1. Intoduction to Hadoop

Hadoop is a rapidly evolving ecosystem of components for implementing the Google MapReduce algorithms in a scalable fashion on commodity hardware. Hadoop enables users to store and process large volumes of data and analyze it in ways not previously possible with less scalable solutions or standard SQL-based approaches.

As an evolving technology solution, Hadoop design considerations are new to most users and not common knowledge.

Hadoop is a highly scalable compute and storage platform. While most users will not initially deploy servers numbered in the hundreds or thousands, It recommends following the design principles that drive large, hyper-scale deployments. This ensures that as you start with a small Hadoop environment, you can easily scale that environment without rework to existing servers, software, deployment strategies, and network connectivity.

1.1 Hadoop Uses

Hadoop brings the ability to cheaply process large amounts of data, regardless of its structure. By large, we mean from 10-100 gigabytes and above. How is this different from what went before?
Existing enterprise data warehouses and relational databases excel at processing structured data and can store massive amounts of data, though at a cost: This requirement for structure restricts the kinds of data that can be processed, and it imposes an inertia that makes data warehouses unsuited for agile exploration of massive heterogenous data. The amount of effort required to warehouse data often means that valuable data sources in organizations are never mined. This is where Hadoop can make a big difference.

Hadoop was originally developed to be an open implementation of Google MapReduce and Google File System. As the ecosystem around Hadoop has matured, a variety of tools have been developed to streamline data access, data management, security, and specialized additions for verticals and industries. Despite this large ecosystem, there are several primary uses and workloads for Hadoop that can be outlined as:

- **Compute** –A common use of Hadoop is as a distributed compute platform for analyzing or processing large amounts of data. The compute use is characterized by the need for large numbers of CPUs and large amounts of memory to store in-process data. The Hadoop ecosystem provides the application programming interfaces (APIs) necessary to distribute and track workloads as they are run on large numbers of individual machines.

- **Storage** –One primary component of the Hadoop ecosystem is HDFS—the Hadoop Distributed File System. The HDFS allows users to have a single addressable namespace, spread across many hundreds or thousands of servers, creating a single large file system. HDFS manages the replication of the data on this file system to ensure hardware failures do not lead to data loss. Many users will use this scalable file system as a place to store large amounts of data that is then accessed within jobs run in Hadoop or by external systems.

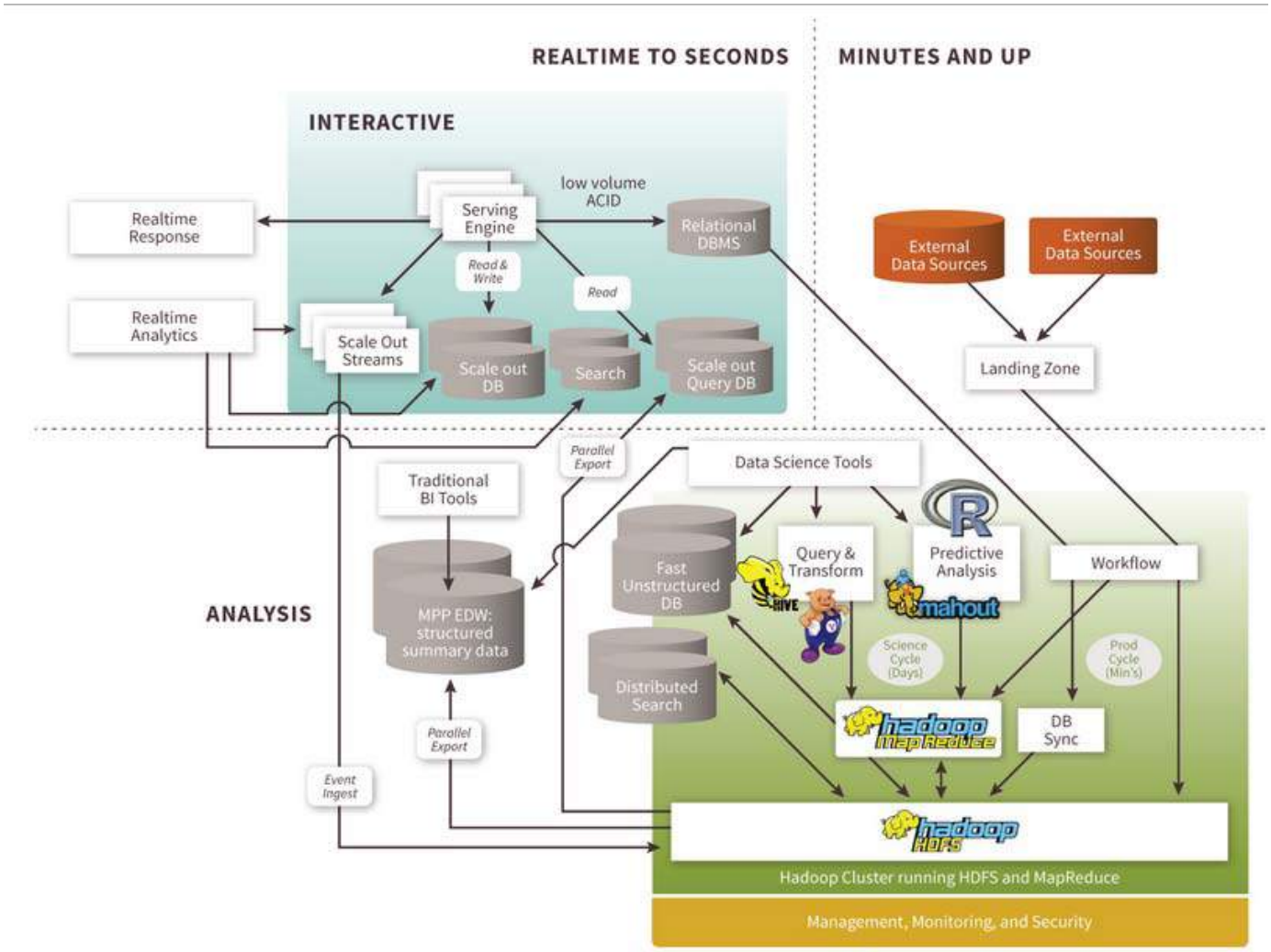
- **Database** –The Hadoop ecosystem contains components that allow the data within the HDFS to be presented in a SQL-like interface. This allows standard tools to INSERT, SELECT, and UPDATE data within the Hadoop environment, with minimal code changes to existing applications. Users will commonly employ this method for presenting data in a SQL format for easy integration with existing systems and streamlined access by users.

