Machine/RFID, Transactions…

Figure 6-4 Landing Zone Warehouse pattern

Consider these summary points about this solution pattern:

- Reporting and analysis in finance, governance, risk, and compliance (GRC) solutions, or similar domains
- Data extracted into physical warehouse from Landing Zone

The following steps are typical and refer to numbers in Figure 6-4 on page 47:

1. Big Data Warehouse HDFS Landing Zone is built using ETL batch processes. A mix of structured, semi-structured, and unstructured data is extracted from various external sources.
2. The Big Data Report Mart is loaded through batch ETL. The structure matches modeled business analysis (BA) elements. Options include In-Memory, Dynamic Cube, and Traditional.
3. Information interaction using SQL or Multidimensional Expressions (MDX) queries.

Solution pattern 2: Virtual Tables

This pattern illustrates a method for delivering large amounts of data to applications. Information is first extracted to a reusable warehouse. Data is then used *in place* with components that offer remote access to the warehouse without further copying source data, as shown in Figure 6-5.

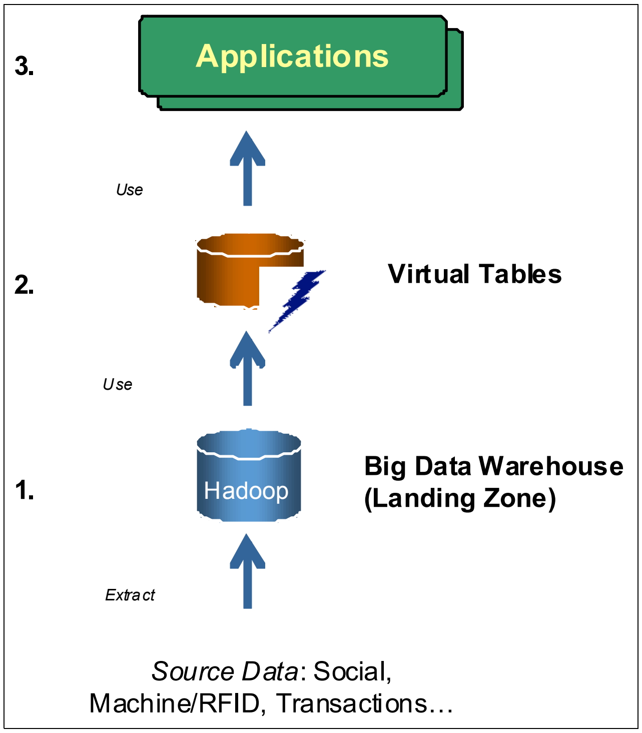


Figure 6-5 Virtual Tables pattern

Consider these summary points about this solution pattern:

- Reporting and analysis in finance, GRC, or similar domains.
- Remote API allows data to stay in the Landing Zone.

The following steps are typical and refer to numbers in Figure 6-5 on page 48:

1. The Big Data Warehouse HDFS (also known as the *Landing Zone*) is built using ETL batch processes. A mix of structured, semi-structured, and unstructured data is extracted from various external sources.
2. Virtual database tables are built in an SQL/Hive style over the Big Data Warehouse HDFS or direct NoSQL queries. The data stays in the HDFS.
3. Information interaction using SQL/MDX queries.

Solution pattern 3: Discovery Tables

This pattern illustrates a discovery and exploration method for delivering flexible data to applications. Information is extracted first to a reusable warehouse and is then extracted and transformed into logical tables. Data is then extracted into a traditional warehouse for use in applications, as shown in Figure 6-6.

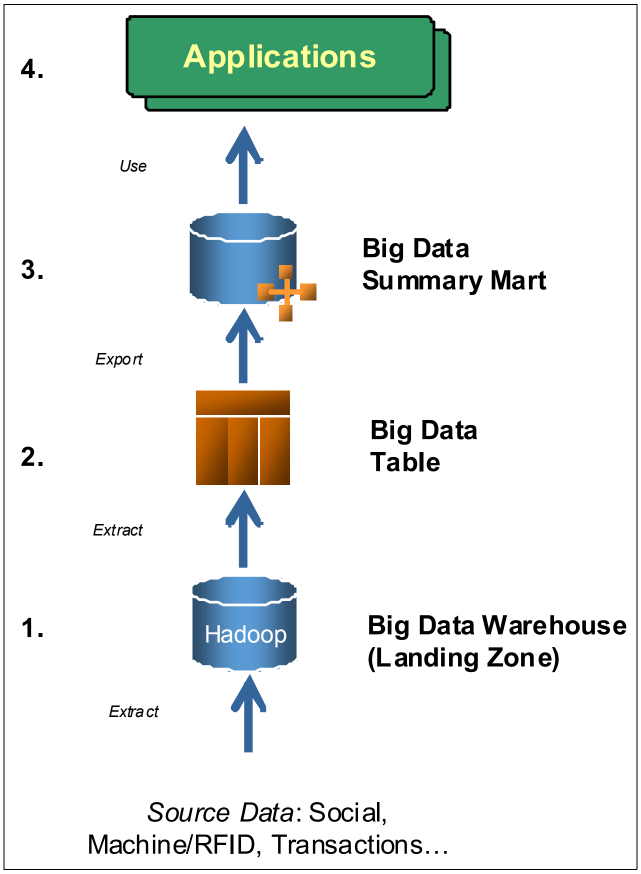


Figure 6-6 Discovery Tables pattern

Consider these summary points about this solution pattern:

- Reporting and analysis in finance, GRC, or similar domains
- Ad hoc exploration and discovery performed using the Big Data Table exploration tool

The following steps are typical and refer to numbers in Figure 6-6:

1. The Big Data Warehouse HDFS Landing Zone is built using ETL batch processes. A mix of structured, semi-structured, and unstructured data is extracted from various external sources.
2. The Big Data Table is built from available sources.

3. The Big Data Summary Mart database is built from Big Data Table content.
4. Information interaction using SQL/MDX or similar queries.

Solution pattern 4: Streams Dynamic Warehouse

This pattern illustrates a processing method for delivering high-volume data to applications. Information is streamed first to a reusable warehouse, with selected information sent to an in-memory warehouse. Data from either of the warehouses is then available for use in applications, as shown in Figure 6-7.

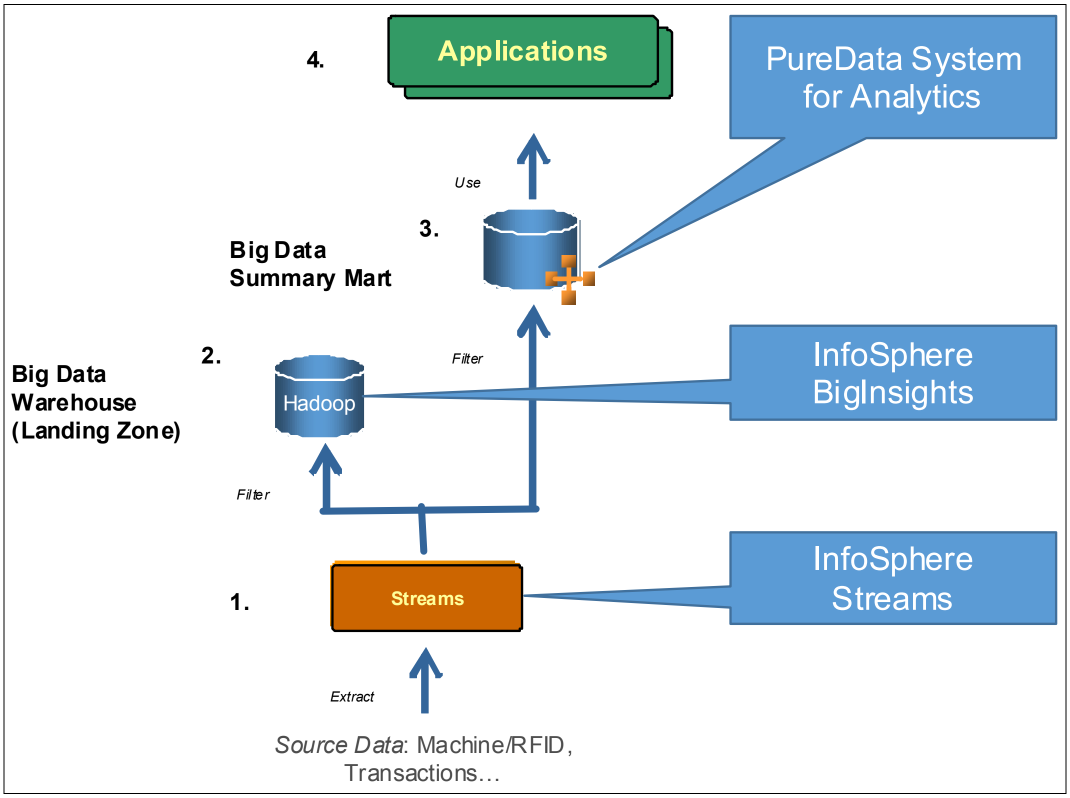


Figure 6-7 Streams Dynamic Warehouse pattern

Consider these summary points about this solution pattern:

- Reporting in finance, GRC, or similar domains.
- Incoming data is filtered by the Streams processor, with a portion of the data sent to the Summary Data Mart.

The following steps are typical and refer to numbers in Figure 6-7:

1. Streams-ingested data is primarily for real-time monitoring applications and includes a mix of raw structured and semi-structured data.
2. The Big Data Warehouse stores selected elements for later analysis.
3. The Big Data Summary Mart is loaded with real-time summary data. Data Mart write-back is used to update content along known dimensions.
4. Information interaction using SQL/MDX queries.

Solution pattern 5: Streams Detail with Update

This pattern illustrates a processing method for delivering high-volume data to applications. Information is streamed first to a reusable warehouse, with selected information sent to an in-memory warehouse. Data is then extracted from the reusable warehouse to a traditional warehouse. Data from either of the top-level warehouses is then available for use in applications, as shown in Figure 6-8.

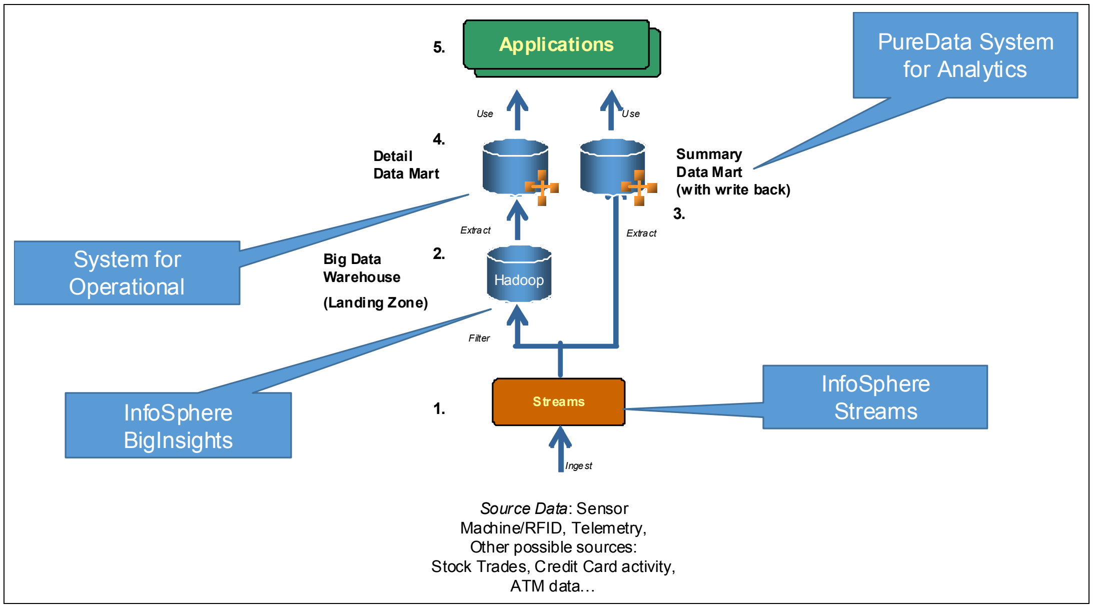


Figure 6-8 Streams Detail with Update pattern

Consider these summary points about this solution pattern:

- Incoming data is filtered by Streams. A portion of the data is sent to the Big Data Warehouse and Detail Data Mart. High-level (transaction-level) data is written to the Summary Data Mart for near real-time analysis.
- All data marts are made available for interactive reporting.

The following steps are typical and refer to numbers in Figure 6-8:

1. Streams-ingested data is primarily for real-time monitoring applications and includes a mix of raw structured and semi-structured data.
2. The Big Data Warehouse is loaded with details that are filtered as required by the Streams process.
3. The Summary Data Mart is loaded with real-time summary data. Data Mart write-back is used to update content along known dimensions.
4. The Detail Data Mart is loaded using batch ETL processes.
5. Information Interaction using SQL/MDX queries.

Solution pattern 6: Direct Augmentation

This pattern illustrates a method for delivering large amounts of high-velocity data to applications. Information is first extracted to a reusable warehouse and augmented in place. Data is then extracted to a traditional warehouse for use in applications, as shown in Figure 6-9 on page 52.

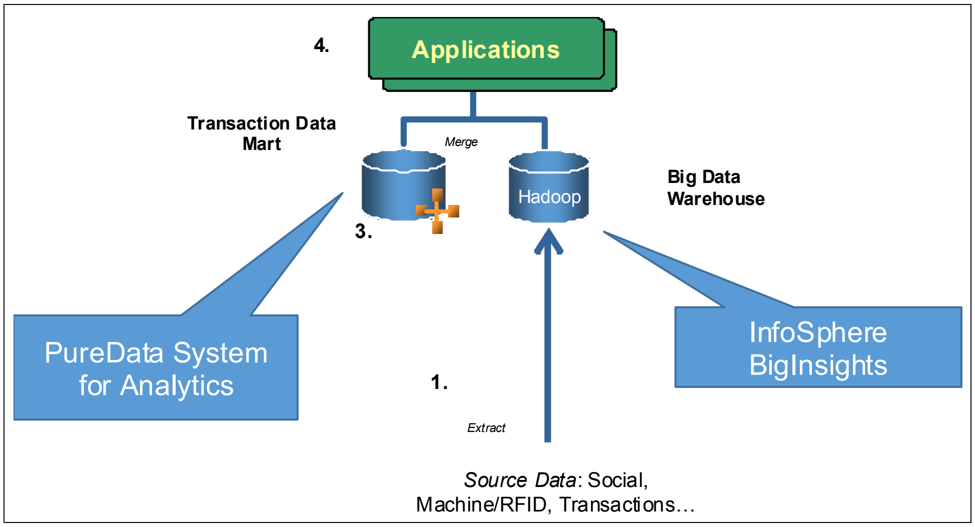


Figure 6-9 Direct Augmentation pattern

Consider these summary points about this solution pattern:

- Reporting is performed against transactions in the warehouse cube and related transcripts in Hadoop.
- Data is merged to show a single summary.

The following steps are typical and refer to numbers in Figure 6-9:

1. The Big Data Warehouse HDFS Landing Zone is built using ETL and involves mostly unstructured operational data.
2. The Virtual Report Mart Virtual database is built using SQL or Hive requests over the Big Data Warehouse HDFS. The data stays in HDFS.
3. The Transaction Data Mart is loaded using batch ETL processes.
4. Information interaction using SQL/MDX queries.

Solution pattern 7: Warehouse Augmentation

This pattern illustrates a method for delivering moderate amounts of high-velocity data to applications. Information is first extracted to a reusable warehouse and augmented in place. Data is then extracted to a traditional warehouse for use in applications, as shown in Figure 6-10 on page 53.