2.1.1 What is Hadoop good for ?

When the original MapReduce algorithms were released, and Hadoop was subsequently developed around them, these tools were designed for specific uses. The original use was for managing large data sets that needed to be easily searched. As time has progressed and as the Hadoop ecosystem has evolved, several other specific uses have emerged for Hadoop as a powerful solution.

- **Large Data Sets** –MapReduce paired with HDFS is a successful solution for storing large volumes of unstructured data.

- **Scalable Algorithms** –Any algorithm that can scale to many cores with minimal inter-process communication will be able to exploit the distributed processing capability of Hadoop.



- **Log Management** –Hadoop is commonly used for storage and analysis of large sets of logs from diverse locations. Because of the distributed nature and scalability of Hadoop, it creates a solid platform for managing, manipulating, and analyzing diverse logs from a variety of sources within an organization.

- **Extract-Transform-Load (ETL) Platform** –Many companies today have a variety of data warehouse and diverse relational database management system (RDBMS) platforms in their IT environments. Keeping data up to date and synchronized between these separate platforms can be a struggle. Hadoop enables a single central location for data to be fed into, then processed by ETL-type jobs and used to update other, separate data warehouse environments.

2.1.2 Not so much ?

As with all applications, some actions are not optimal for Hadoop. Because of the Hadoop architecture, some actions will have less improvement than others as the environment is scaled up.

- Small File Archive –Because of the Hadoop architecture, it struggles to keep up with a single file system name space if large numbers of small objects and files are being created in the HDFS. Slowness for these operations will occur from two places, most notably the single NameNode getting overwhelmed with large numbers of small I/O requests and the network working to keep up with the large numbers of small packets being sent across the network and processed.
- High Availability –A single NameNode becomes a single point of failure and should be planned for in the uptime requirements of the file system. Hadoop utilizes a single NameNode in its default configuration. While a second, passive NameNode can be configured, this must be accounted for in the solution design.

2.2 Hadoop Architecture and Components

2.2.1 Hadoop design

Figure 2 depicts the representation of the Hadoop ecosystem. This model does not include the applications and end-user presentation components, but does enable those to be built in a standard way and scaled as your needs grow and your Hadoop environment is expanded.

The representation is broken down into the Hadoop use cases from above: Compute, Storage, and Database workloads. Each workload has specific characteristics for operations, deployment, architecture, and management. The solutions are designed to optimize for these workloads and enable you to better understand how and where Hadoop is best deployed. Apache Hadoop, at its core, consists of 2 sub-projects – Hadoop MapReduce and Hadoop Distributed File System. Hadoop MapReduce is a programming model and software framework for writing applications that rapidly process vast amounts of data in parallel on large clusters of compute nodes. HDFS is the primary storage system used by Hadoop applications. HDFS creates multiple replicas of data blocks and distributes them on compute nodes throughout a cluster to enable reliable, extremely rapid computations. Other Hadoop-related projects at Apache include Chukwa, Hive, HBase, Mahout, Sqoop and ZooKeeper.

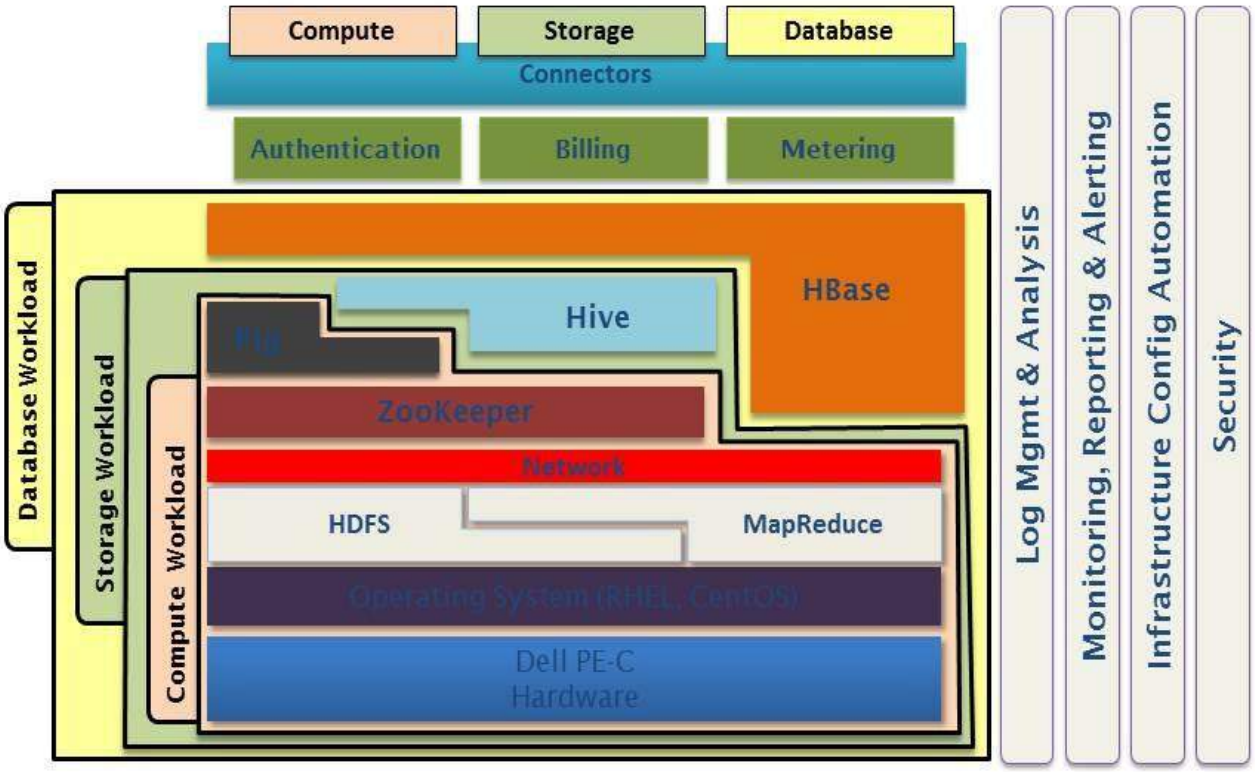


Figure 1. The Representation of the Hadoop ecosystem.

Apache Hadoop has been the driving force behind the growth of the big data industry. You'll hear it mentioned often, along with associated technologies such as Hive and Pig. But what does it do, and why do you need all its strangely-named friends, such as Oozie, Zookeeper and Flume?

Hadoop brings the ability to cheaply process large amounts of data, regardless of its structure. By large, we mean from 10-100 gigabytes and above. How is this different from what went before?

Existing enterprise data warehouses and relational databases excel at processing structured data and can store massive amounts of data, though at a cost: This requirement for structure restricts the kinds of data that can be processed, and it imposes an inertia that makes data warehouses unsuited for agile exploration of massive heterogenous data. The amount of effort required to warehouse data often means that valuable data sources in organizations are never mined. This is where Hadoop can make a big difference.

The core of Hadoop: MapReduce

Created at Google n response to the problem of creating web search indexes, the MapReduce framework is the powerhouse behind most of today's big data processing. In addition to Hadoop, you'll find MapReduce inside MPP and NoSQL databases, such as Vertica or MongoDB.
The important innovation of MapReduce is the ability to take a query over a dataset, divide it, and run it in parallel over multiple nodes. Distributing the computation solves the issue of data too large to fit onto a single

machine. Combine this technique with commodity Linux servers and you have a cost-effective alternative to massive computing arrays.

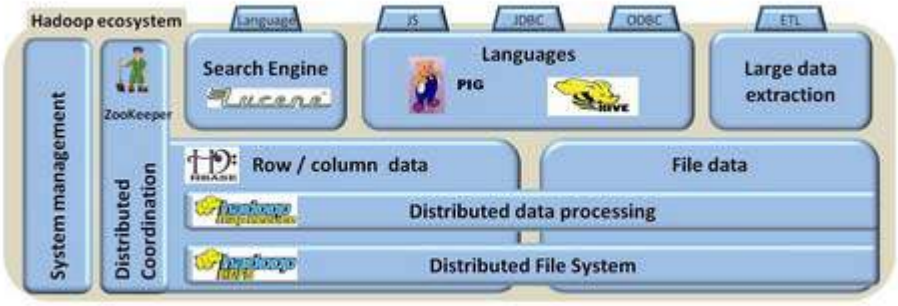
At its core, Hadoop is an open source MapReduce implementation. Funded by Yahoo, it emerged in 2006 and,according to its creator Doug Cutting, reached “web scale” capability in early 2008.

Hadoop’s lower levels: HDFS and MapReduce

Above, we discussed the ability of MapReduce to distribute computation over multiple servers. For that computation to take place, each server must have access to the data. This is the role of HDFS, the Hadoop Distributed File System.

HDFS and MapReduce are robust. Servers in a Hadoop cluster can fail and not abort the computation process. HDFS ensures data is replicated with redundancy across the cluster. On completion of a calculation, a node will write its results back into HDFS.
There are no restrictions on the data that HDFS stores. Data may be unstructured and schemaless. By contrast, relational databases require that data be structured and schemas be defined before storing the data. With HDFS, making sense of the data is the responsibility of the developer’s code.

Programming Hadoop at the MapReduce level is a case of working with the Java APIs, and manually loading data files into HDFS.



Hadoop HDFS Cluster nodes

Hadoop has a variety of node types within each Hadoop cluster; these include DataNodes, NameNodes, and EdgeNodes. Names of these nodes can vary from site to site, but the functionality is common across the sites. Hadoop’s architecture is modular, allowing individual components to be scaled up and down as the needs of the environment change. The base node types for a Hadoop HDFS CIsuter cluster are:

- **NameNode** –The NameNode is the central location for information about the file system deployed in a Hadoop environment. An environment can have one or two NameNodes, configured to provide minimal redundancy between the NameNodes. The NameNode is contacted by clients of the Hadoop Distributed File System (HDFS) to locate information within the file system and provide updates for data they have added, moved, manipulated, or deleted.
- **DataNode** –DataNodes make up the majority of the servers contained in a Hadoop environment. Common Hadoop environments will have more than one DataNode, and oftentimes they will number in the hundreds

based on capacity and performance needs. The DataNode serves two functions: It contains a portion of the data in the HDFS and it acts as a compute platform for running jobs, some of which will utilize the local data within the HDFS.

- **EdgeNode** –The EdgeNode is the access point for the external applications, tools, and users that need to utilize the Hadoop environment. The EdgeNode sits between the Hadoop cluster and the corporate network to provide access control, policy enforcement, logging, and gateway services to the Hadoop environment. A typical Hadoop environment will have a minimum of one EdgeNode and more based on performance needs.