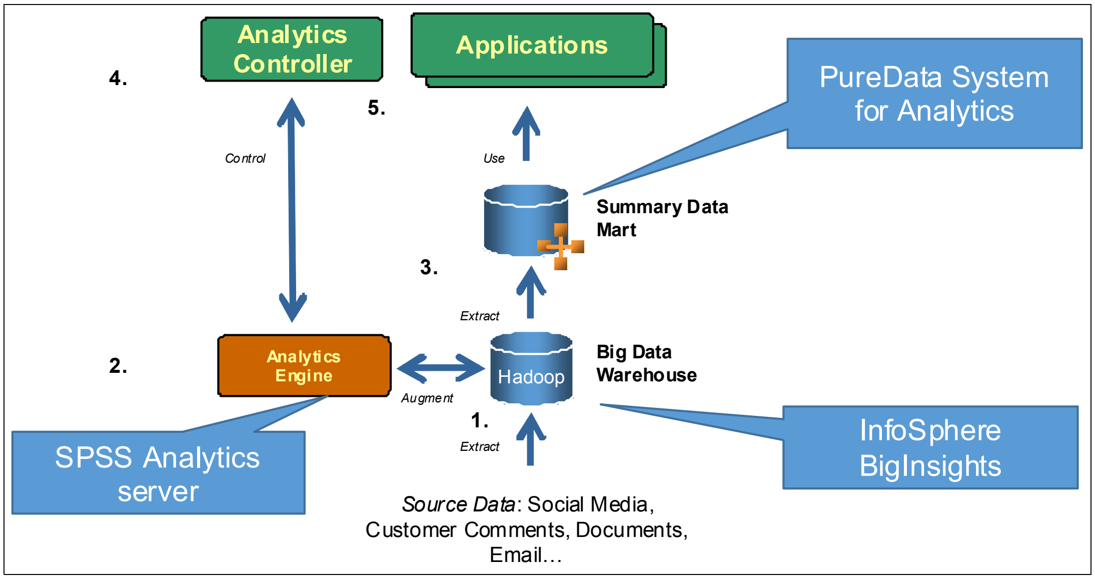


Figure 6-10 Warehouse Augmentation pattern

Consider these summary points about this solution pattern:

- The Big Data Warehouse is augmented by the analytics engine.
- The Summary Data Mart is made available for information interaction.

The following steps are typical and refer to numbers in Figure 6-10:

1. The Big Data Warehouse HDFS Landing Zone is built using ETL.
2. An analytics engine, such as SPSS, augments the Big Data Warehouse.
3. The Summary Data Mart is loaded using batch ETL processes.
4. An analytics application, such as Consumable Analytics, controls the analytics engine.
5. Information interaction using SQL/MDX queries.

Solution pattern 8: Streams Augmentation

This pattern illustrates a method for delivering large amounts of high-velocity data to applications. Information is streamed and augmented in a single step as it is placed in a reusable warehouse. Data is then extracted to a traditional warehouse for use in applications, as shown in Figure 6-11 on page 54.

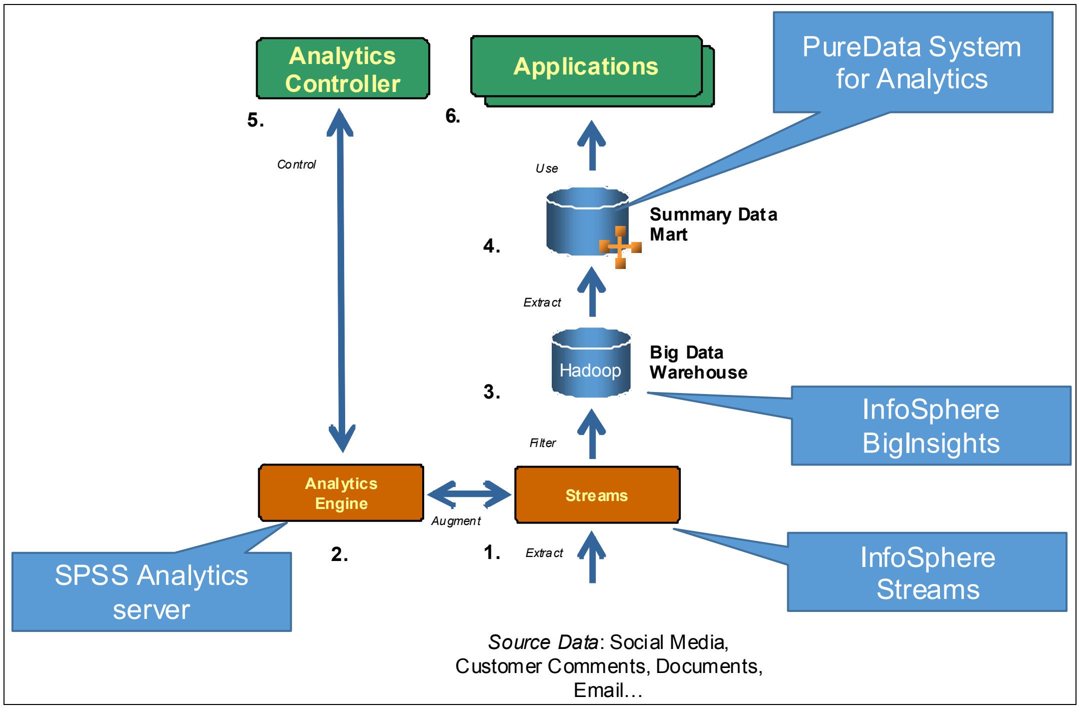


Figure 6-11 Streams Augmentation pattern

Consider these summary points about this solution pattern:

- Streams are augmented in near real time by the analytics engine.
- The Summary Data Mart is made available for information interaction.

The following steps are typical and refer to numbers in Figure 6-11:

1. InfoSphere Streams processes incoming data.
2. An analytic engine, such as SPSS, augments Streams.
3. The Big Data Warehouse HDFS Landing Zone is built from filtered Streams data.
4. The Summary Data Mart is loaded by using batch ETL processes.
5. An analytics application, such as Consumable Analytics, controls the analytics engine.
6. Information interaction using SQL/MDX queries.

Solution pattern 9: Dynamic Search Cube

This pattern illustrates a method for delivering large amounts of indexed data to applications. Information is extracted to a reusable warehouse that is crawled to build an index. Data is then used in place with components that offer remote access to the index and the reusable warehouse, as shown in Figure 6-12 on page 55.

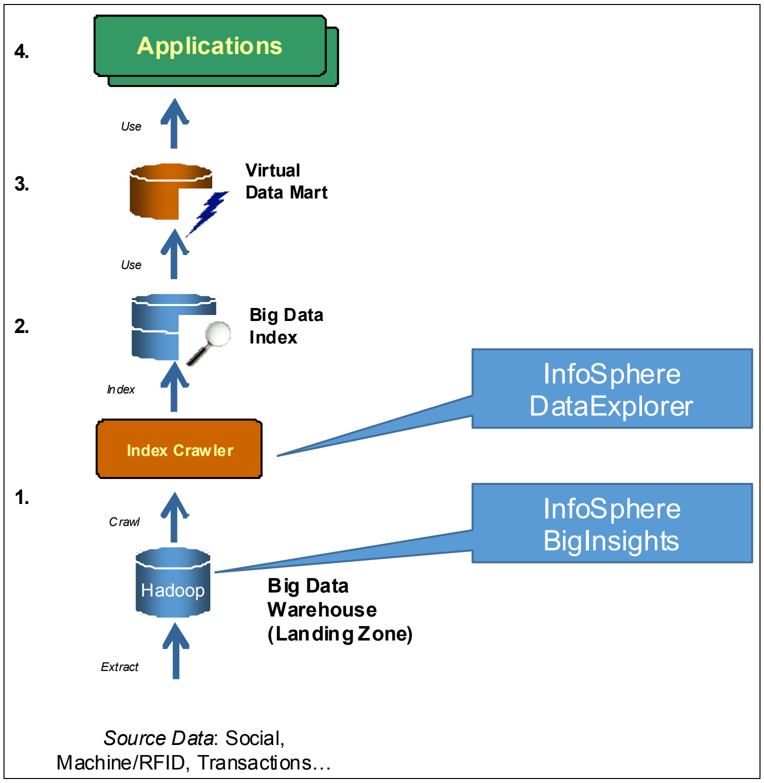


Figure 6-12 Dynamic Search Cube pattern

Consider these summary points about this solution pattern:

- Text sources are collected in the Big Data Index.
- The Virtual Data Mart provides an OLAP view of text facets.

The following steps are typical and refer to numbers in Figure 6-12:

1. An index crawler scans the content.
2. The Big Data Warehouse Index is built with facets representing definitive filter criteria.
3. The Virtual Data Mart is created using the faceted search index.
4. Interactive reporting using SQL/MDX queries.

6.1.6 Component patterns

Component patterns define components in a granular fashion to promote integration in both homogeneous and heterogeneous environments. These patterns were used to assemble the solution patterns shown in the previous sections.

Component pattern 1: Source Data

This pattern illustrates the typical Big Data sources before extraction (see Figure 6-13).

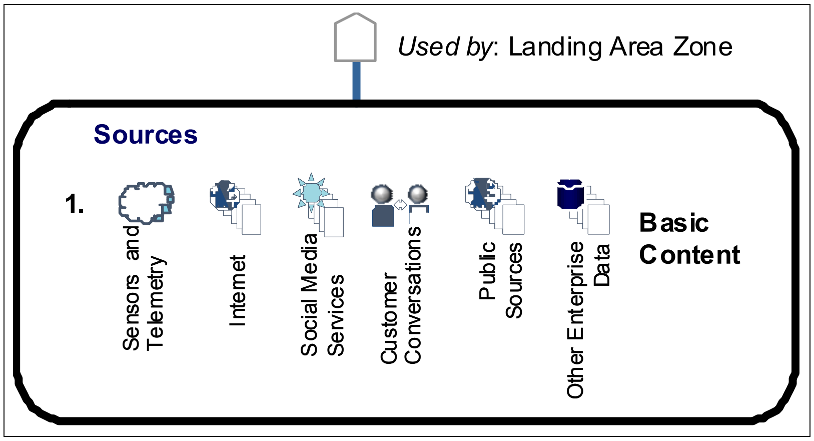


Figure 6-13 Source Data pattern

The basic content comes from a broad range of external suppliers.

The following component detail refers to number 1 in Figure 6-13:

1. Source Data is the full range of large scale external and internal sources of unstructured, semi-structured, and unstructured formats.

   Data Rate, Data Quality, and Response Times are variable, although it is reasonable to expect lower performance than HDFS and other alternatives.

   Unmodeled data is included from other parts of the enterprise that are commonly used between business silos.

Component pattern 2: Source Event

This pattern illustrates typical event processing (see Figure 6-14).

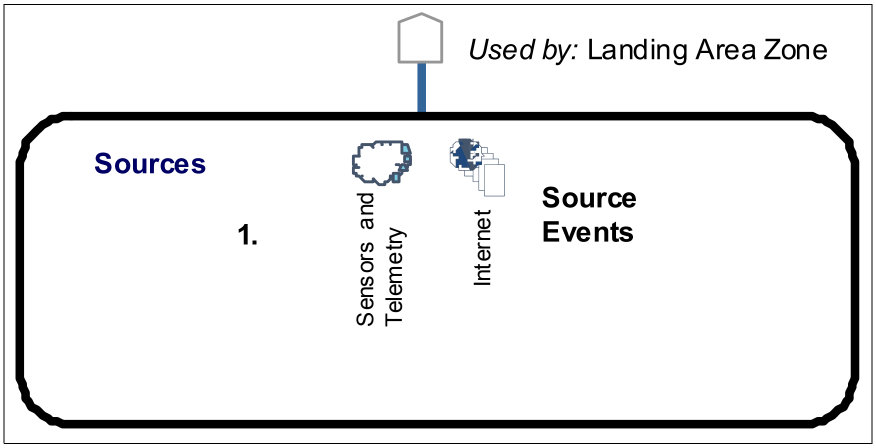


Figure 6-14 Source Event pattern

Source Events are from a subset of Source Variety providers (S1).

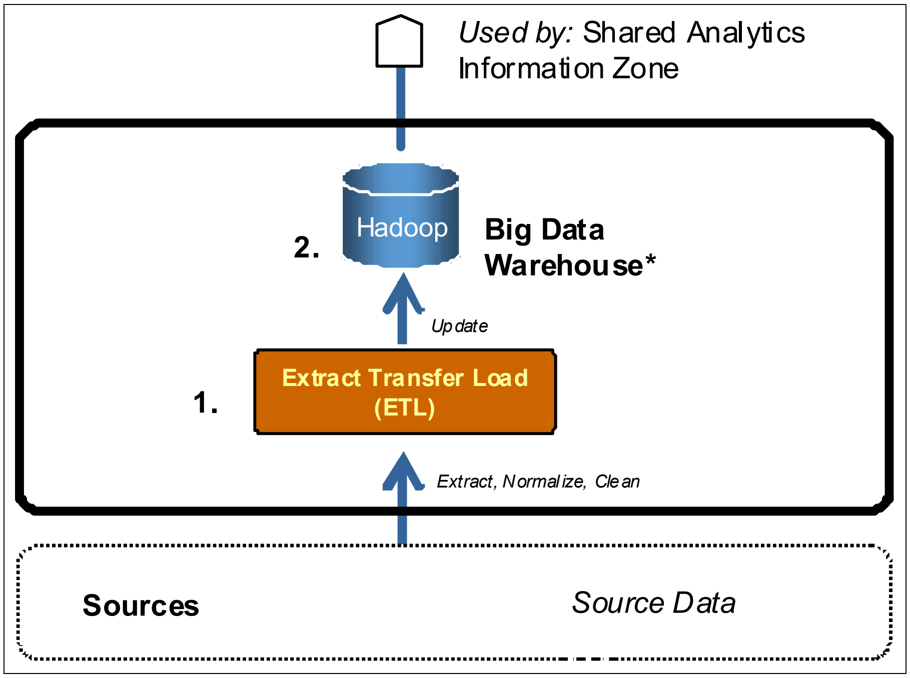
The following component detail refers to numbers in Figure 6-14 on page 56:

1. Source Events are generally high-volume and granular data elements in unstructured, semi-structured, and unstructured formats. They typically are transient in nature and the lifespan or usefulness of the data is time-limited.

   Data Rate, Data Quality, and Response Times are variable, although it is reasonable to expect high volume and low latency.

Component pattern 3: Landing Area Zone ETL

This pattern illustrates a typical extract, transfer, and load (ETL) process (see Figure 6-15).



Used by: Shared Analytics Information Zone
2.
Hadoop
Big Data Warehouse*
Update
1.
Extract Transfer Load (ETL)
Extract, Normalize, Clean
Sources
Source Data

Figure 6-15 Landing Area Zone ETL pattern

The Landing Area Zone containing the Big Data Warehouse is built from various external sources using enterprise-proven ETL methodologies.

The following component details refer to numbers in Figure 6-15:

1. ETL processes use software as a service (SaaS), Search, SQL, and other accepted methods over well-defined Source Data to build the Big Data Warehouse. Update intervals vary depending on requirements.
2. The Big Data Warehouse HDFS provides columnar access to data, typically using NoSQL requests.

Component pattern 4: Landing Area Zone Search and Survey

This pattern illustrates a typical low-level search and survey process (see Figure 6-16).

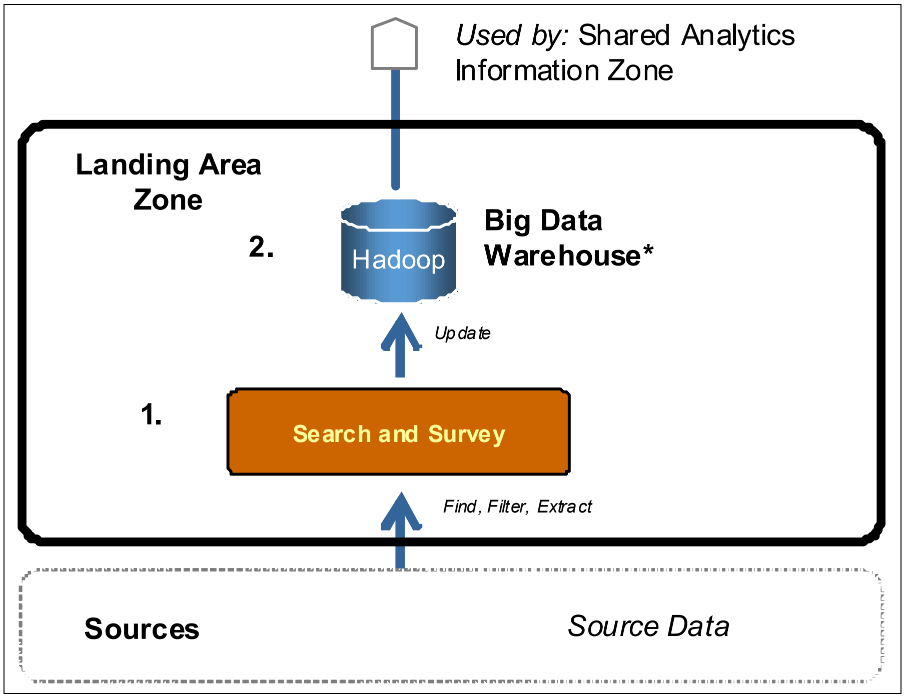


Figure 6-16 Landing Area Zone Search and Survey pattern

The Landing Area Zone containing the Big Data Warehouse is built from various external sources using ad hoc search and survey methodologies currently associated with a Data Scientist.