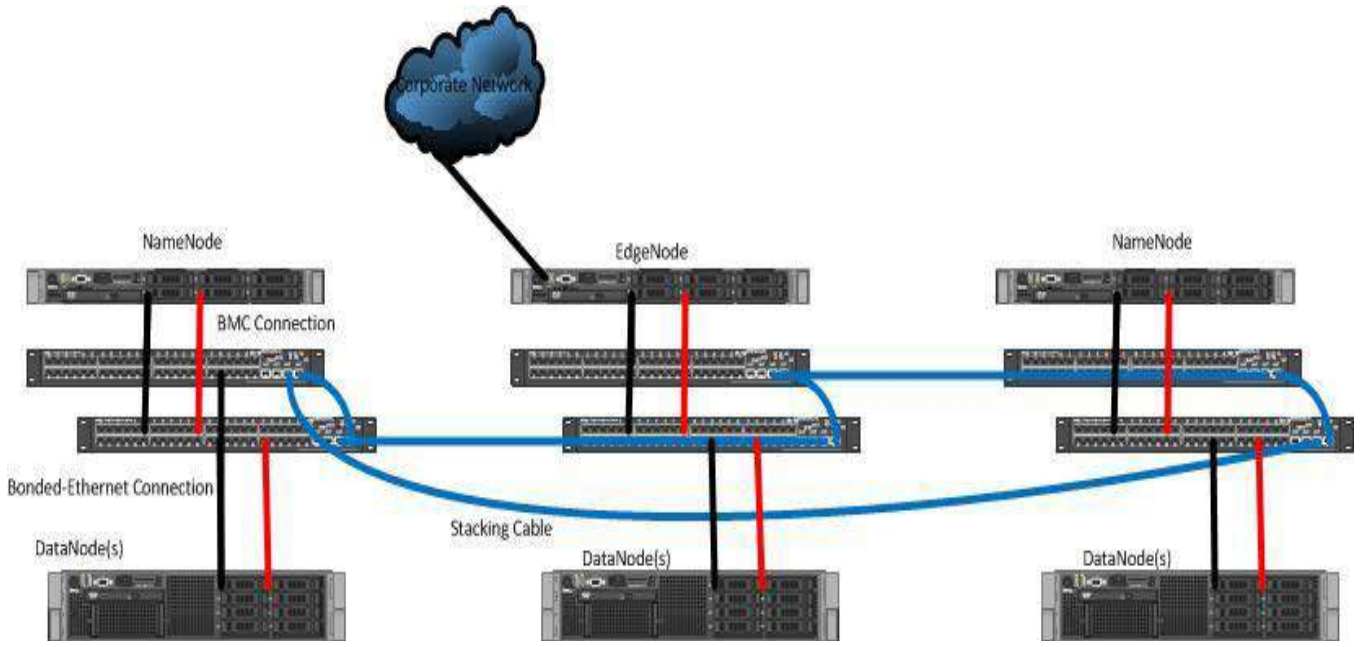


Figure 1. Node types within Hadoop clusters.

Improving programmability: Pig and Hive

Working directly with Java APIs can be tedious and error prone. It also restricts usage of Hadoop to Java programmers. Hadoop offers two solutions for making Hadoop programming easier.

Pig is a programming language that simplifies the common tasks of working with Hadoop: loading data, expressing transformations on the data, and storing the final results. Pig's built-in operations can make sense of semi-structured data, such as log files, and the language is extensible using Java to add support for custom data types and transformations.
Hive enables Hadoop to operate as a data warehouse. It superimposes structure on data in HDFS and then permits queries over the data using a familiar SQL-like syntax. As with Pig, Hive's core capabilities are extensible.

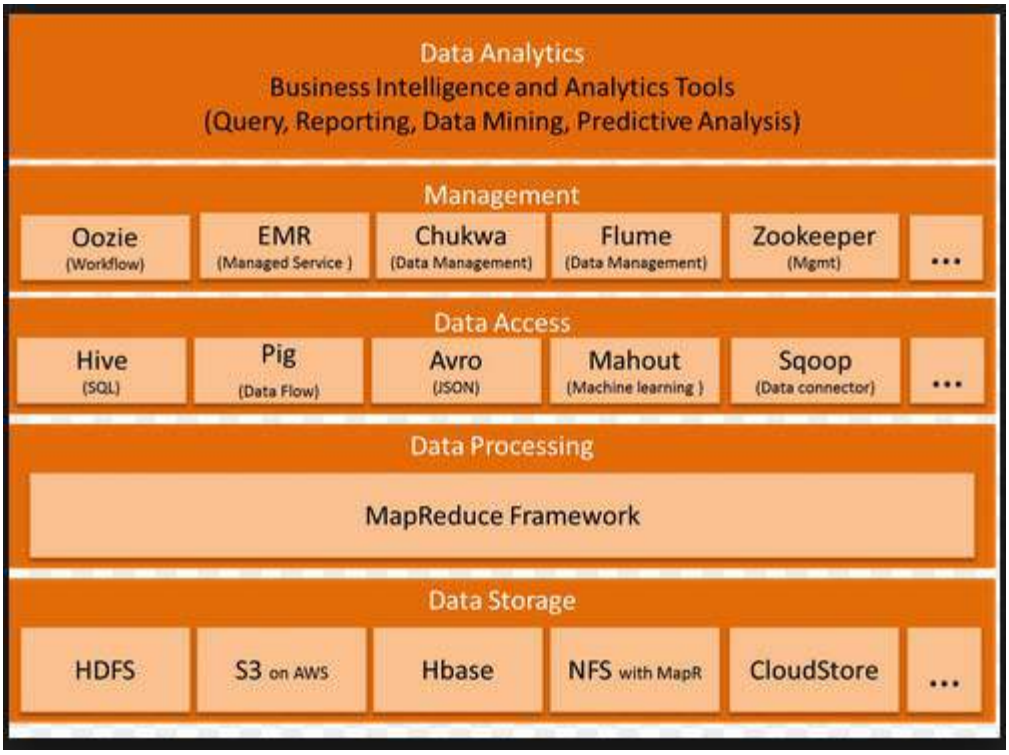
Choosing between Hive and Pig can be confusing. Hive is more suitable for data warehousing tasks, with predominantly static structure and the need for frequent analysis. Hive's closeness to SQL makes it an ideal point of integration between Hadoop and other business intelligence tools.

Pig gives the developer more agility for the exploration of large datasets, allowing the development of succinct scripts for transforming data flows for incorporation into larger applications. Pig is a thinner layer over Hadoop than Hive, and its main advantage is to drastically cut the amount of code needed compared to direct use of Hadoop's Java APIs. As such, Pig's intended audience remains primarily the software developer.

Improving data access: HBase
At its heart, Hadoop is a batch-oriented system. Data are loaded into HDFS, processed, and then retrieved. This is somewhat of a computing throwback, and often, interactive and random access to data is required.

Enter Hbase , a column-oriented database that runs on top of HDFS. Modeled after Google's BigTable, the project's goal is to host billions of rows of data for rapid access. MapReduce can use HBase as both a source and a destination for its computations, and Hive and Pig can be used in combination with HBase.
In order to grant random access to the data, HBase does impose a few restrictions: Hive performance with HBase is 4-5 times slower than with plain HDFS, and the maximum amount of data you can store in HBase is approximately a petabyte, versus HDFS' limit of over 30PB.