The following component details refer to numbers in Figure 6-16:

1. Search and Survey finds and extracts data using SaaS, Search, SQL, and other accepted methods over ad hoc and partially understood Source Data to build the Big Data Warehouse. Update intervals vary depending on requirements.
2. The Big Data Warehouse HDFS provides columnar access to data, typically using NoSQL requests.

Component pattern 5: Landing Area Zone Stream Filter

This pattern illustrates a typical streaming process (see Figure 6-17).

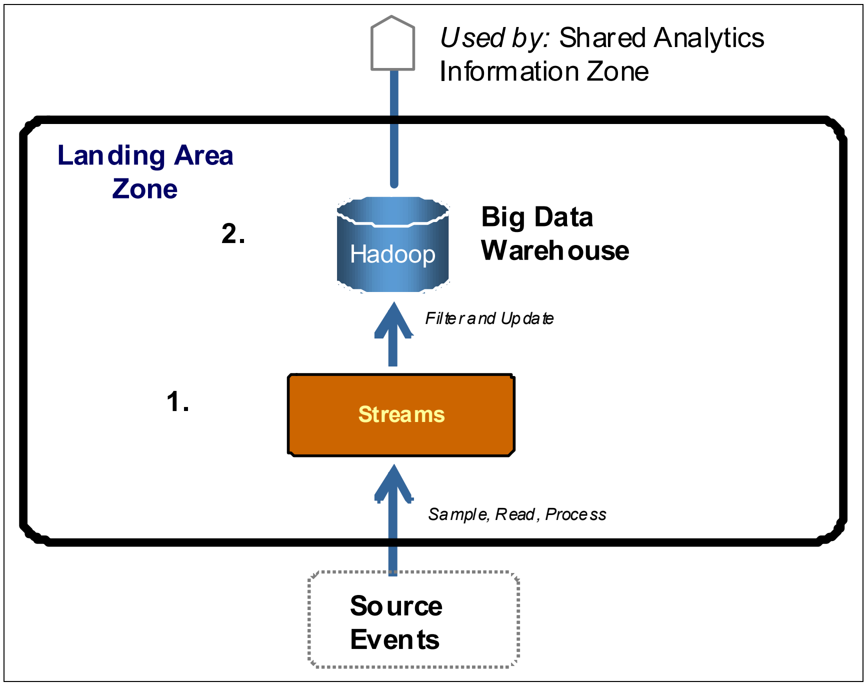


Figure 6-17 Landing Area Zone Stream Filter pattern

The Landing Area Zone containing the Big Data Warehouse is built from Source Events.

The following component details refer to numbers in Figure 6-17:

1. Streams-processed events send arbitrary data to the Real-time Application and filtered data to the Big Data Warehouse as part of a continuous cycle.
2. The Big Data Warehouse HDFS provides columnar access to data, typically using NoSQL requests.

Component pattern 5.1: Landing Area Zone Stream Augmentation

This pattern illustrates a typical streaming and augmentation process (see Figure 6-18).

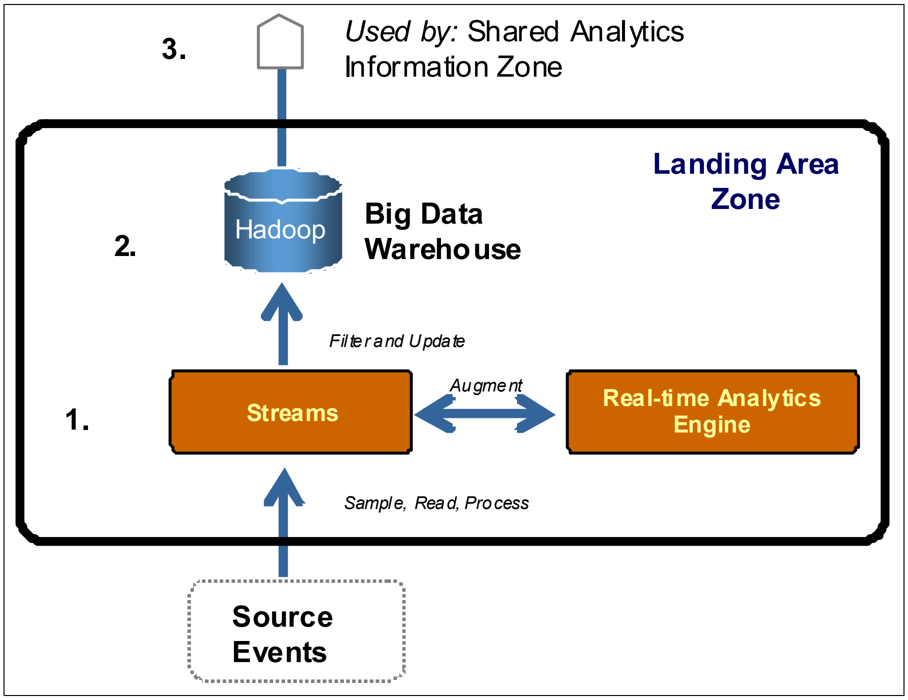


Figure 6-18 Landing Area Zone Stream Augmentation pattern

Consider these summary points about this solution pattern:

- This is a subpattern variant of Component Pattern 5: Landing Area Zone Stream Filter.
- The Landing Area Zone containing the Big Data Warehouse is built from Source Events that are optionally processed in conjunction with one or more Real-time Applications (not otherwise relevant to this pattern). The only stipulation is that data must be processed at least as fast as it arrives.

The following component details refer to numbers in Figure 6-18:

1. Source Events are generally high-volume and granular data elements in unstructured, semi-structured, and unstructured formats.
2. Streams-processed events send arbitrary data to the Real-time Analytics Engine and filtered data to the Big Data Warehouse as part of a continuous cycle.
3. The Big Data Warehouse HDFS provides columnar access to data, typically using NoSQL requests.

Component pattern 5.2: Landing Area Zone Warehouse Augmentation

This pattern illustrates an ETL and augmentation process for moderate velocity data (see Figure 6-19).

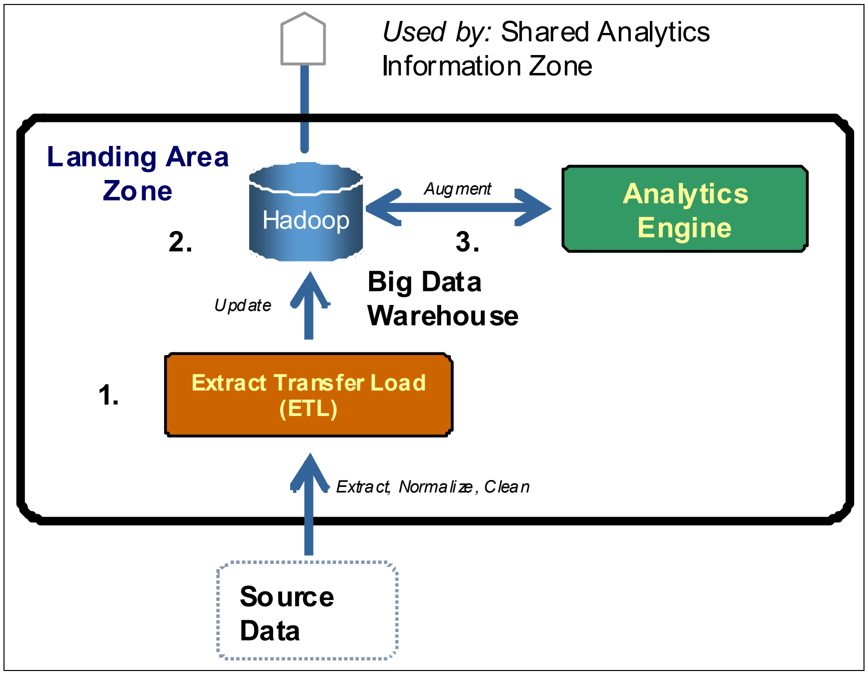

Used by: Shared Analytics Information Zone
Landing Area Zone
Hadoop
Augment
Analytics Engine
2.
3.
Update
Big Data Warehouse
1.
Extract Transfer Load (ETL)
Extract, Normalize, Clean
Source Data

Figure 6-19 Landing Area Zone Warehouse Augmentation pattern

Consider these summary points about this solution pattern:

- This is a subpattern variant of Component Pattern 5.1: Landing Area Zone Stream Augmentation.
- Augmentation of data is not necessarily as time critical as it is in a direct augmentation scenario.

The following component details refer to numbers in Figure 6-19:

1. ETL processes use software as a service (SaaS), Search, SQL, and other accepted methods over well-defined Source Data to build the Big Data Warehouse. Update intervals vary depending on requirements.
2. The Master Big Data Warehouse HDFS provides columnar access to data, typically using NoSQL requests.
3. The analytics engine augments, aggregates, and modifies data in the Master Big Data Warehouse.

Component pattern 6: Landing Area Zone Index

This pattern illustrates a data indexing process (see Figure 6-20).

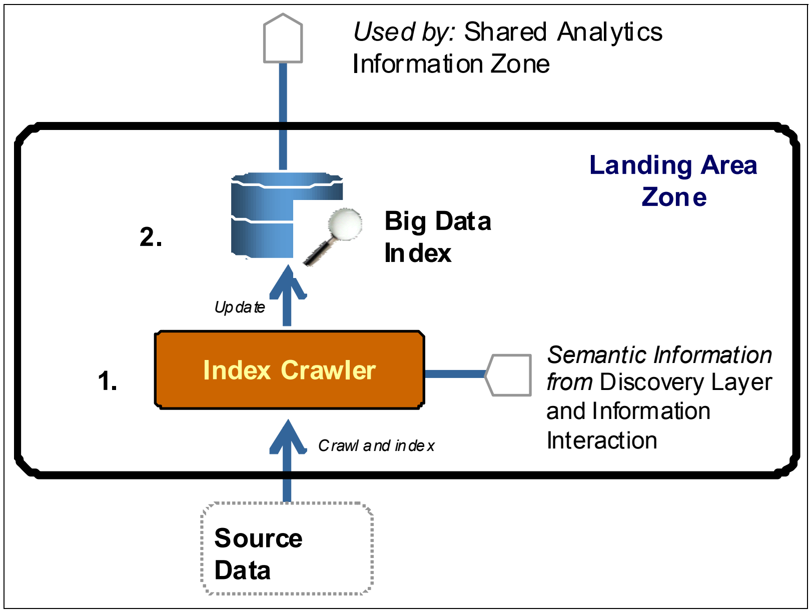

Used by: Shared Analytics Information Zone
Landing Area Zone
2.
Big Data Index
Update
1.
Index Crawler
Semantic Information from Discovery Layer and Information Interaction
Crawl and index
Source Data

Figure 6-20 Landing Area Zone Index pattern

A domain expert (typically a Data Scientist) constructs the Big Data Table using various tools and sources.

The following component details refer to numbers in Figure 6-20:

1. The Exploration Mart is built from Landing Area Zone sources and other sources.
2. Query and Discovery tools aid with search, extraction, and disambiguation of uncertain sources and relationships.
3. Presentation, Visualization, and Sharing are optionally performed over the Exploration Mart.
4. The Exploration Mart is optionally exported to other applications and tools.

Component pattern 7: Exploration Mart

This pattern illustrates a data exploration process that creates a summary table (see Figure 6-21).

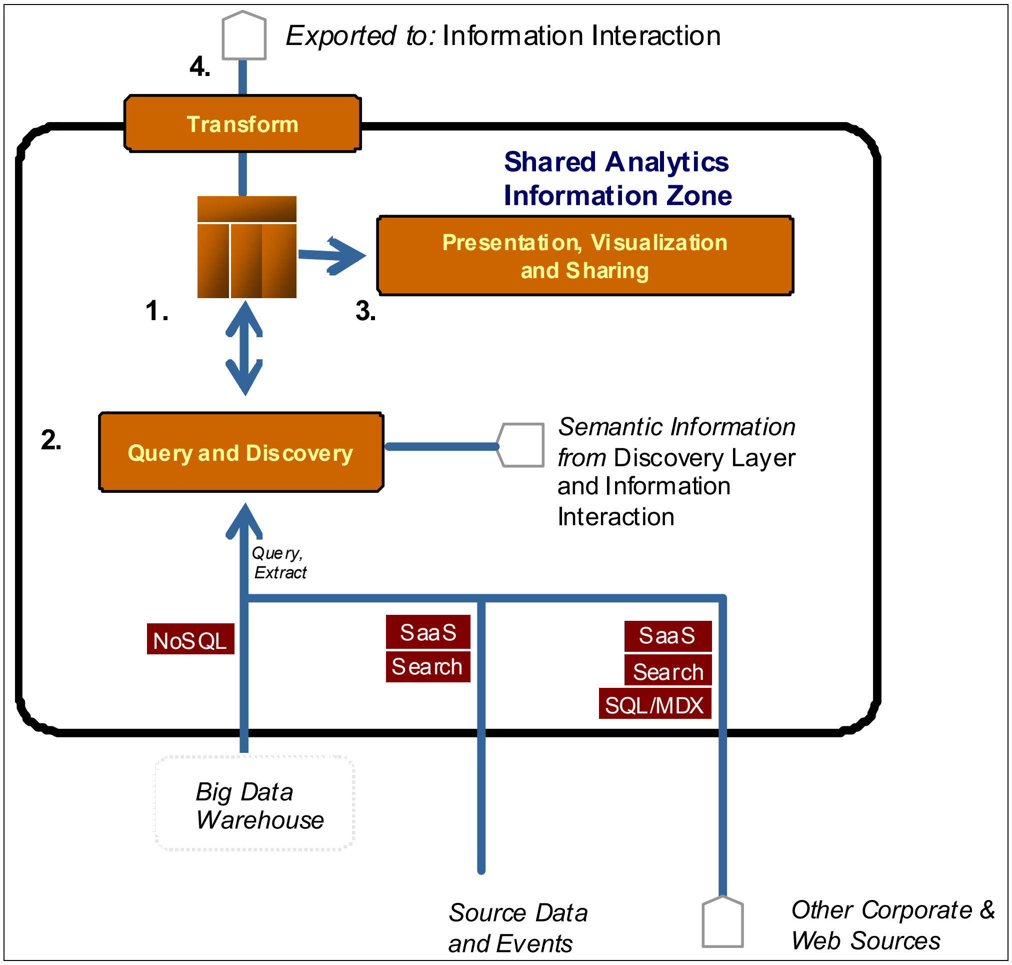

Exported to: Information Interaction
4.
Transform
Shared Analytics Information Zone
Presentation, Visualization and Sharing
1.
3.
2.
Query and Discovery
Semantic Information from Discovery Layer and Information Interaction
Query, Extract
NoSQL
SaaS
Search
SaaS
Search
SQL/MDX
Big Data Warehouse
Source Data and Events
Other Corporate & Web Sources

Figure 6-21 Exploration Mart pattern

A domain expert (typically a Data Scientist) constructs the Big Data Table using various tools and sources.

The following component details refer to numbers in Figure 6-21:

1. The Exploration Mart is built from Landing Area Zone sources and other sources.
2. Query and Discovery tools aid with search, extraction, and disambiguation of uncertain sources and relationships.
3. Presentation, Visualization, and Sharing are optionally performed over the Exploration Mart.
4. The Exploration Mart is optionally exported to other applications and tools.

Component pattern 7.1: Analytics Mart

This pattern illustrates a data exploration process that creates a summary warehouse (see Figure 6-22).