HBase is ill-suited to ad-hoc analytics and more appropriate for integrating big data as part of a larger application. Use cases include logging, counting and storing time-series data



The Hadoop Bestiary

Ambari	Deployment, configuration and monitoring
Flume	Collection and import of log and event data
HBase	Column-oriented database scaling to billions of rows

HCatalog	Schema and data type sharing over Pig, Hive and MapReduce
HDFS	Distributed redundant file system for Hadoop
Hive	Data warehouse with SQL-like access
Mahout	Library of machine learning and data mining algorithms
MapReduce	Parallel computation on server clusters
Pig	High-level programming language for Hadoop computations
Oozie	Orchestration and workflow management
Sqoop	Imports data from relational databases
Whirr	Cloud-agnostic deployment of clusters
Zookeeper	Configuration management and coordination

Getting data in and out

Improved interoperability with the rest of the data world is provided by Sqoop and . Sqoop is a tool designed to import data from relational databases into Hadoop, either directly into HDFS or into Hive. Flume is designed to import streaming flows of log data directly into HDFS.
Hive's SQL friendliness means that it can be used as a point of integration with the vast universe of database tools capable of making connections via JBDC or ODBC database drivers.

Coordination and workflow: Zookeeper and Oozie

With a growing family of services running as part of a Hadoop cluster, there's a need for coordination and naming services. As computing nodes can come and go, members of the cluster need to synchronize with each other, know where to access services, and know how they should be configured. This is the purpose of Zookeeper.
Production systems utilizing Hadoop can often contain complex pipelines of transformations, each with dependencies on each other. For example, the arrival of a new batch of data will trigger an import, which must then trigger recalculations in dependent datasets. The Oozie component provides features to manage the workflow and dependencies, removing the need for developers to code custom solutions.

Management and deployment: Ambari and Whirr

One of the commonly added features incorporated into Hadoop by distributors such as IBM and Microsoft is monitoring and administration. Though in an early stage,Ambari aims to add these features to the core Hadoop project. Ambari is intended to help system administrators deploy and configure Hadoop, upgrade clusters, and monitor services. Through an API, it may be integrated with other system management tools.

Though not strictly part of Hadoop,Whirr is a highly complementary component. It offers a way of running services, including Hadoop, on cloud platforms. Whirr is cloud neutral and currently supports the Amazon EC2 and Rackspace services.

Machine learning: Mahout

Every organization's data are diverse and particular to their needs. However, there is much less diversity in the kinds of analyses performed on that data. The Mahout project is a library of Hadoop implementations of common analytical computations. Use cases include user collaborative filtering, user recommendations, clustering and classification.

2.2.2 Network

Figure 3 depicts a common network architecture for the top-of-rack (ToR) switches within a Hadoop environment. These connect directly to the DataNodes and allow for all inter-node communication within the Hadoop environment. Hadoop networks should utilize inexpensive components that are employed in a way that maximizes performance for DataNode communication. The Hadoop software has many features to ensure availability in the event of hardware failure. This saves the common costs associated with expensive chassis-based switches with redundant power supplies and controllers. Some costs can be saved in the environment, with no impact to performance, by utilizing common ToR switches for all node connectivity. An example of such a switch is the ™ PowerConnect™ 6248.
This recommended architecture ensures connectivity in the event of a switch failure by creating link aggregation groups (LAGs) between the DataNodes and two separate switches, eliminating the potential for a single switch failure causing a loss of connectivity.

Figure 3. A common network architecture for the top-of-rack switches within a Hadoop environment.

Hadoop is a highly scalable software platform that requires a network designed with the same scalability in mind. To ensure maximum scalability, It recommends a network architecture that allows users to start with small Hadoop configurations and grow those over time by adding components, without requiring a rework of the existing environment. Figure 4 depicts a common EoR switch architecture for Hadoop to allow for maximum bandwidth between nodes, while scaling to sizes of environments commonly seen with Hadoop.
Redundancy for the network is most commonly implemented at the end-of-row (EoR) tier of the network. It recommends putting two switches in a redundant configuration at the EoR tier and connecting them redundantly to all ToR switches. This will ensure maximum bandwidth between all switches as the environment grows, as well as availability from a single switch failure anywhere in the fabric.

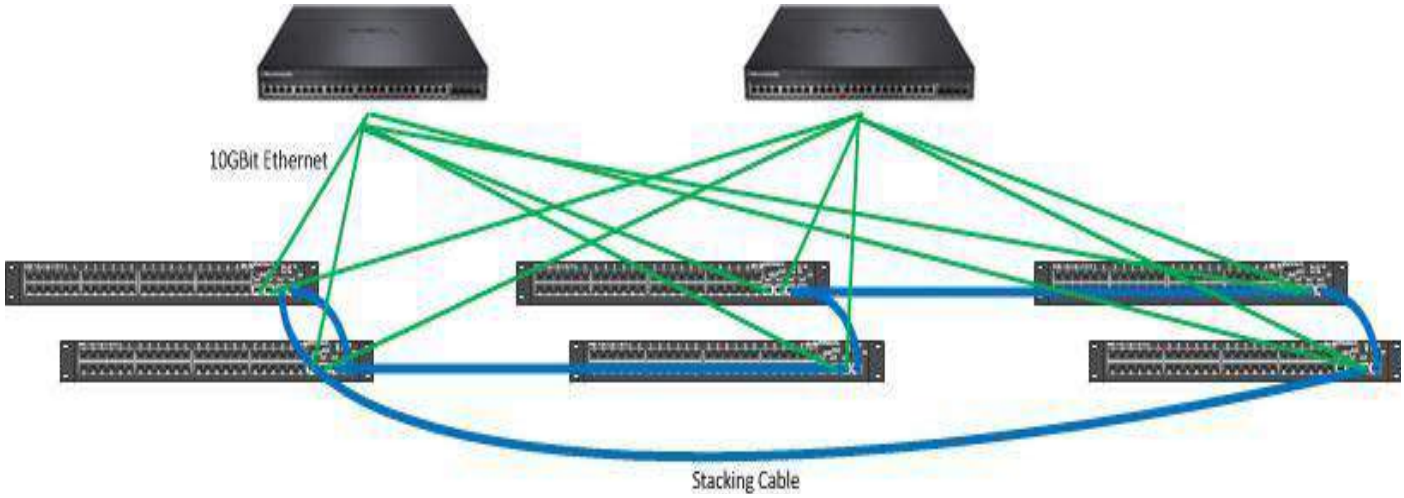


Figure 4. A common ToR switch architecture for Hadoop.