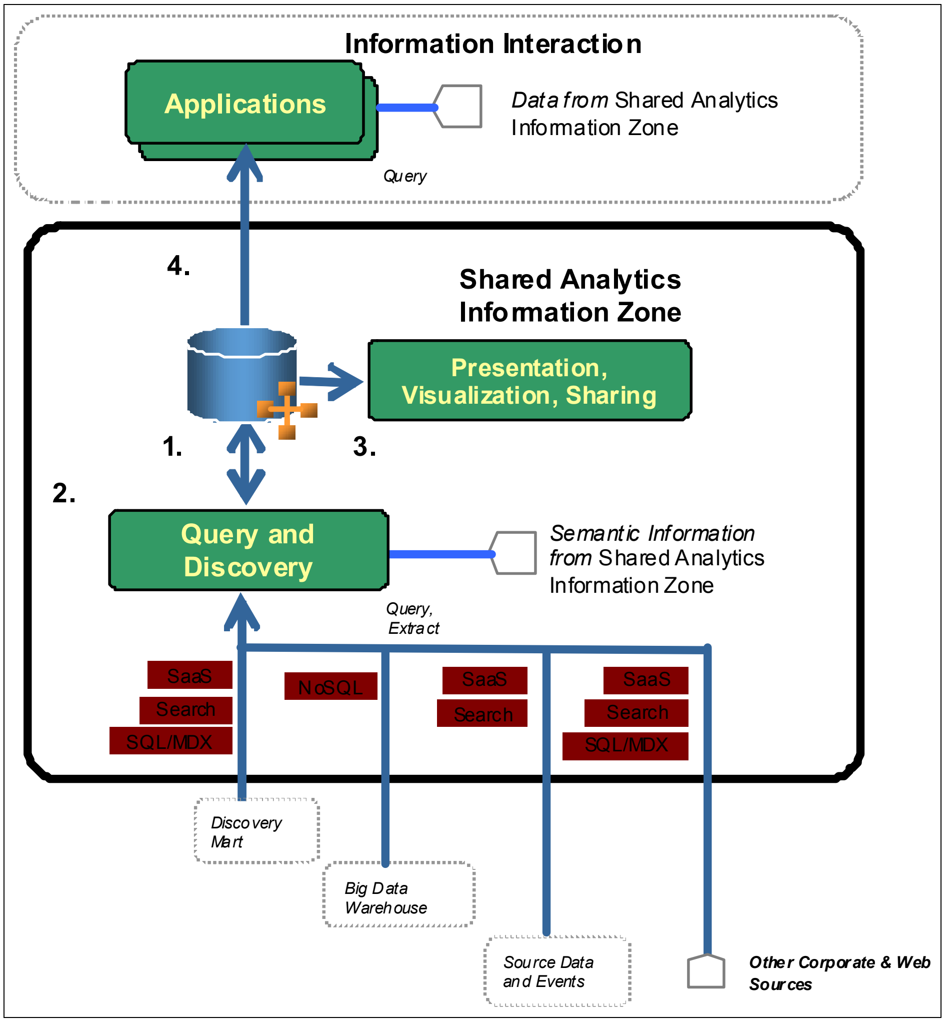

Information Interaction
Applications
Data from Shared Analytics Information Zone
Query
4.
Shared Analytics Information Zone
Presentation, Visualization, Sharing
1.
3.
2.
Query and Discovery
Semantic Information from Shared Analytics Information Zone
Query, Extract
SaaS
Search
SQL/MDX
NoSQL
SaaS
Search
SaaS
Search
SQL/MDX
Discovery Mart
Big Data Warehouse
Source Data and Events
Other Corporate & Web Sources

Figure 6-22 Analytics Mart pattern

Consider these summary points about this solution pattern:

- This is a subpattern variant of Component Pattern 7: Exploration Mart where exploration is moved to the information interaction area.
- A domain expert (typically aligned with the line of business) constructs BA Analysis data using various tools and sources.

The following component details refer to numbers in Figure 6-22:

1. The Analytics Mart is built from Discovery Layer Landing Area Zone sources and other sources.
2. Query and Discovery tools aid with search, extraction, and disambiguation of uncertain sources and relationships.
3. Presentation, Visualization, and Sharing are optionally performed over the Analytics Mart.
4. The Analytics Mart is optionally shared to other components in the information interaction area.

Component pattern 8: Report Mart

This pattern illustrates how to build a traditional summary report mart (see Figure 6-23).

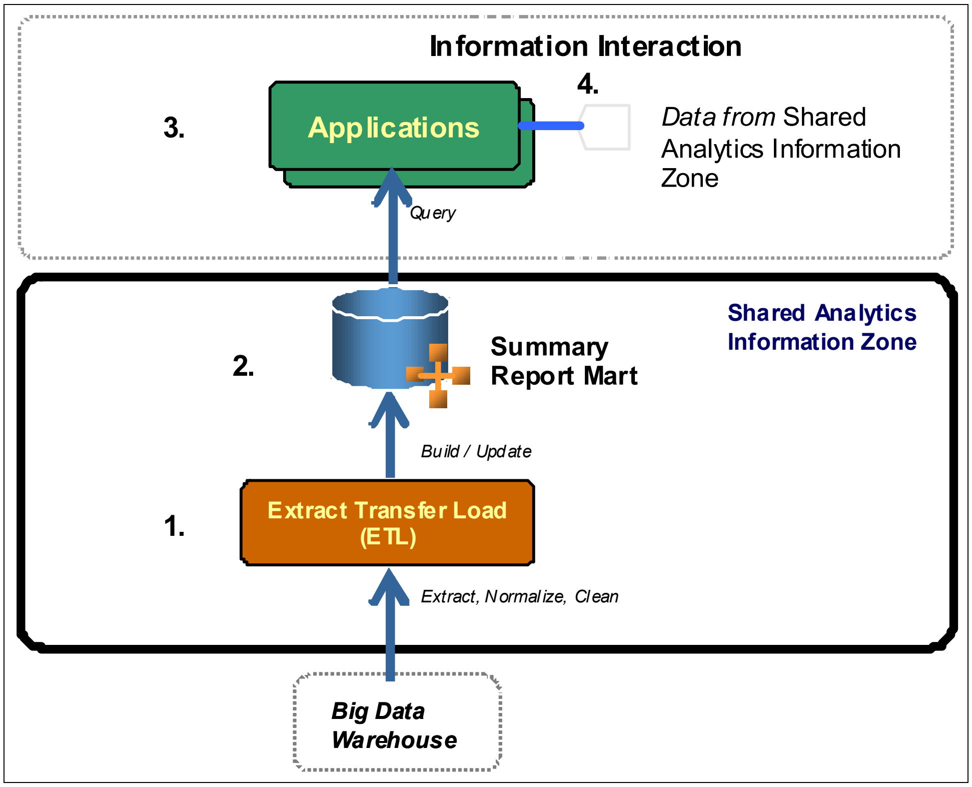

Information Interaction
3.
Applications
4.
Data from Shared Analytics Information Zone
Query
Shared Analytics Information Zone
2.
Summary Report Mart
Build / Update
1.
Extract Transfer Load (ETL)
Extract, Normalize, Clean
Big Data Warehouse

Figure 6-23 Report Mart pattern

Applications exist in the Reporting, BA, finance, GRC, or similar domains.

The following component details refer to numbers in Figure 6-23:

1. ETL batch processes are used to load the Data Mart from the Big Data Warehouse HDFS.
2. There is a Summary Report Mart with a schema that follows modeled BA elements for subsequent information interaction. Options include In-Memory, Dynamic, and Traditional.
3. Applications use SQL/MDX queries.
4. Data is optionally merged from other Discovery Layer and BA sources.

Component pattern 8.1: Virtual Report Mart

This pattern illustrates how to build a virtual report mart over existing data using SQL (see Figure 6-24).

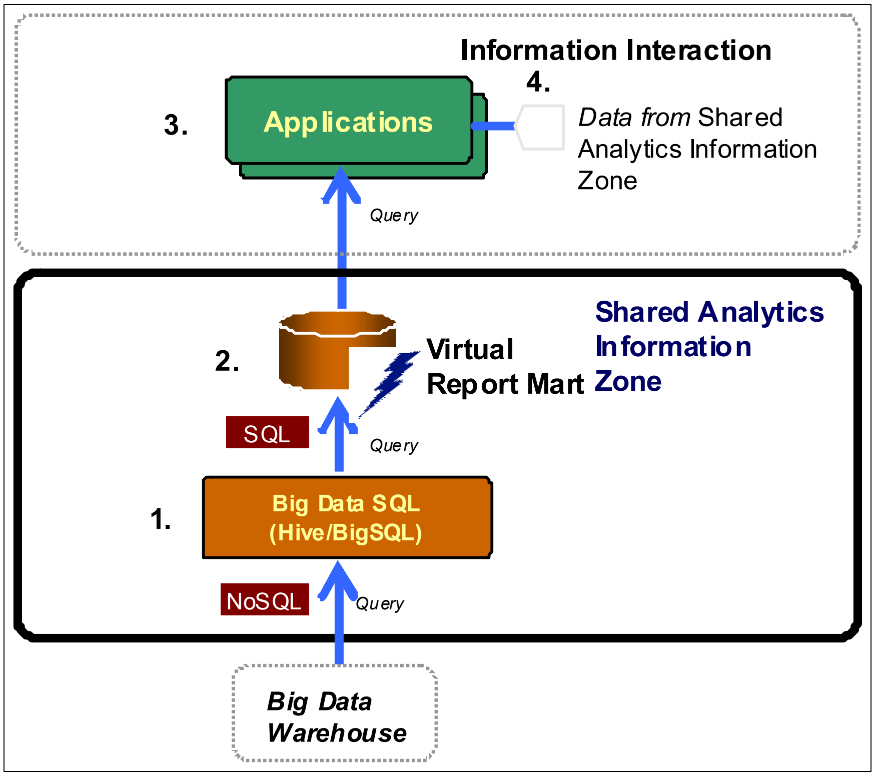


Figure 6-24 Virtual Report Mart pattern

Consider these summary points about this solution pattern:

- This is a subpattern variant of Component Pattern 8: Report Mart.
- Applications exist in the Reporting, BA, finance, GRC, or similar domains.

The following component details refer to numbers in Figure 6-24:

1. The Big Data SQL adapter provides SQL access to the Big Data Warehouse HDFS.
2. There is a Virtual Report Mart with a schema that follows modeled BA elements for subsequent information interaction.
3. Reporting and Analytics use SQL/MDX queries.
4. Data is optionally merged from other Discovery Layer and BA sources.

Component pattern 8.2: Virtual Search Mart

This pattern illustrates how to build a virtual report mart over existing data using search components (see Figure 6-25).

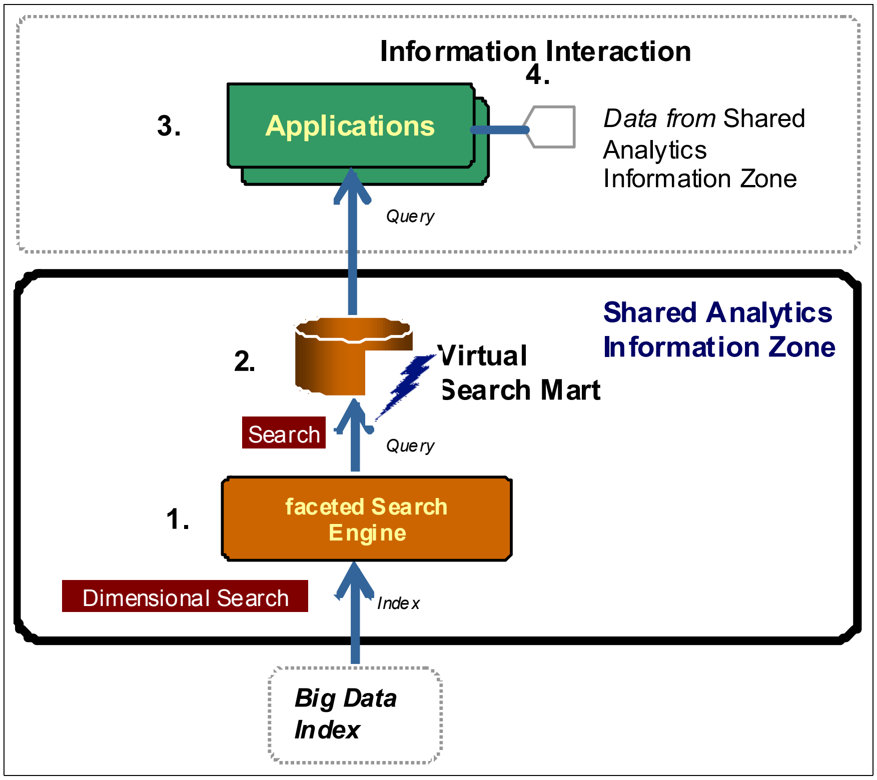


Figure 6-25 Virtual Search Mart pattern

Consider these summary points about this solution pattern:

- This is a subpattern variant of Component Pattern 8: Report Mart.
- Applications exist in Reporting, BA, finance, GRC, or similar domains.

The following component details refer to numbers in Figure 6-25:

1. A Faceted Search Engine provides dimensional results over the Big Data Index.
2. There is a Virtual Report Mart with a schema that follows modeled BA elements for subsequent information interaction.
3. Reporting and Analytics use Dimensional Search Queries.

Component pattern 9: Predictive Analytics

This pattern illustrates how analytics augment warehouse data (see Figure 6-26).

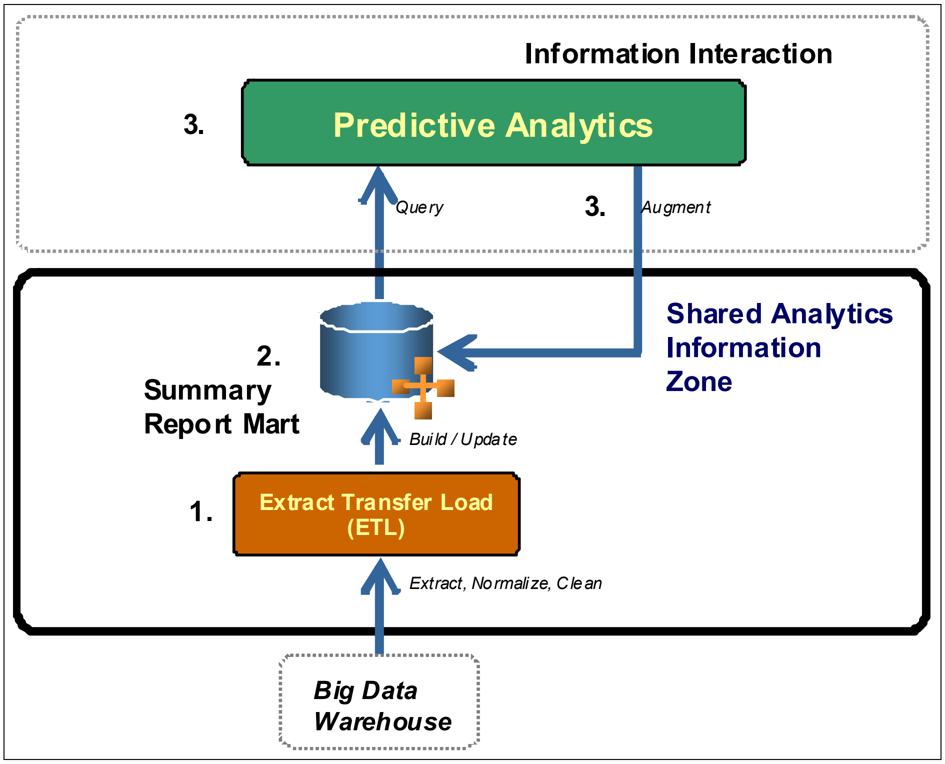


Figure 6-26 Predictive Analytics pattern

Consider these summary points about this solution pattern:

- Applications exist in Reporting, BA, finance, GRC, or similar domains.
- Predictive Analytics augment existing Data Marts.

The following component details refer to numbers in Figure 6-26:

1. ETL batch processes load the Data Mart from the Big Data Warehouse HDFS.
2. There is a Summary Report Mart with a schema that follows modeled BA elements for subsequent information interaction. Options include In-Memory, Dynamic, and Traditional.
3. Applications use SQL/MDX queries.
4. Discovery Layer data is augmented.

7

Picking a solution pattern

This chapter is intended to help architects and designers choose the solution pattern that best meets their requirements.

The primary decision points in this process are presented in Figure 7-1 on page 70. The initial data ingestion decision is depicted in Figure 7-2 on page 72. The requirements for analytics, data longevity, and velocity are then considered, leading to the pattern choices shown in the remaining figures in the chapter.

7.1 Data ingestion

As shown in Figure 7-1, you make choices for data ingestion based on how the data will be used, such as for analytics or reporting, and the characteristics of the data being ingested, such as its velocity and transience. These choices then lead to the flows for data preparation, data streamlining, and data alignment.

If there is a need to iterate data for analytics and reporting, Figure 7-2 on page 72, Part B, covers the flow for aligning the data for use in a data mart or to normalize the data to create virtual tables.