2.3 Hadoop performance benchmark

As with all distributed transaction systems, performance is a primary concern as the environment scales. One method for tracking performance is through the use of common benchmarks for comparing your environment to other similar environments, as well as comparing your environment to itself before and after changes. Within the Hadoop software ecosystem, there are several benchmark tools included that can be used for these comparisons.

2.3.1 Teragen

Teragen is a utility included with Hadoop for use when creating data sets that will be used by Terasort. Teragen utilizes the parallel framework within Hadoop to quickly create large data sets that can be manipulated. The time to create a given data set is an important point when tracking performance of a Hadoop environment.

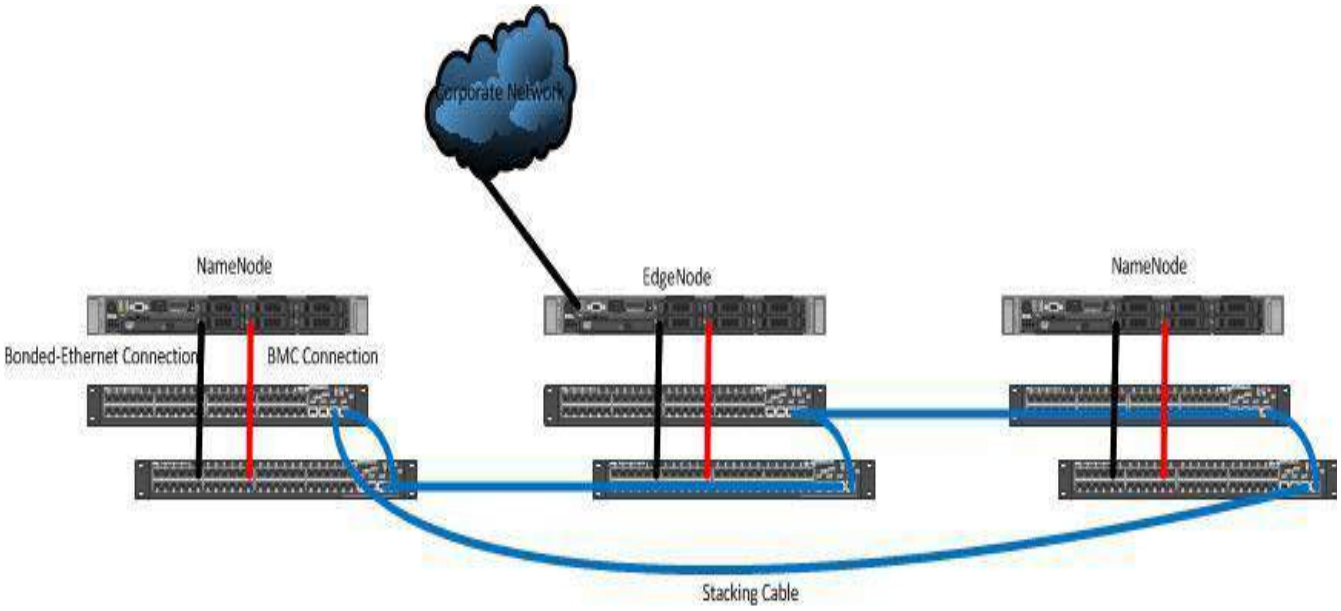
```
# bin/hadoop jar hadoop-*-examples.jar teragen 10000000000 in-dir
```

2.3.2 Terasort

Terasort is a compute-intensive operation that utilizes the Teragen output as the Terasort input. Terasort will read the data created by Teragen into the system's physical memory and then sort it and write it back out to the HDFS. Terasort will exercise all portions of the Hadoop environment during these operations.

```
# bin/hadoop jar hadoop-*-examples.jar terasort  in-dir out-dir
```

2.3.3 Teravalidate



Teravalidate is used to ensure the data produced by Terasort is accurate. It will run across the Terasort output data and verify all data is properly sorted, with no errors produced, and let the user know the status of the results.

```
# bin/hadoop jar hadoop-*-examples.jar teravalidate out-dir report-dir
```

2.4 Hadoop Challenges

With all large environments, deployment of the servers and software is an important consideration. It provides best practices for the deployment of Hadoop solutions. These best practices are implemented through a set of tools to automate the configuration of the hardware, installation of the operating system (OS), and installation of the Hadoop software stack from Cloudera.

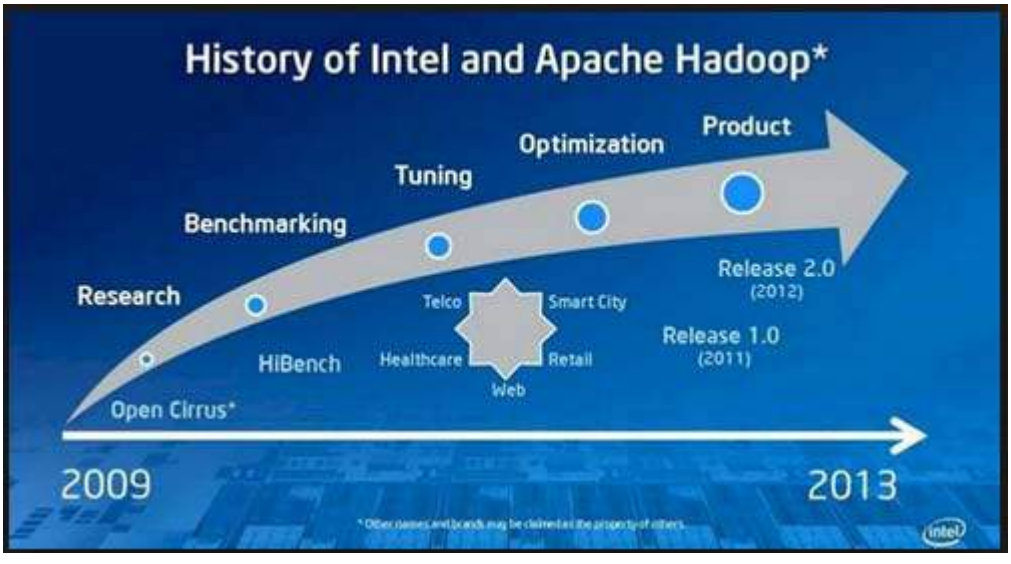
As with many other types of information technology (IT) solutions, change management and systems monitoring are a primary consideration within Hadoop. The IT operations team needs to ensure tools are in place to properly track and implement changes, and notify staff when unexpected events occur within the Hadoop environment.