If the data has a short period of validity, Figure 7-3 on page 73 describes the flow for handling streaming data.

If the data does not require iterative processing, is not short lived, and is not coming at a high velocity, Figure 7-2 on page 72, Part A, describes the flow for preparing the data so that it can be used either for search drive analysis or dimensional analysis.

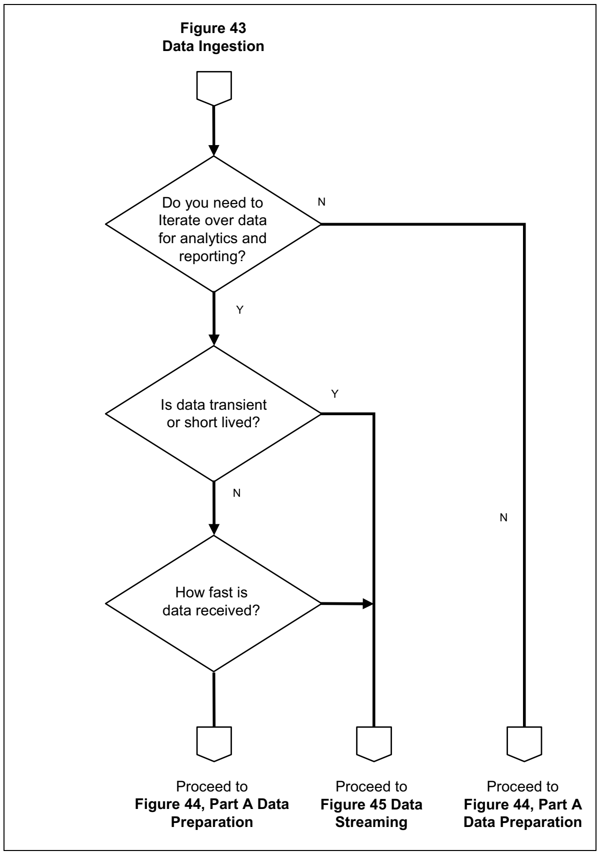


Figure 43
Data Ingestion
Do you need to Iterate over data for analytics and reporting?
N
Y
Is data transient or short lived?
Y
N
N
How fast is data received?
Proceed to Figure 44, Part A Data Preparation
Proceed to Figure 45 Data Streaming
Proceed to Figure 44, Part A Data Preparation

Figure 7-1 Picking a solution pattern, Data Ingestion

7.2 Data preparation and data alignment

In Figure 7-2 on page 72, you make selections based on the data structure, the ease with which associations to existing metadata can be created, or whether a set of column values can be created to look up the unstructured data.

Structured data will typically be used for a wide variety of reporting and analytics and will be handled separately. The flow for selecting a pattern for handling structured data is depicted in Figure 7-2 on page 72, Part A.

Some unstructured data can be easily parsed and mapped to existing metadata (for example, free-form data can have a simple regex applied to detect addresses), after which it can be processed like structured data, as shown in Figure 7-2 on page 72, Part A.

Other types of unstructured data might require analytical processing to determine associations (for example, converting a cellular phone tower ID to latitude and longitude coordinates). After these analytical and statistical associations are completed, the data can be processed like structured data, as shown in Figure 7-2 on page 72, Part A.

Other types of unstructured data might require building search facets or creating indexes to make better sense of it. The flow for selecting a pattern for this type of data is described in Figure 7-4 on page 74.

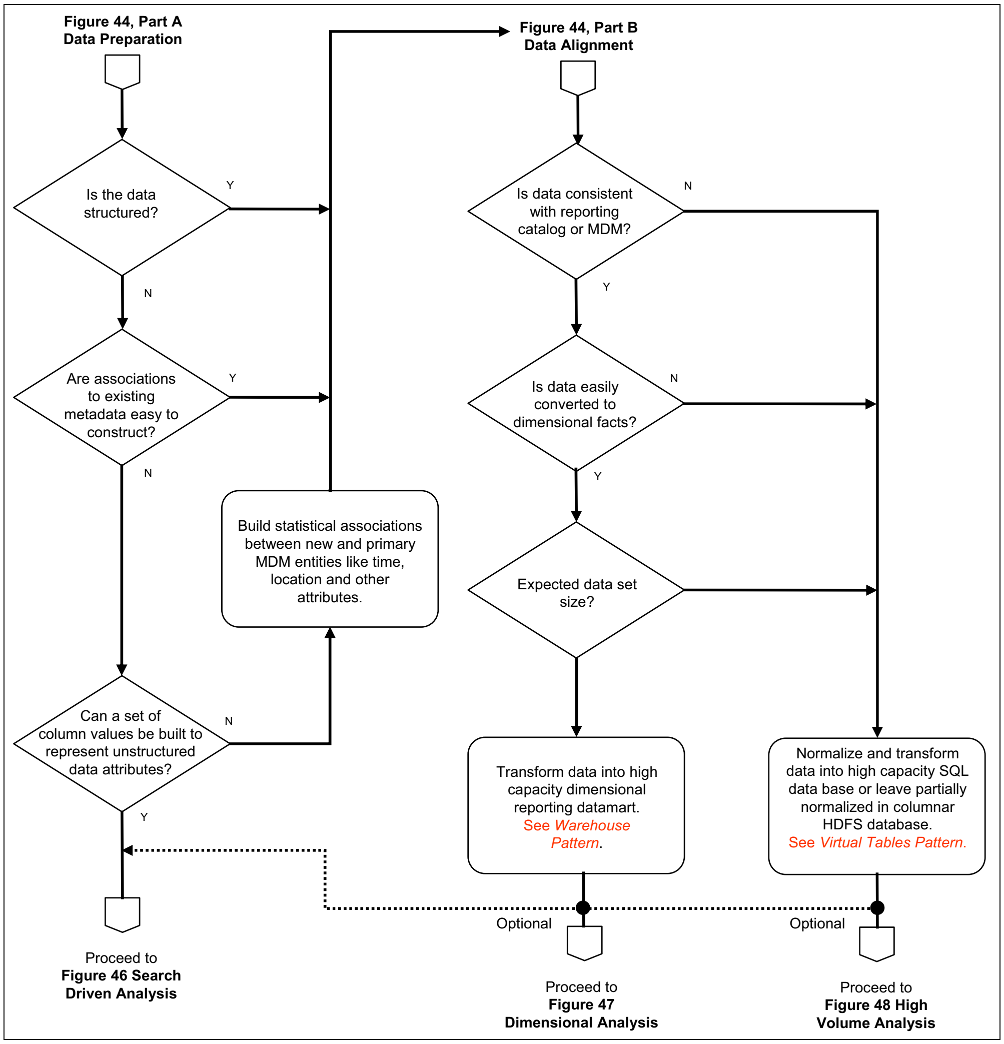


Figure 44, Part A
Data Preparation
Is the data structured?
Y
N
Are associations to existing metadata easy to construct?
Y
N
Build statistical associations between new and primary MDM entities like time, location and other attributes.
Can a set of column values be built to represent unstructured data attributes?
N
Y
Proceed to
Figure 46 Search Driven Analysis
Figure 44, Part B
Data Alignment
Is data consistent with reporting catalog or MDM?
N
Y
Is data easily converted to dimensional facts?
N
Y
Expected data set size?
Transform data into high capacity dimensional reporting datamart. See Warehouse Pattern.
Normalize and transform data into high capacity SQL data base or leave partially normalized in columnar HDFS database. See Virtual Tables Pattern.
Optional
Optional
Proceed to
Figure 47 Dimensional Analysis
Proceed to
Figure 48 High Volume Analysis

Figure 7-2 Picking a solution pattern: Data preparation and data alignment

7.3 Data streaming

In Figure 7-3 on page 73, you make selections based on the nature of the streaming data and the intended usage.

If the streaming data requires low-latency processing, the Low Latency Warehouse Augmentation Pattern is used. There might be additional need for using the data for reporting, master data management, or as fact tables, in which case the selection process described in Figure 7-2, Part A, can be followed.

For streaming data that does not require low-latency processing, the warehouse can either be directly augmented or new data sets can be stored in the Big Data Landing Zone. After that, additional sage selections can be made as described in Figure 7-2, Part A.

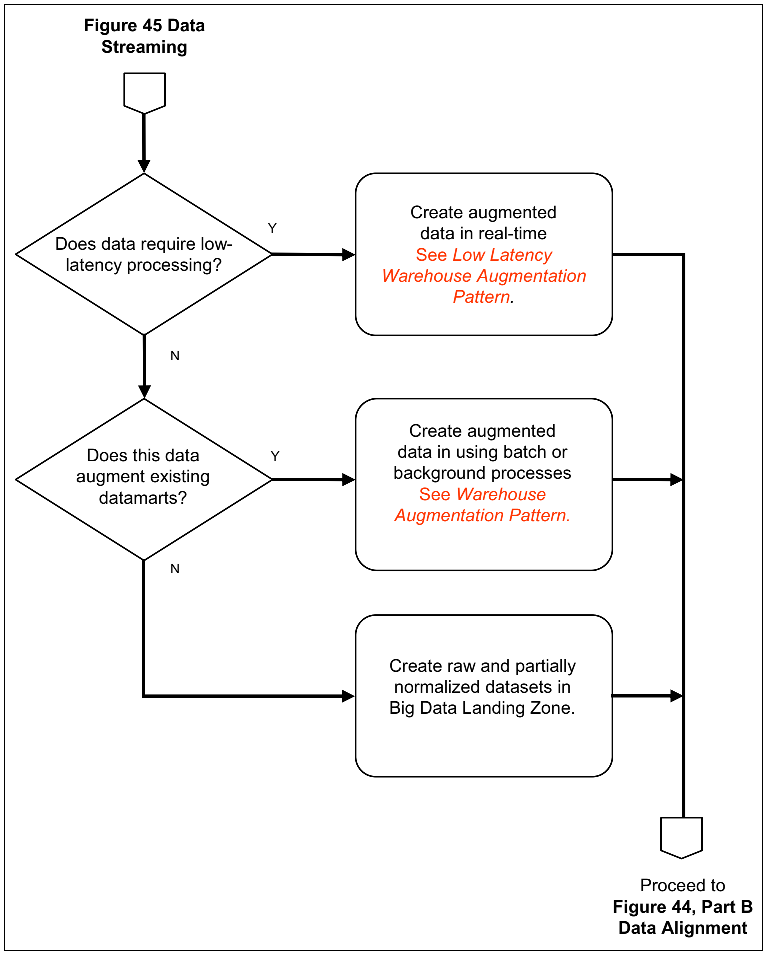


Figure 45 Data Streaming
Does data require low-latency processing?
Y
Create augmented data in real-time
See Low Latency Warehouse Augmentation Pattern.
N
Does this data augment existing datamarts?
Y
Create augmented data in using batch or background processes
See Warehouse Augmentation Pattern.
N
Create raw and partially normalized datasets in Big Data Landing Zone.
Proceed to
Figure 44, Part B Data Alignment

Figure 7-3 Picking a solution pattern: Data streaming

7.4 Search-driven analysis

In Figure 7-4 on page 74, you make selections based on whether the incoming unstructured data can be mapped to current reporting dimensions. If the data cannot be mapped, new search facets need to be developed to explore data and find new patterns within it. Otherwise, the data can be indexed and topic clusters can be created for Search Driven Analysis.

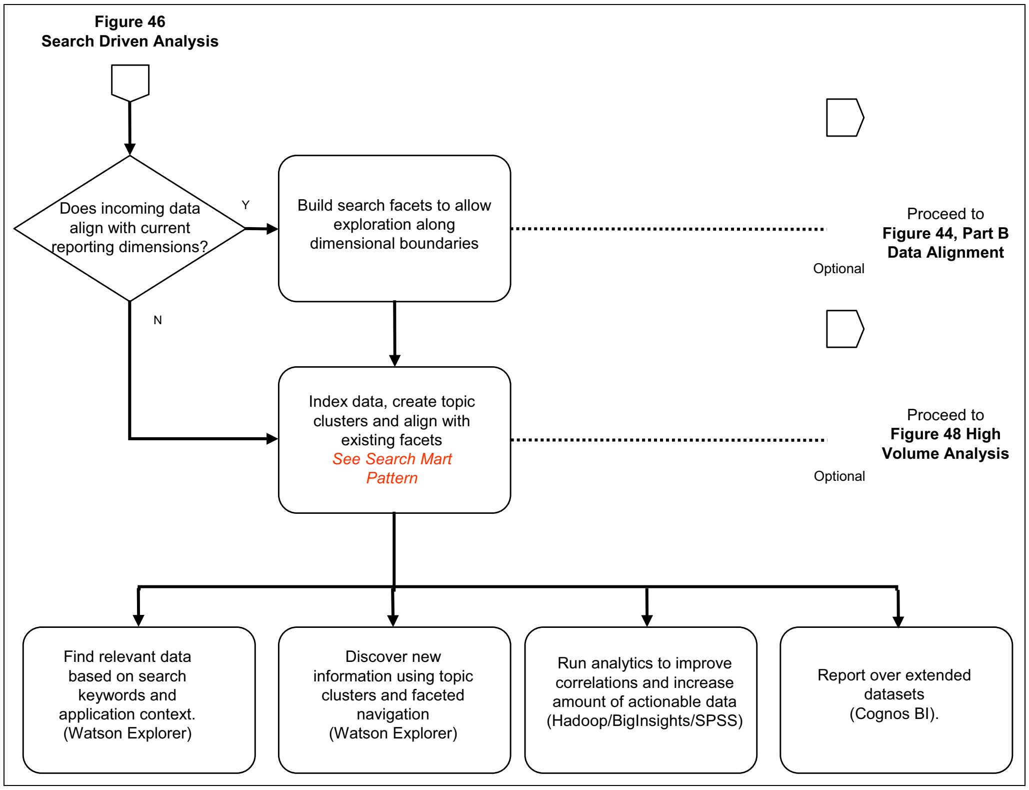


Figure 46
Search Driven Analysis
Does incoming data align with current reporting dimensions?
Y
N
Build search facets to allow exploration along dimensional boundaries
Optional
Proceed to Figure 44, Part B Data Alignment
Index data, create topic clusters and align with existing facets
See Search Mart Pattern
Optional
Proceed to Figure 48 High Volume Analysis
Find relevant data based on search keywords and application context. (Watson Explorer)
Discover new information using topic clusters and faceted navigation (Watson Explorer)
Run analytics to improve correlations and increase amount of actionable data (Hadoop/BigInsights/SPSS)
Report over extended datasets (Cognos BI).

Figure 7-4 Picking a solution pattern: Search-driven analysis

7.5 Dimensional analysis

Figure 7-5 on page 75 provides the flow for analysis of the incoming data. If the incoming data can be aligned with current reporting dimensions, virtual tables can be created in Hadoop, as explained in the Virtual Tables pattern. Optionally, the incoming data can be used to augment existing data warehouses, for example, in these update patterns:

- Discovery Tables Pattern: Used to discover new information in the incoming data by using navigation, discovery, and visualization tools
- Warehouse Augmentation Pattern: Used to update existing data warehouses (for example, to enrich existing data)
- Transaction Merge Update Pattern: Used to augment transactional systems based on information extracted from incoming data
- Delivering business intelligence (BI) reports over extended data sets

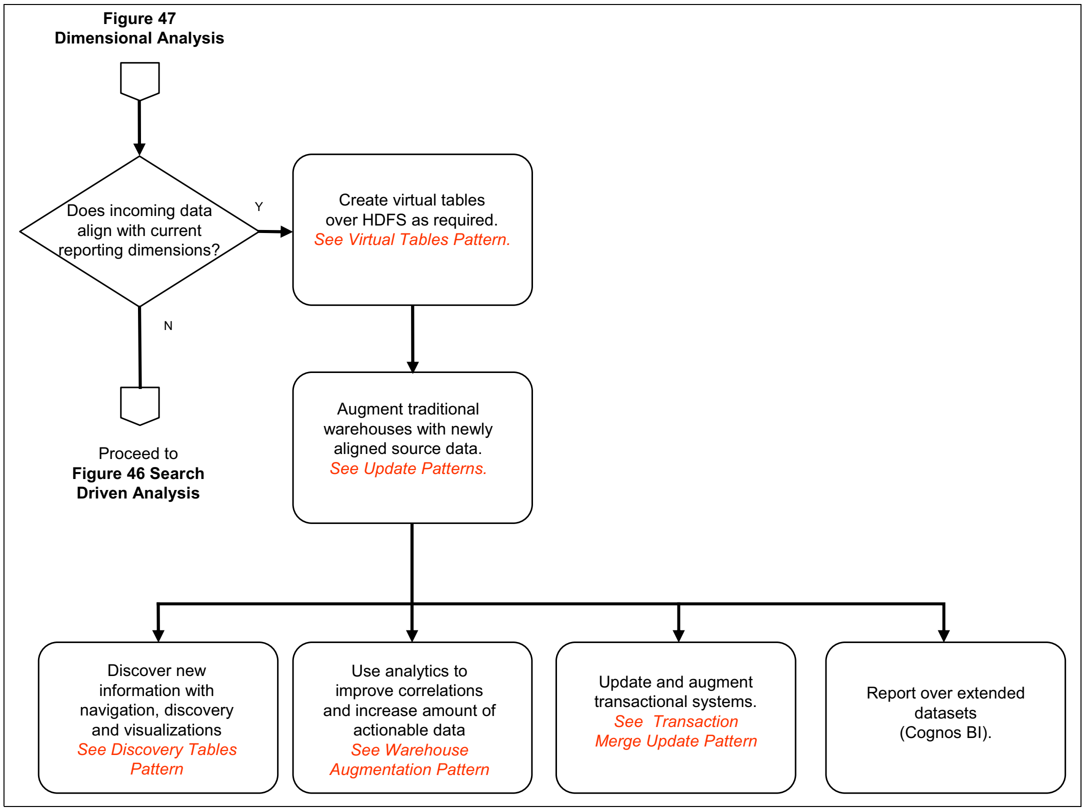


Figure 47
Dimensional Analysis
Does incoming data align with current reporting dimensions?
Y
N
Create virtual tables over HDFS as required.
See Virtual Tables Pattern.
Proceed to
Figure 46 Search Driven Analysis
Augment traditional warehouses with newly aligned source data.
See Update Patterns.
Discover new information with navigation, discovery and visualizations
See Discovery Tables Pattern
Use analytics to improve correlations and increase amount of actionable data
See Warehouse Augmentation Pattern
Update and augment transactional systems.
See Transaction Merge Update Pattern
Report over extended datasets (Cognos BI).

Figure 7-5 Picking a solution pattern: Dimensional analysis

7.6 High-volume analysis

Figure 7-6 on page 76 provides the possible flow for maintaining the data in the Big Data Landing Zone and employing it for various purposes, for example:

- Finding relevant data based on application context by employing the application code in Java, SQL, and so on
- Discovering new correlations, visualizations, and dashboards using analytical tools, such as SPSS or Watson Data Explorer
- Running analyses to improve existing models and improve actionable insight
- Aligning data as required for further discovery using the Discovery Tables Pattern
- Augmenting data warehouses using the Update patterns

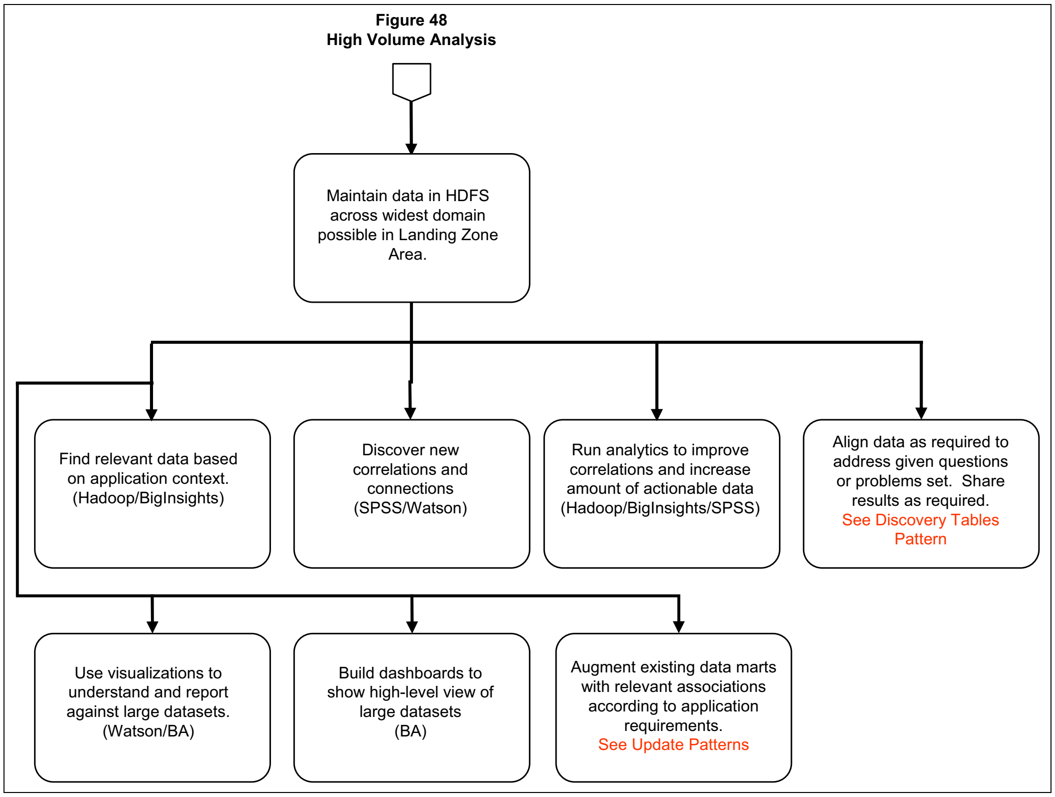


Figure 48
High Volume Analysis
Maintain data in HDFS across widest domain possible in Landing Zone Area.
Find relevant data based on application context. (Hadoop/BigInsights)
Discover new correlations and connections (SPSS/Watson)
Run analytics to improve correlations and increase amount of actionable data (Hadoop/BigInsights/SPSS)
Align data as required to address given questions or problems set. Share results as required. See Discovery Tables Pattern
Use visualizations to understand and report against large datasets. (Watson/BA)
Build dashboards to show high-level view of large datasets (BA)
Augment existing data marts with relevant associations according to application requirements. See Update Patterns

Figure 7-6 Picking a solution pattern: High-volume analysis

8

NoSQL technology

When we look at handling data, we think of relational database technologies. Relational databases, such as Oracle, IBM DB2®, and MySQL, can solve most traditional data management needs and are still widely used.

However, when it comes to big data, the sheer scale of data that is collected and analyzed creates unique challenges. Numerous technologies have been developed over the last 10 - 15 years to address these challenges. This section provides an overview of various NoSQL data stores that can be used.

8.1 Definition and history of NoSQL

The “no” in NoSQL means different things for different people. Initially, these databases did not support SQL, but now the “no” means *not only SQL*.

NoSQL data stores are next generation data stores that typically are non-relational, distributed, open source, and horizontally scalable. Characteristics of these databases include some combination of being schema-free, offering replication support, simple APIs, achieving eventual consistency/BASE, and support for massive amounts of data.

The need for NoSQL emerged with web companies that were trying to scale their databases to match their growing business needs without having to buy expensive hardware. The explosion of new web applications and later, social networks created demand for handling massive amounts of data, including fast querying and analytical capabilities. Even unstructured data, such as multimedia audio and video content, had to be accommodated.

The term NoSQL first appeared in 1998, when it was used to describe a relational database developed by Carlos Strozzi that provided no form of the SQL language for querying. This initial usage is considered unrelated to the NoSQL movement of today. The term was reintroduced by Rackspace in 2009 for one of its conferences.

Figure 8-1 provides a timeline of some of the key NoSQL technologies.

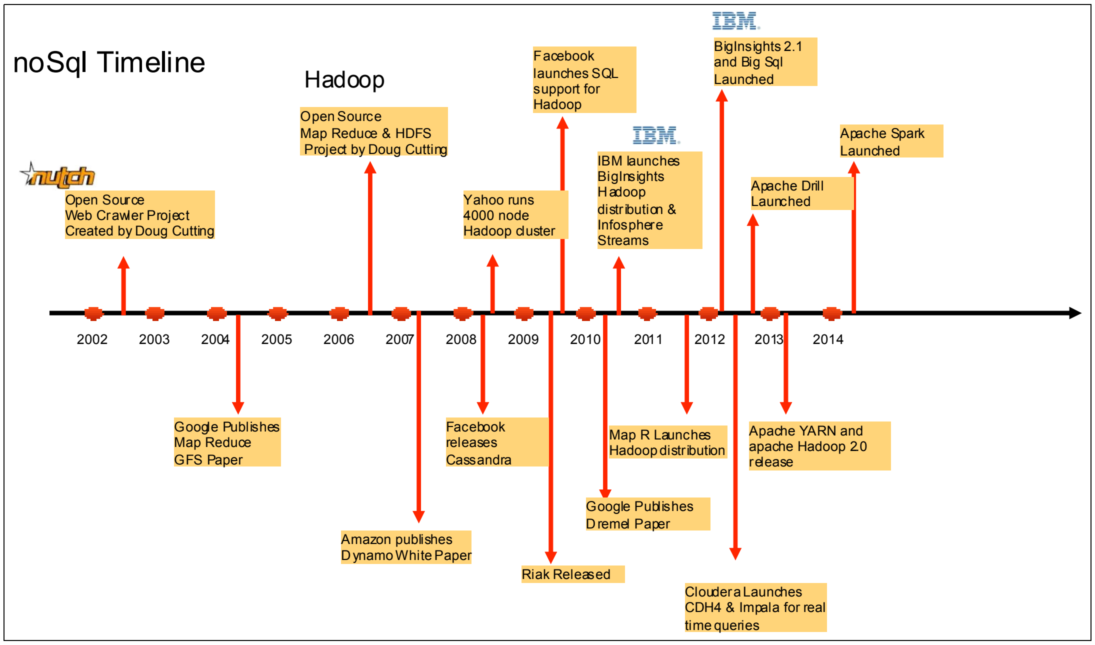

noSql Timeline
Hadoop
nutch
Open Source Web Crawler Project Created by Doug Cutting
Open Source Map Reduce & HDFS Project by Doug Cutting
Yahoo runs 4000 node Hadoop cluster
Facebook launches SQL support for Hadoop
IBM
IBM launches BigInsights Hadoop distribution & Infosphere Streams
IBM
BigInsights 2.1 and Big Sql Launched
Apache Drill Launched
Apache Spark Launched
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
Google Publishes Map Reduce GFS Paper
Facebook releases Cassandra
Amazon publishes Dynamo White Paper
Riak Released
Map R Launches Hadoop distribution
Google Publishes Dremel Paper
Cloudera Launches CDH4 & Impala for real time queries
Apache YARN and apache Hadoop 2.0 release

Figure 8-1 NoSQL timeline

The growth of web applications, such as search and social media, led to the need for data stores that can scale out across multiple servers. And as the size of the database grew, it became more difficult to scale the traditional relational database management system (RDBMS) servers. The only remaining approach was to try to distribute the data. Yet creating a distributed database and ensuring that data is synchronized across nodes can be challenging.

Relational databases, such as Oracle and DB2, support distributed databases and ensure that all of the nodes are in sync using the 3-phase commit. This approach is fine when the nodes of the database are fairly close together but can lead to performance issues when the nodes are geographically distributed. Applications used by banks and financial institutions, which need to guarantee data synchronization across nodes, still rely heavily on relational databases.

A set of properties referred to as *Atomicity, Consistency, Isolation, and Durability* (ACID) apply specifically to database transactions. They are defined this way:

- *Atomicity*: Everything in a transaction must happen successfully or none of the changes are committed. This avoids situations in which a transaction that changes multiple pieces of data suddenly fails halfway through the process, resulting in only a few, incomplete changes being made.
- *Consistency*: The data will only be committed if it passes all the rules established for the database, such as data types, triggers, and constraints.
- *Isolation*: One transaction will not affect another transaction by changing data that the other operation relies on, and users will not see partial results of a transaction in progress (depending on the isolation mode being used).
- *Durability*: After data is committed, it is durably stored and kept safe against errors, crashes, or other software malfunctions within the database.

8.1.1 The CAP theorem

Eric Brewer’s CAP theorem describes the problem that is associated with a distributed database. The theorem is a set of basic requirements that describe any distributed system, not just storage and database systems. Here are the requirements:

- Consistency

  All of the servers in the system will have the same data, so anyone using the system will get the latest data regardless of any updates that are occurring. Relational databases, such as DB2, Oracle, Informix®, and MySQL, enforce consistency. However, these databases scale vertically, not horizontally. Techniques, such as *sharding* (a process of storing data records across multiple machines), have been used to distribute relational databases. Databases, such as Amazon Dynamo DB and Cassandra, distribute data and rely on what is called *eventual consistency*. Eventual consistency relies on deciding to which row everything eventually converges. This is difficult in the case of two writers writing at the same time, so timestamps and vector clocks are used to resolve issues.

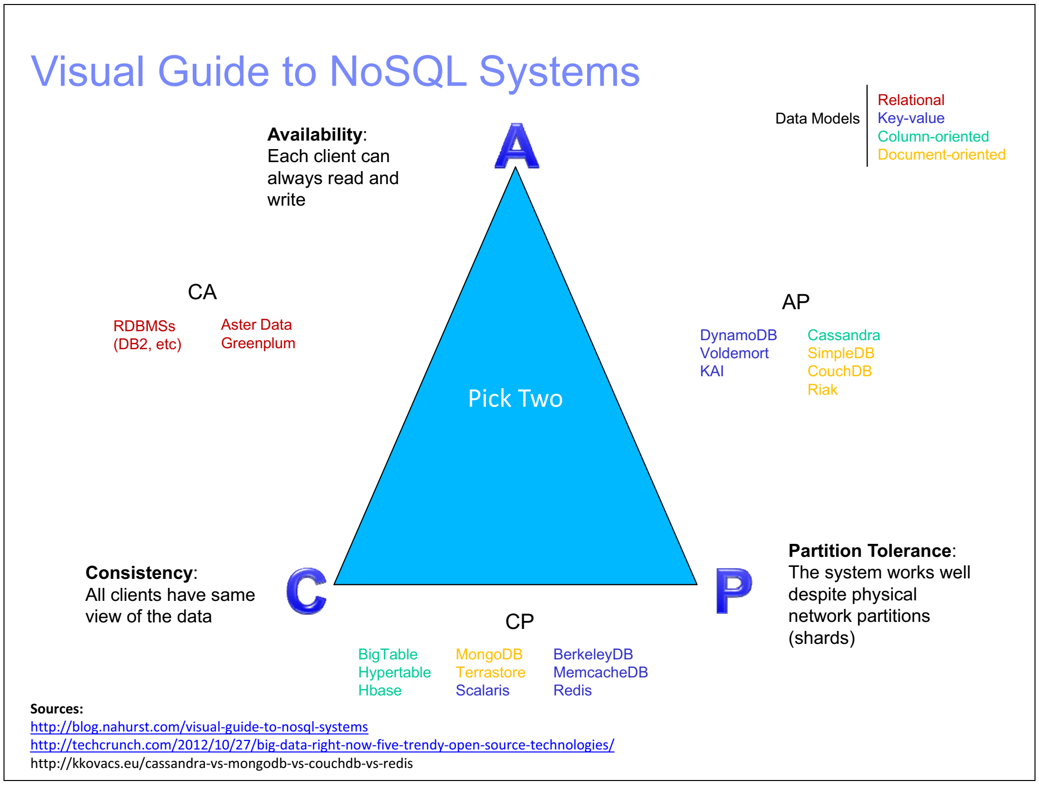
- Availability

  All of the servers will always return the data they have, even if it is not the latest data or consistent across the system. The question to ask is, “If some nodes fail, does everything still work or does the system come to a halt?”

- Partition tolerance

  The system as a whole continues to operate even if individual servers fail or cannot be reached. The question to ask is, “If two nodes of the distributed system cannot talk to each other, can they move forward on their own?”

Figure 8-2 on page 80 illustrates how the CAP theorem applies to this discussion.



Visual Guide to NoSQL Systems

Sources:
http://blog.nahurst.com/visual-guide-to-nosql-systems
http://techcrunch.com/2012/10/27/big-data-right-now-five-trendy-open-source-technologies/
http://kkovacs.eu/cassandra-vs-mongodb-vs-couchdb-vs-redis

Figure 8-2 CAP theorem

Theoretically, it is impossible to meet all three of the theorem's requirements, so two of them must be chosen, and those choices typically dictate the selection of a technology to use. For example, Domain Name System (DNS) servers and web caches are examples of distributed databases that have sacrificed in the area of consistency. Both of them are always available and the system can operate even if a few of the nodes fail. These systems use the concept of optimistic updating with conflict resolution.