Hadoop is a constantly growing, complex ecosystem of software and provides no guidance to the best platform for it to run on. The Hadoop community leaves the platform decisions to end users, most of whom do not have a background in hardware or the necessary lab environment to benchmark all possible design solutions. Hadoop is a complex set of software with more than 200 tunable parameters. Each parameter affects others as tuning is completed for a Hadoop environment and will change over time as job structure changes, data layout evolves, and data volume grows.
As data centers have grown and the number of servers under management for a given organization has expanded, users are more conscious of the impact new hardware will have on existing data centers and equipment.

2.5 Hadoop Best Practices

2.5.1 Intel's Best practices :

Intel's first internal big data computeintensive production platform with the Intel Distribution of Hadoop launched at the end of 2012(see below figure) . This platform is already delivering value in our first three use cases, helping them to identify new opportunities as well as reduce IT costs and enabling new product offerings.



Using their Hadoop implementation, Intel IT customers can now use the power of big data on a robust and scalable enterprise-ready platform based on the Intel Distribution that is integrated across their BI capability landscape. Their three use cases are fully operational, and a number of new uses cases are in various stages of development. Their internal big data platform expands their BI data container strategy, enabling the following:

- Structured and multistructured analytic data use cases.
- Platform design and architecture rightsized for today and the immediate future.
- Scalable and expandable design capable of meeting evolving needs.

Three Big Data Use Cases and Their Estimated Value By designing the platform for three specific use cases, Intel IT delivered nearly immediate value to the organization. Each use case is delivering or has the potential to deliver BI results worth millions of dollars to Intel.

CONTEXTUAL RECOMMENDATION ENGINE

Their big data platform enables a generic, reusable context-aware recommendation engine and analytic capabilities for a mobile location-based service. This service combines new, intelligent context-aware capabilities—including an algorithm for collaborative filtering—to help users find products, information, and services with map management technologies. The recommendation engine design is already being used for additional uses cases. In sales, they are using it to help decide what products should be offered to which resellers to maximize their sales.
Their recommendation engine may be offered in the future as a paid service.

LOG INFORMATION ANALYTICS FOR INCIDENT PREDICTION

Their big data platform is helping to identify and correlate potential issues opened with IT. They are tracking the event log data that precedes incident data and using linear regression and time-series analysis to predict future behavior and impact. Such predictive analytics help reduce incidents and the impact on users and IT, decreasing IT operations, support costs, and time-toresolution with proactive and predictive issue and symptom analysis.
We estimate that using our big data platform for incident prediction will provide a 10- to 30-percent reduction in new incidents at an estimated IT cost avoidance of USD 4 million over two years.

WEB ANALYTICS FOR CUSTOMER INSIGHT

They are landing and ingesting web data in Hadoop and integrating this external data with internal customer data to provide customer and network usage analytics for Intel.com and customer advertising. These web analytics give our sales and marketing groups the ability to perform deep analysis of web usage data for marketing or content navigation purposes. These analytics also provide the means to predict and adjust product positioning and pricing based on response to marketing campaigns, as well as improve the efficiency of the Intel supply chain. Intel sales and marketing groups estimate the ROI for web analytics in demand generation will be USD 10 million by 2014. Intel predicts that the smart analytics applied to the Intel supply chain will deliver up to USD 20 million in value in 2013 by improving availability of the right products at the right time and helping maintain the proper inventory levels for each region.

2.5.2 Dell Best Practices

Dell documents all the facility requirements, including space, weight, power, cooling, and cabling for Dell's defined reference architectures and provides this information to customers prior to any system purchase. This information allows you to make informed decisions regarding the best placement of your Hadoop solutions and the ongoing operational costs associated with them.

The Dell | Cloudera Hadoop solution addresses the challenges discussed in this white paper in a variety of ways to ensure streamlined deployment of low-risk solutions:

- The Dell | Hadoop solution includes the tools necessary to manage the environment's configuration from a single location and monitor the environment for unexpected changes.
- As part of the Dell | Hadoop solution, Dell provides reference architectures and associated benchmarks to streamline the Hadoop design process and minimize the risk to performance bottlenecks.

- This information allows you to make informed decisions regarding the best placement of your Hadoop solution and the ongoing operational costs associated with the solution.
- Dell provides results from lab tests and recommendations for Hadoop tunable parameters for your use in streamlining the time necessary to take a Hadoop installation from bare hardware to a production, operational environment.

Dell has worked with Cloudera to design, test, and support an integrated solution of hardware and software for implementing the Hadoop ecosystem. This solution has been engineered and validated to work together and provide known performance parameters and deployment methods. Dell recommends that you utilize known hardware and software solutions when deploying Hadoop to ensure low-risk deployments and minimal compatibility issues. Dell's solution ensures that you get maximum performance with minimal testing prior to purchase and minimal risk with solution deployment.