So, CAP provides the basic requirements for a distributed system (selecting two of the three options) and ACID is a set of guarantees about the way that transactional operations will be processed. ACID is provided by most classic RDBMS, such as MySQL, DB2, Oracle, and others. These databases are known for storing data in tables of information with relations between tables, and having data queries through some type of SQL.

Modern Internet-based databases, such as NoSQL databases, are instead focused on Basically Available, Soft-State, eventually consistent (BASE). These have no transactions in the classical sense and introduce constraints on the data model to enable better partition schemes, such as the Dynamo system.

8.2 Types of NoSQL databases

NoSQL databases can be classified into the one of these categories:

- Key-Value stores
- Document databases
- Columnar databases
- Graph databases

8.2.1 Key-Value stores

A *Key-Value store* is simply a global collection of Key-Value pairs. The design of this data store was inspired by Amazon's Dynamo and the memcached distributed in-memory cache (memcached provides an in-memory Key-Value store that is not persisted).

Key-Value stores store data values in a system so that those values can later be recalled using a key. The values are arbitrary data that the application can interpret as it sees fit. This simple and schema-less data model allows for easy scaling and simple APIs that can enhance the functionality of existing systems. The data store is useful for providing a low-latency solution. These systems usually provide object replication for recovery, partitioning of data over many machines, and rudimentary persistence. Examples of Key-Value stores are redis, Riak, Scalaris, and Voldemort.

8.2.2 Document databases

Document databases contain documents, which are records that describe both the data in the document and the actual data. Documents can be complex or nested to provide additional subcategories of information about the object being stored. In a relational database system, a schema must be defined before records can be added to the database. The `schema` is the structure of the database described in a formal language. The schema provides a blueprint for the data tables and the relationships between the tables.

Document databases provide a way to add new attributes to an object more easily than with a relational database. Like the Key-Value store, document databases can also partition the data across multiple machines and replicate the data for automatic recovery. A database in a document database is considered a *collection*, which in turn contains documents. Documents can contain nested documents, creating a complex structure. Documents have attributes in a Key/Value format. Unlike a relational database, the data in a document database is not normalized but can be duplicated for faster searches. Examples of document databases are MongoDB, CouchDB, and SimpleDB. IBM Cloudant is based on CouchDB and provides a powerful document database that is also available on the cloud.

8.2.3 Columnar and column-family databases

A *columnar database* stores data in columns instead of rows. The major differences between traditional row-oriented databases and column-oriented databases lie in the performance, storage requirements, and method of modifying the schema.

The columnar database is designed to proficiently write and read data from hard disk storage, which can speed up the time to return a query. A major benefit of a columnar database is that it helps compress the data, which accelerates columnar operations, such as MIN, MAX, SUM, COUNT, and AVG. Another benefit is that column-oriented database management systems (DBMS) are self-indexing systems and use less disk space than relational DBMSs (RDBMS).

Examples of columnar databases include Vertica, Greenplum, and Aster. In addition, traditional database vendors, such as Oracle and IBM, now offer columnar storage options within their database offerings. Hbase is another type of column-distributed data store that uses the Hadoop Distributed File System (HDFS) as the data store. It is modeled after Google BigTable and can host large tables (billions of columns and rows) because it is layered on Hadoop clusters of commodity hardware. Other examples include Cassandra and Amazon DynamoDB.

8.2.4 Graph databases

Graph databases, unlike NoSQL and relational databases, are designed for lightning-fast access to complex data in social networks, recommendation engines, and networked systems.

Graph theory dates back to 1735, when Leonard Euler solved the Seven Bridges of Konigsberg problem by devising a topology consisting of nodes and relationships to answer the then-famous question, “Is it possible to trace a walk through the city that crosses every bridge just once?” Graph theory has since found many uses, but only recently has it been applied to storing and managing data. It turns out that graphs are an intuitive way to represent relationships between data.

Graph databases let us represent related data as it inherently exists: as a set of objects connected by a set of relationships, each with its own set of descriptive properties. Graph databases differ from other databases in the sense that the data is *the structure*. The data representation also differs from other types of databases. This is similar to the way that we think about complex systems. Social networks, such as Facebook and LinkedIn, use graph databases to store and represent their users and their relationships. Common examples of graph databases include neo4j, FlockDB, AllegroGraph, and GraphDB. In addition, GraphLab offers a graph API to enable machine learning. Titan is another scalable graph database that is now integrated into Apache Falcon and provides connections into data stores in Cassandra, HBASE, Oracle, and so on.

8.2.5 Hadoop and the Hadoop Distributed File System

Distributed data stores, such as Apache HBase and Hive, rely on the Hadoop Distributed File System (HDFS). Hadoop is a scalable, fault-tolerant, distributed system for data storage and processing. Hadoop was designed based on the Google File System (GFS). It was designed to work with commodity servers and ensure that data is not lost because of hardware component failures. A Hadoop cluster consists of a master node called the *Name Node* and multiple child or data nodes. These data nodes are separate physical servers with dedicated storage (similar to a PC hard disk drive) rather than common shared storage. The core Hadoop has the following main components:

- HDFS: The Hadoop Distributed File System is used to store both structured and unstructured data. Hadoop splits the files into multiple parts (the default block size is 64 MB) and spreads them across data nodes. By default, HDFS stores three copies of the file in different data nodes.
- MapReduce: Data stored in HDFS is processed using MapReduce programs that are written primarily in Java. The term *MapReduce* refers to two separate and distinct tasks that Hadoop programs perform. The first is the map job, which takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (Key-Value pairs). The reduce job takes the output from a map as input and combines the data tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce job is always performed after the map job.

Databases, such as HBase (based on Google Bigtable) and Hive, rely on Hadoop HDFS for data storage. Hadoop and HDFS-based data stores are still suitable for batch-oriented workloads.

8.2.6 Choosing the right distributed data store

With such a wide of variety of relational and NoSQL databases, choosing the right database to use can be confusing. Most financial applications that need guaranteed committing of data can use traditional RDBMS, such as Oracle or DB2. For web-based applications, such as search and social media, which require global distributed databases, the read/write characteristics are critical.

Table 8-1 shows various NoSQL data stores and their characteristics.

Table 8-1 NoSQL data stores and characteristics

System	Primary index	Secondary index	Transaction support	Joins	Integrity constraints	Views	Type
RDBMS	Yes	Yes	Yes	Yes	Yes	Yes	Tables
memcached	Yes	Yes	No	No	No	No	Key-Value
Hadoop MapReduce	No	Yes	No	Yes	No	No	Key-Value
CouchDB	Yes	Yes	Yes	MR	No	No	Document
Hbase	Yes	Yes	Yes	MR		No	Column-Family
MongoDB	Yes	Yes	Yes	No	No	No	Document
Amazon DynamoDB	Yes	No	No	No	No	No	Column-Family
Hive	No	No	No	Yes	Yes	Yes	Tables
Cassandra	Yes	Yes	Yes	No	Yes	No	Column-Family
Voldemort	Yes	No	No	No	No	No	Key-Value
Riak	Yes	Yes	Yes	MR	No	No	Key-Value
Google Dremel	No	No	No		Yes	Yes	Tables
Google Spanner	Yes	Yes	Yes		No	Yes	Tables
Accumulo	Yes	Yes	Yes	MR	No	No	Column-Family
Impala	No	No	No	Yes	No	Yes	Tables

9

Big data and analytics implications for the data center

As the technological world moves from predominantly systems of record to a coexistence with systems of engagement, the nature of workloads in the data center changes radically. This change has several implications for the data center, as outlined in Table 9-1.

Table 9-1 Big data and analytics implications for the data center

Characteristic	Systems of record	Systems of engagement
Workloads	► Core data and transactions ► Operational analytics	► Social, mobile, and user interface ► Big data analytics
Scaling	► Heterogeneous scaling for most workloads ► Frequent scaling up	► Homogeneous scaling ► Typically horizontal scale-out
Resources	Resources are pooled and dedicated.	Shared resources use commodity hardware.
Applications	► Monolithic applications ► Waterfall methodology	► Composition of Services model ► Develops deployment
Infrastructure attributes	► Infrastructure to support transactional integrity (which can affect scale beyond a certain point) ► Extreme focus on quality of service (QoS) and service level agreement (SLA) governance ► Infrastructure requires continuous scalability ► Capacity is preallocated	► Recovery-oriented computing to build Internet-scale services ► Increasing focus on QoS, including SLA governance ► Systems designed for eventual consistency ► Capacity is allocated on request
Infrastructure efficiency	Focus on consolidation, standardization, simplification, and automation	

9.1 The hybrid data center

We can foresee a hybrid data center that supports both systems of record and systems of engagement. This hybrid data center can be achieved by using the key characteristics of cloud computing: automation, elasticity, standardization, and consolidation. This data center might have resources that are pooled and allocated only on demand, and the ability to support core data transactions and big data systems of eventual consistency.

Table 9-2 shows the critical characteristics of a hybrid data center.

Table 9-2 Hybrid data center characteristics

Component	**Characteristic**
Workload type	► Systems of record ► Systems of engagement
Workload diversity	► Broad workloads consolidated on service-optimized platforms ► Hybrid of dedicated and shared resources
Application environment	Composed through web standards
Platform topology	► Increasingly centralized control ► Emergence of a converged infrastructure
Platform dedication and access level	► Shared with ubiquitous mobile access ► Self-served as a *service*
Deployment model	► Hybrid of on-premises and off-premises
Management structure	► Self-provision portal ► High levels of automation ► Dynamic resource management
Networking	Converged network fabric
Security	► Protect the data ► Trusted platform
Platform architecture	Consolidation of architectures with Lintel-Wintel orientation
Systems definition	Increasing emphasis on software-defined environments (networking, storage, and servers)

9.2 Big data and analytics service ecosystem on cloud

A service provider can offer big data and analytics capabilities as a service ecosystem on a cloud, with organizations selecting their desired capabilities from a service catalog.

This big data and analytics capability service adoption model is suitable for most organizations, whether they are new to big data or have been working with it for years. The model can be applied whether an organization needs only a few big data capabilities or many big data capabilities, whether or not it has in-house expertise, and even whether it wants to outsource the big data and analytics capability implementation.

Most organizations draft their big data goals first, prioritize them, and then implement the capabilities in a phased manner. The following list shows the most common big data capabilities that can be provided as services in a cloud model:

- Extract, transform, and load (ETL) as a service
- Visualization and search as a service
- Analytics as a service
- Hadoop NoSQL as a service (analytics infrastructure)
- Data warehousing as a service
- Reporting as a service
- Entity analytics as a service
- Policy and data governance as a service
- Database as a service

In this kind of service hosting model, the service provider provides the service so that the subscribing organizations can focus on their overall business and IT goals, for example:

- When the government intelligence agencies have an ad hoc need to perform social media analytics to assess and identify threats during national political events, they can subscribe to analytics as a service.
- Financial organizations that need to generate quarterly and yearly reports can subscribe to reporting as a service to accomplish those goals without having to invest in technologies that will only be used occasionally.
- Cyber security and crime-fighting agencies that must collaborate in their work can avoid the need for multiple software and hardware infrastructures by subscribing to information sharing as a service, where they can each have a common portfolio of services and systems to provide consistent results.

For organizations that are just entering the big data space, the cloud-based service consumption model provides an easy way for them to explore their options and skills while building business cases to support their use.

Most public cloud services today focus on infrastructure as a service (IaaS). Numerous vendors provide different components of the big data platform, and certain software vendors provide specific analytical capabilities as a service using public clouds. For example, companies, such as Tableau, provide visualization as a service on both the Amazon Cloud as well as the IBM SoftLayer.

Cloud platforms, such as IBM Cloud Foundry or Heroku, which provide platform as a service (PaaS), are suitable for big data services. IBM Bluemix, which is built on the Cloud Foundry, includes a few of the services listed above.

Figure 9-1 shows the various big data and analytics capabilities that can be delivered and consumed using the cloud.

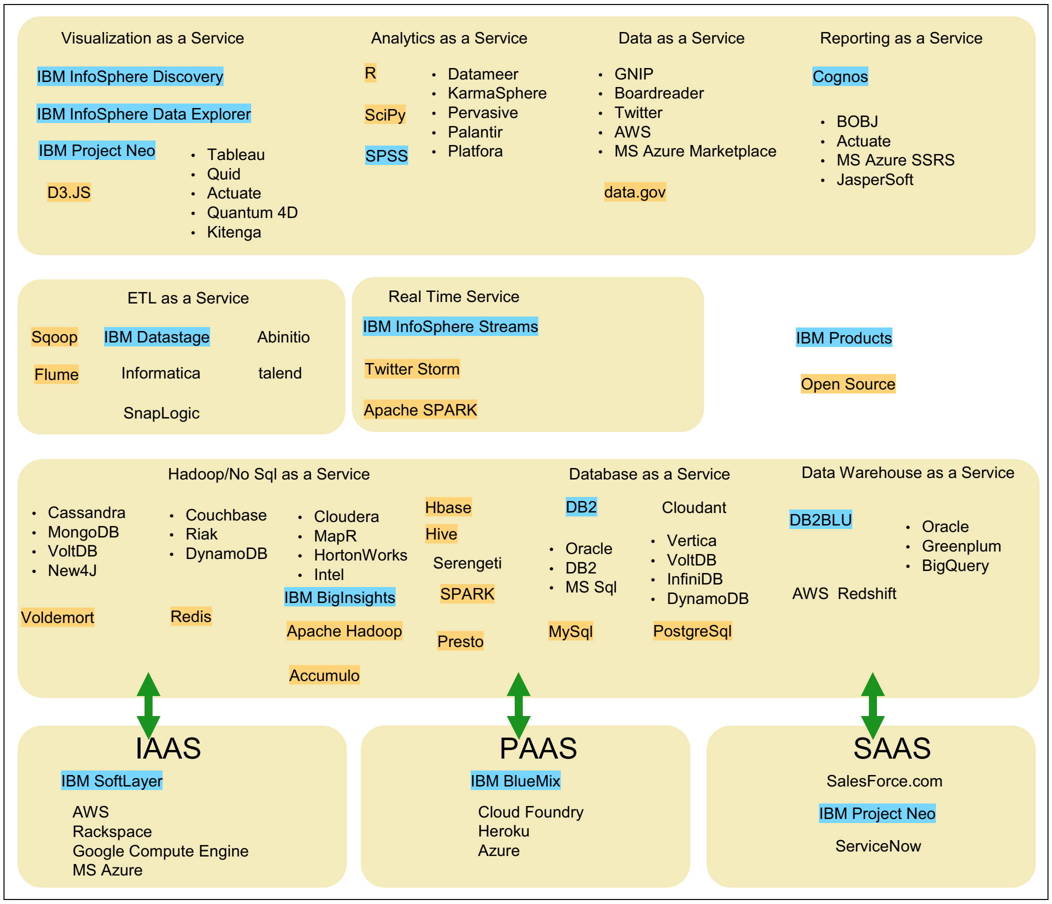


Figure 9-1 Big data and analytics services enabled on the cloud

Most public cloud services currently focus on IaaS. The providers offer a multi-tenant infrastructure. Private cloud services are built for use for a single company and can be housed either within a company’s data center (on-premises) or off-site in a third-party data center. Hybrid clouds allow for combining aspects of public and private clouds, where companies can move their workloads between the two models based on their governance model.

The most commonly known public cloud providers include Amazon Web Services (AWS), Microsoft Azure, Rackspace, IBM SoftLayer, Google Compute Engine, HP Public Cloud, and so on. As companies start looking at public clouds, they need to consider the issues of vendor lock-in and security. To prevent vendor lock-in, consider cloud services based on OpenStack. Rackspace, IBM, and HP cloud offerings are based on the OpenStack API.

Companies also need to consider the total cost of ownership (TCO) when they consider moving major workloads into a public cloud environment. The cost of moving data back and forth between your company’s internal systems and a third-party cloud can add up quickly. Most public cloud vendors provide virtual machine instances. The performance of these virtual machine instances might not be suitable for enterprise application requirements.

When evaluating cloud vendors, look for cloud vendors with *bare-metal server* options. The cost of transferring data between data centers can also add up quickly. This is especially true for most big data applications. When deploying one of the patterns described in the previous sections of this document, be sure to consider the amount of data that might have to be transferred.

9.3 IBM SoftLayer for big data and analytics

With its SoftLayer cloud, IBM provides unique capabilities that enable companies to quickly deploy a public big data and analytics solution. The following key aspects of SoftLayer clouds differentiate them from other public cloud offerings:

- **Large, fast networks**

  Companies offering cloud capabilities need significant bandwidth. A couple of servers in a single data center are not sufficient. The SoftLayer network and capacity are considerable. Thirteen separate data centers house about 200,000 servers at locations around the world (including the United States, Singapore, the United Kingdom, and the Netherlands), each connected to the others with dedicated 20 Gbps private network fiber links. SoftLayer has a triple network architecture (Public, Management, and Private).

- **Virtual servers for convenience, bare metal for performance**

  A *virtual server* is a virtual machine running on a hypervisor (an operating system) and hosted on a physical server. The same physical server can host multiple virtual servers. So, one server running a Linux operating system can host additional virtual servers running Microsoft Windows or Linux. SoftLayer offers virtual servers that can be set up in minutes and scaled as needed.

  Virtual servers also come with a performance penalty, however. The virtual machine abstracts the physical resources of the server, such as its RAM and CPU, creating an overhead that makes it inappropriate for some high-performance workloads. This is where *bare metal* (a term used to describe a non-virtualized dedicated server) is a better option. SoftLayer can create bare-metal servers for you, where you design the hardware to match your requirements and then manage the server from the cloud. This approach delivers the advantages of centralized cloud management but with no virtualization performance penalty. Newer technology, such as Docker, which is based on Linux containers, offers options to run applications directly on top of bare metal as well as on virtual servers. With Docker, applications are self-contained and can be rapidly provisioned and moved to other Docker hosts. SoftLayer and other public clouds, such as Google and AWS, provide support for Docker.

- **Completely customizable**

  The SoftLayer infrastructure can be customized to meet your needs, from altering the network speed and bandwidth to choosing a specific operating system to imposing the most appropriate security option with SecurityLayer Services. In addition, you can manage SoftLayer with API access to more than 2,000 command and control functions. One of the key technologies used for big data is Hadoop. Hadoop clusters are designed to be rack aware. So, while you configure the cluster, you must map the data nodes to the racks.

  In a typical public cloud environment, you do not have control of the infrastructure to build a cluster. SoftLayer provides API access to each node and rack, so you can deploy a fully rack-aware Hadoop cluster. Furthermore, SoftLayer provides Rapid provisioning (a Hadoop cluster with 1,000 data nodes can be provisioned in 10 minutes) and accelerated analytics (a 10 TB sort has been completed in 15 minutes, which beat the previous mark that was set in 2009 and used six times the hardware).

Several other SoftLayer features specifically support Hadoop:

  - Deeply integrated and tuned analytics stack with best-of-breed cluster management
  - Hadoop rack-awareness
  - Exchange compression and anti-colocation of reduce allotments
  - Increased Data Input phase redundancy
  - HDFS buffer size tuning
  - Tune Mapper and Reducer counts
  - Distinct Mapper and Reducer Java Message Service (JVM) management (reducers need more memory)
  - Modifiable exchange sort implementation

- **Data center management from anywhere**

  The SoftLayer mobile app (available for iOS, Android, and Windows Phone) helps you monitor your infrastructure. You can view server performance details, monitor bandwidth usage, initiate support requests, and even turn servers on and off.

10

Big data and analytics on the cloud: Architectural factors and guiding principles

This chapter discusses critical architectural factors for deploying big data and analytics on the cloud and offers guidance for making the best decisions.

10.1 Architectural principles

First, it is worth highlighting several broad big data architectural principles that can guide your thinking and help steer you to more granular decisions later.

10.1.1 Security

The Security principle states that *all data, whether in motion or stationary, will be secured, ensuring traceability, auditability, and governance*.

You must design for security. Too often, security is added as an afterthought retrofit, when it is much harder and more expensive to implement. The overarching objective of this principle is data security and accountability.

10.1.2 Speed

The Speed principle states that *the timely delivery of data to the user is the primary design consideration. Just in time, versus as fast as possible, is the focus*.

Time is money. That is the mantra of the business world. Therefore, if you ask a customer how fast something needs to be, the answer will be, as fast as possible. This principle reminds you that context is what is truly important.

The overarching objective of this principle is speed for business benefit. Ensuring faster delivery comes with a cost. It is important to gauge whether that additional cost provides the expected return on the investment.

10.1.3 Scalability

The Scalability principle states that *compute and storage capacity must be available on demand, in keeping with the volume of data and business needs. Horizontal scaling (adding more nodes to a cluster) versus vertical scaling (adding more resource to a single node) must be the prime focus*.

A common mistaken perception is that cloud provides inexhaustible resources. But while they are not inexhaustible, cloud resources are easily and rapidly scaled to meet growing demand. The challenge for the architect, therefore, is to articulate the scalability benefits of cloud computing when compared to earlier approaches, and to balance the needed capabilities with the client’s IT budget.

An IBM developerWorks article, *Architecting applications for the cloud*, shows how to create applications for the cloud that can scale based on demand:

http://www.ibm.com/developerworks/cloud/library/cl-cloudappdevelop/

Massive Parallel Processing (MPP) environments are another example of horizontal scaling implementations. The objective is to provide capacity on demand, never too much, which translates into wasted expense, and never too little, which translates into wasted opportunity.

An exception to this advice is the IBM z/VM® platform. Cloud services on z/VM are becoming attractive to more companies based on the relative cost of providing services on distributed platforms versus IBM z/OS® platforms. Does it matter which platform is used to deliver a service? If the quality of service (QoS) commitments and service level agreements (SLAs) are met, the user of the service does not care. The driving force is price, not platform. Therefore, if z/VM is an option, look closely at vertical scaling. The scalability principle is still relevant to z/VM.

10.1.4 Reliability and Availability

The Reliability and Availability principle states that *data that is critical to the business will be reliably captured, delivered, and maintained. By definition, this precludes "all data," but defines a scope that will be consistent with business needs.*

Every architect will tell you that when they ask a customer which components of a solution they want to make highly available, they customer will respond "everything." The objective is always to make the solution as available as it needs to be, but this principle exists to remind you that not everything needs to be highly available.

The objective is reliable data protection consistent with business need. It is important to note that this principle is stated in the context of the data, not the system. The key element of *any* big data solution is the data.

10.2 Architectural decisions

With the guiding principles established, the next step is to make decisions.

10.2.1 Performance

Performance can be discussed from different perspectives. The big data perspective can be framed by these questions:

- Performance is measured from the perspective of the user experience. So, who is the user of your solution?
- Where must the solution components be placed to fulfill the performance requirements?
- Which components best serve the performance requirements?

Assumptions

Start with these assumptions:

- Deploying a Hadoop cluster on physical servers provides better performance than on virtual servers or in a cloud.
- If all of the components of the solution are in the same environment, whether it is a private cloud or a public cloud, the same performance characteristics will be observed. Therefore, this decision is only relevant to a hybrid solution.

Considerations

Remember these factors:

- Don’t lose sight of the performance cost of moving the data. Latency can be introduced by moving the data from one location to another location.
- Requirements for random access versus sequential access will affect your technology choices (Hadoop, HBase, or other data stores).
- Different requirements exist for read-heavy versus write-heavy operations. This will affect technology choices.
- The volume data directly affects performance. The more data that needs to be moved, the greater impact it will have on the performance of the solution.
- Performance is also affected by the choice between virtual and physical. Analytic components will run faster on physical hardware than when the same actions are virtualized. Several factors are related:
  - In-memory databases improve performance but reduce scale:
    - Memory capacity limits the amount of data that can be loaded.
    - The amount of memory that can be physically held in the server is limited.
    - The performance-scale balance can be addressed by the compromise between in-memory and traditional databases in a hybrid setup.

Recommendations

With the stated assumptions and considerations, ensure that you perform these tasks:

- Understand the characteristics of the workload and choose the appropriate technology to support it.
- Determine whether the performance required of the most demanding workloads can be met on virtual components, and if not, use physical ones.
- Less demanding workloads are usually safe to run on virtual components, but this must be validated against the requirements.

10.2.2 Scalability

The topic of scalability is framed with these questions:

- Scale out or scale up?
- Proprietary or open source?
- What is the best way to provide scalability for the solution?

Considerations

Keep these factors in mind:

- When choosing a component, appliance, or solution, think about extensibility. How will it scale out or up?
- How likely is the need to scale beyond the originally sized environment?
- Scaling out and scaling up present different implementation and architectural challenges that must be considered carefully.
- Consider using technology where scaling can be done independently for compute, storage, or network needs. For example, certain big data appliances force you to buy an additional appliance just to add more storage.

Recommendation

For big data solutions, try to scale out (horizontally) instead of scaling up (vertically). z/OS-based solutions are exceptions to this rule. Cloud platforms provide an ideal environment for horizontal scaling because of their inherent elastic characteristics (remember the illusion of being an inexhaustible resource). This makes the cloud a good platform to place a component with this requirement.

10.2.3 Financial

We all want to design the best solution with all the features, but the reality is that such solutions can be prohibitively expensive. So financial considerations are always important, and the factors of performance, reliability, and availability are almost always at odds with cost.

Considerations

Keep these concepts and questions in mind:

- Virtual is less expensive than physical.
- What is the cost of data transmission?
- Open or proprietary solutions?
- What is the cost of a public cloud versus a private cloud?
- Frequency of use (if a service is infrequently used, it can be ideal for placement with a public cloud service on pay-as-you-go terms).

Recommendation

Build a solution that is *fit for purpose* at the most reasonable price, based on business need.

A good exercise is drawing the component and logical operational models of the solution so that you can clearly see the capabilities that are being delivered with each piece of the solution. Important factors include the interaction and message flows between the components, the locations, the nodes, and the deployment possibilities. These models are invaluable tools in your architect toolkit.

10.2.4 Reliability

The key component of a big data solution is the data. Therefore, a critical question that must be answered is “How reliable does the data delivery need to be?”

Considerations:

- Reliability is concerned with the guaranteed delivery of data from source to landing zone to discovery layer to application layer.
- Although retry might seem like an obvious requirement, consider that a *retry* capability might be adequate to meet the business need.

Considerations

Keep these factors in mind:

- Is there a business need to ensure reliable, guaranteed data delivery from the source to the landing zone?
- Is there a business need to ensure reliable and guaranteed data delivery from the landing zone to the discovery layer?
- Is there a business need to ensure reliable and guaranteed data delivery from the discovery layer to the application layer?

Recommendation

Probe the business requirements and deliver reliability where it meets a specific business need. Where guaranteed reliability is not necessary, a capable retry mechanism might be good enough, and it will certainly be less expensive.

10.2.5 Backup and recovery

> **Important:** Every enterprise solution needs data backup and recovery capabilities.

Considerations

Keep these factors in mind:

- Depending on the big data solution used, ensure that the correct backup and restore systems and procedures are in place.
- Is there a business need for defining recovery point objective (RPO) and recovery time objective (RTO)?
- What is the impact of data in the source layer being lost or corrupted?
- What is the impact of data in the landing zone being lost or corrupted?
- What is the impact of data in the discovery layer being lost or corrupted?
- What is the impact of data in the application layer being lost or corrupted?

Recommendation

Carefully consider these points and provide a backup and recovery solution that protects the data, meets the RPO and RTO, and ensures business continuity.

10.2.6 Locations and placement

Where do you place big data solution components?

Recommendation

Consider several of the relevant architectural decisions and place the components to meet the needs in those areas:

- Data sensitivity
- Performance
- Scalability
- Financial
- Availability
- Backup and recovery
- Disaster recovery

The decisions in these areas will each provide guidance for your placement choices and constraints. If the decision is a choice between on-premises and off-premises cloud locations for a component, the guiding factor is usually cost.

10.2.7 Sensitive data

The sensitivity of data is an increasingly important consideration in solution architecture and can dictate where a particular solution component or data set must be placed. Organizations must comply with important regulatory standards, such as personally identifiable information (PII), Health Insurance Portability and Accountability Act (HIPAA), and Sarbanes-Oxley Act (SOX). Safe Harbor-related laws also need to be considered.

Recommendation

Keep all sensitive data on-premises or in a private cloud or data center, that is, within the network boundaries of the company. You still can use off-premises big data services for other components of the solution. An interesting point to think about is that government intrusion is less likely for on-premises or private cloud data.

10.2.8 Disaster recovery

Any good plan ensures that the solution remains available in a disaster. This problem does not go away because you are considering the use of cloud technologies.

Considerations

Keep these factors in mind:

- Is there a business need for a disaster recovery solution?
- Is reduced performance acceptable after a disaster? What are the minimum application, data, and performance requirements defined within the business continuity plan (BCP)?
- After a disaster, how quickly does the business need to recover? What is the RTO?
- Is data loss acceptable after a disaster? What is the RPO?

Recommendation

Design a disaster recovery solution that meets the BCP. Reduced performance is often acceptable for an agreed-to period after a disaster, and this needs to be defined in the BCP.

10.2.9 Availability

Determine whether components must be made highly available (HA). Do not assume that HA is necessary for everything. Omnipresent HA is expensive.

Considerations

Keep these factors in mind:

- What is the impact of components in the source layer being unavailable?
- What is the impact of components in the landing zone being unavailable?
- What is the impact of components in the discovery layer being unavailable?
- What is the impact of the components in the application layer being unavailable?
- What is the impact of integration between the components in the application layer being unavailable?

Recommendation

Add resilience to components where the impact of their unavailability will adversely affect the service being delivered to the business.

Related publications

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

Other publications

Here are some additional publications related to the topic of big data analytics in the cloud:

- *IBM System x Reference Architecture for Hadoop*:

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

- *IBM Smart Cloud - Building a Cloud-Enabled Data Center*:

  http://www.redbooks.ibm.com/abstracts/redp4893.html

Online resources

Numerous resources are available from IBM and on the Internet to help you get started with big data and learn about deploying big data solutions on a cloud. Here are some links to help you begin your journey:

- Join the Big Data University to learn about Hadoop and other big data technologies. Many courses are free:

  http://www.bigdatauniversity.com/

- Download a no-charge BigInsights Quickstart VM to learn more about Hadoop, HBase, Hive, BigSQL, and so on:

  http://www-01.ibm.com/software/data/infosphere/biginsights/quick-start/

- Read analyst reports and papers from Forrester, Gartner, and IBM Institute for Business Value.
- Learn about architecting cloud solutions using the IBM Cloud Computing Reference Architecture 3.0:

  https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Wf3cce8ff09b3_49d2_8ee7_4e49c1ef5d22/page/IBM%20Cloud%20Computing%20Reference%20Architecture%203.0

- Contact your IBM representative and schedule a no-charge big data workshop to discuss leading practices and business value for your organization.
- Attend a Developer Day to get hands-on experience with big data technologies. Contact your IBM representative for dates and locations.
- Register with the Big Data Reference Architecture Community to access Big Data Reference Architecture updates.
- View big data reference architecture videos.
- Look for opportunities to use the logical reference architectures with your clients and provide feedback in the community.

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Building Big Data and Analytics Solutions in the Cloud

Redpaper™

Characteristics of big data and key technical challenges in taking advantage of it

Impact of big data on cloud computing and implications on data centers

Implementation patterns that solve the most common big data use cases

This IBM Redpaper publication is aimed at chief architects, line-of-business executives, and CIOs to provide an understanding of the cloud-related challenges they face and give prescriptive guidance for how to realize the benefits of big data solutions quickly and cost-effectively.

The paper covers these topics:

- The characteristics of big data
- The business drivers and key technical challenges in exploiting big data
- The impact of big data on cloud computing environments
- Functional and infrastructure architecture considerations for big data and data analytics in the cloud
- Implementation patterns to solve the most common big data use cases
- The implications of big data on data centers
- A roundup of available NoSQL technologies

In addition, this paper introduces a taxonomy-based tool that can quickly help business owners understand the type of big data problem at hand.

This paper is based on the knowledge IBM has gained in both cloud computing and big data, and builds upon the popular Cloud Computing Reference Architecture (CCRA) and its principles. The objective is not to provide an exhaustive list of every capability required for every big data solution hosted in a cloud environment. Rather, the paper aims to give insight into the most common use cases encountered across multiple industries and the common implementation patterns that can make those use cases succeed.

REDP-5085-00