Supported solutions

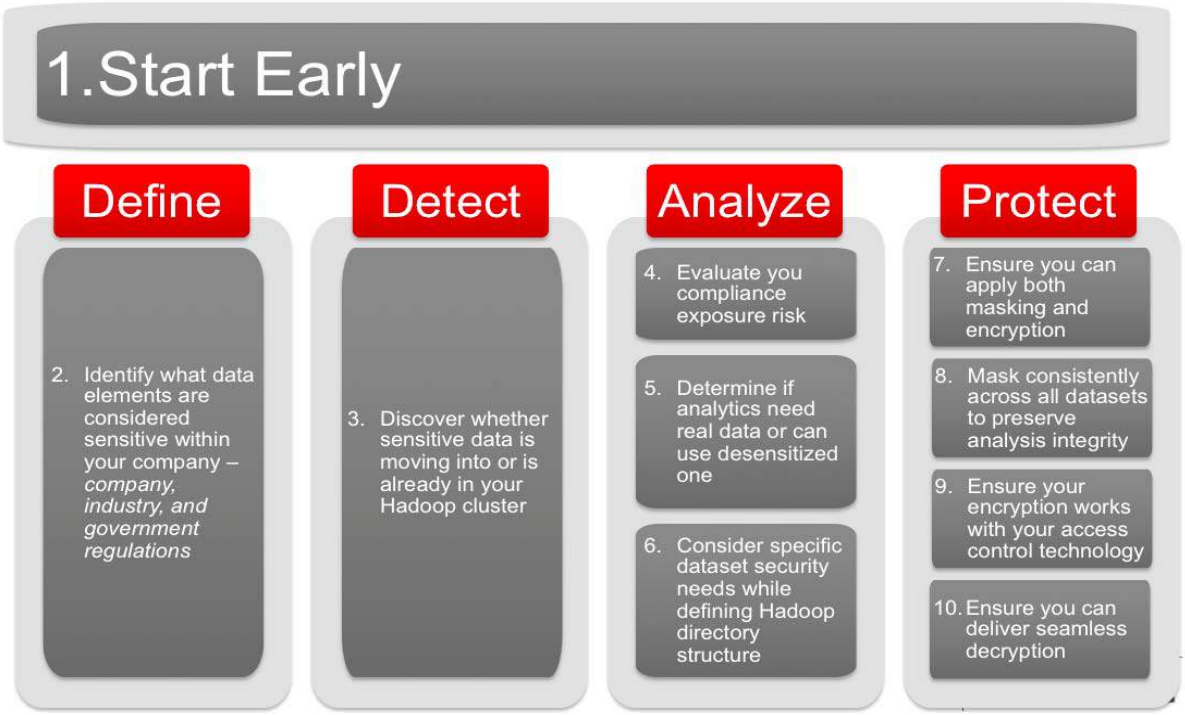
Dell recommends that you purchase and maintain support for the entire ecosystem of your Hadoop solution. Today's solutions are complex combinations of components that require upgrades as new software becomes available and assistance when staff is working on new parts of the solutions. The Dell | Cloudera Hadoop solution provides a full line of services, including deployment, hardware support, and software support, so you always have a primary contact for assistance to ensure maximum availability and stability of your Hadoop environment.

2.5.3 Need for Bul's eye approach to data privacy protection for hadoop

Bridging the priority gap between those who need to make data driven business decision and those who are responsible for enforcing compliance requires a viable action plan that provides the necessary protection with minimum impact on Hadoop rollout expectations. Dataguise has defined the following 10 simple steps to help companies define the right action plan:

1. Start Early! Plan your data privacy protection strategy in the planning phase of your Hadoop deployment, preferably before you start moving any data into Hadoop, so that you can prevent damaging compliance exposures for your company and avoid unpredictability in your rollout schedule.
2. Identify what data elements are defined as sensitive within your company. Consider company privacy policies, pertinent industry regulations and governmental regulations.
3. Discover whether sensitive data is embedded in data meant to be stored or already stored in your Hadoop deployments.
4. Determine your compliance exposure risk based on the information you collected.
5. Determine whether the data mining your business plans to perform requires access to real data or can use desensitized one. This will allow you choose the right remediation techniques (masking or encryption). If in doubt, remember that masking provides the most secure remediation while encryption gives you the most flexibility, should your needs change.
6. Ensure your data protection solution supports both masking and encryption remediation techniques, especially if you choose to keep both masked and unmasked version of sensitive data in separate Hadoop directories.

7. Ensure the data protection technology you use implements consistent masking across all your data files (Joe become Peter in all files) to preserve the accuracy of data analysis across every data aggregation dimension.
8. Determine whether a tailored protection for specific datasets is required and consider dividing Hadoop directories into smaller groups for which security can be managed as a unit.
9. Ensure your encryption solution interoperates with your access control technology and that both allow users with different credentials to have the appropriate access to data in your Hadoop cluster.
10. Ensure that when encryption is required, you also deploy technology that allows for seamless decryption to ensure an expedite access to data.



3. Conclusion

Based on our study, it believes obtaining optimal results from a Hadoop implementation begins with carefully choosing themost advantageous hardware and software. Fine-tuning the environment to achieve the highest ROI calls for an in-depth analysis of available Hadoop distributions, cost-efficient infrastructure choices, and careful integration with the existing BI environment to ensure efficient data transfer, strong security,and high availability.an iterative approach enabled us to costeffectively develop a big data platform for our three targeted use cases that can be ultimately scaled to handle a diverse range of additional use cases. Through this process, we have also been able to develop a number of best practices and how-to procedural guidelines that will continue to guide us as we expand our big data platform and build additional big data platforms in the future.

To cope with the increasingly rich array of data sources, constant change, rapidly growing volume of data, and complex set of analytics needs of the enterprise,information technology teams will need to evolve their infrastructure to cope in a way that doesn't break the budget. By using the right tool for the job, this can be achieved.In this paper we've discussed one potential architecture which is a very robust solution to the challenges . Hadoop's MapReduce and HDFS use simple, robust techniques on inexpensive computer systems to deliver very high data availability and to analyze enormous amounts of information quickly. Hadoop offers enterprises a powerful new tool for managing big data.

4. References

Google MapReduce
http://labs.google.com/papers/mapreduce.html

Google File System
http://labs.google.com/papers/gfs.html

Apache Hadoop
http://hadoop.apache.org/

Cloudera Hadoop
http://www.cloudera.com/products-services/enterprise/

Teragen
http://hadoop.apache.org/common/docs/r0.20.2/api/org/apache/hadoop/examples/terasort/TeraGen.html

Terasort
http://hadoop.apache.org/common/docs/r0.20.2/api/org/apache/hadoop/examples/terasort/TeraSort.html

Teravalidate
http://hadoop.apache.org/common/docs/r0.20.2/api/org/apache/hadoop/examples/terasort/TeraValidate.html

Hadoop Distributed File System
http://hadoop.apache.org/hdfs/