April 2014, HAPPIEST MINDS TECHNOLOGIES

Big Data - Infrastructure Considerations

Author
Anand Veeramani / Deepak Shivamurthy

happiest minds
The Mindful IT Company
Born Digital . Born Agile

SHARING. MINDFUL. INTEGRITY. LEARNING. EXCELLENCE. SOCIAL RESPONSIBILITY.

Copyright Information

This document is an exclusive property of Happiest Minds Technologies It is intended for limited circulation.

Contents

Abstract

Big Data is a much talked about technology across businesses today. A vast majority of organizations spanning across industries are convinced of its usefulness, but the implementation focus is primarily application oriented than infrastructure oriented. However, the infrastructure architecture for any Big Data cluster is of critical importance because it affects the performance of the cluster. Modeling the infrastructure architecture for Big Data essentially requires balancing cost and efficiency to meet the specific needs of businesses.

This paper takes a closer look at the Big Data concept with the Hadoop framework as an example. We look at the architecture and methods of implementing a Hadoop cluster, how it relates to server and network infrastructure and the typical storage requirements for a Big Data cluster. We also look at Information Security in the context of Big Data at a high level. The content presented here is largely based on academic work, experiments conducted within Happiest Minds Technologies labs and experiences derived from implementations for our customers.

Introduction

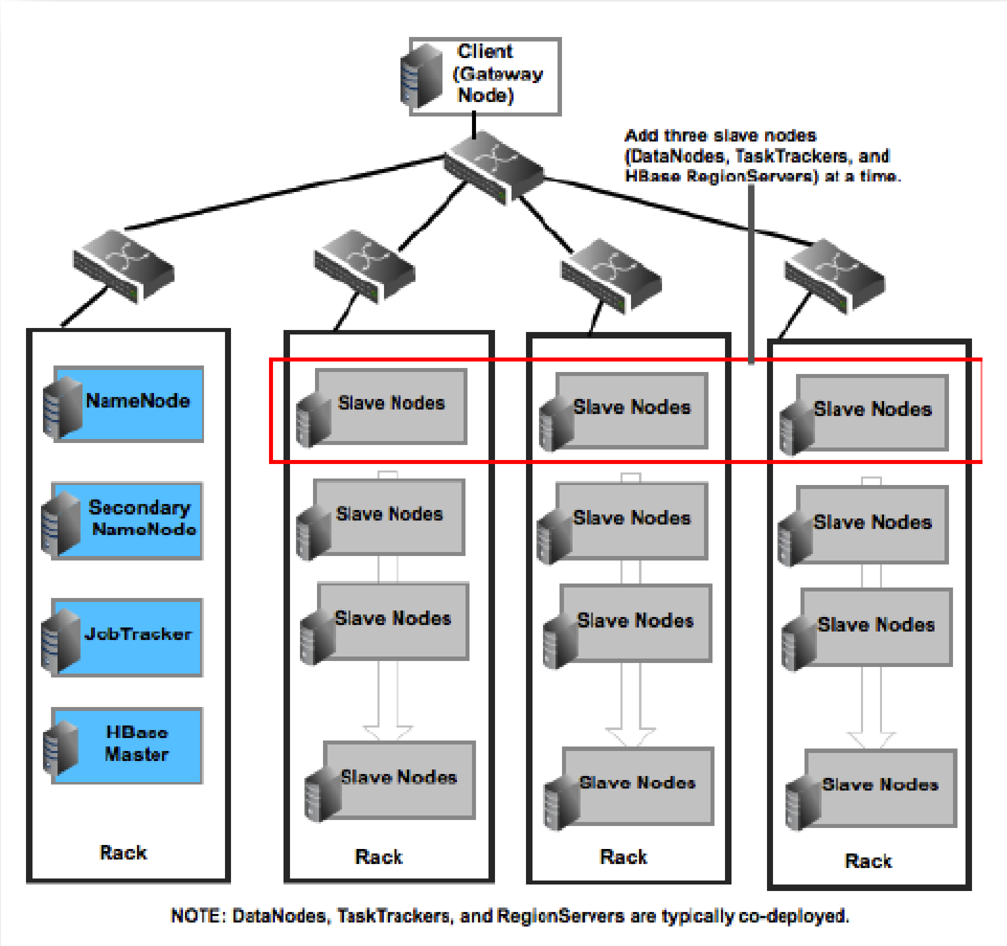
The volume of data generated globally is growing at a phenomenal scale and pace. The variety of data generated further additions to its complexity. Data is continuously being generated by sensors and humans; and volumes will grow exponentially over time. Cellular Networks and Social Networking applications are some of the major contributors to data generation. Big Data has opened up a completely new avenue for organizations to leverage these growing information assets to better understand and compete in the market.

Big Data can be defined as any data repository with the following characteristics:

- Handles large amounts (a petabyte or more) of data
- Has distributed redundant data storage
- Processes tasks in parallel
- Provides data processing (MapReduce or equivalent) capabilities
- Centrally managed and orchestrated
- Is relatively inexpensive
- Accessible —easy to use and available
- Extensible — basic capabilities can be augmented and altered

The current focus of the development community globally is driving the creation of best practices and learnings for Big Data adoption. Within Happiest Minds, we have experienced the pressing need for a Big Data specific architecture framework. Based on our understanding of the Big Data architecture design process and its limitations, this paper recommends a robust approach to address these limitations. The process is adaptive and can be extended to adopt best practices, as it evolves.

Going forward, we shall use Hadoop as an example of a Big Data product. The most important components of Hadoop are the Hadoop Distributed File System (HDFS) which provides storage and MapReduce, for parallel processing of large data sets.



Hadoop Cluster Node-level architecture

NameNode:
The NameNode is the master of the HDFS that directs slave DataNodes to perform low level input/output tasks.

DataNode:
Each slave machine, referred to as DataNode, reads and writes from HDFS blocks to actual files on the local file system.

Secondary NameNode:
Secondary NameNode (SNN) assists in monitoring the state of the HDFS cluster.

JobTracker:
JobTracker is the master overseeing the overall execution of a MapReduce job. It acts as a liaison between Hadoop and the application.

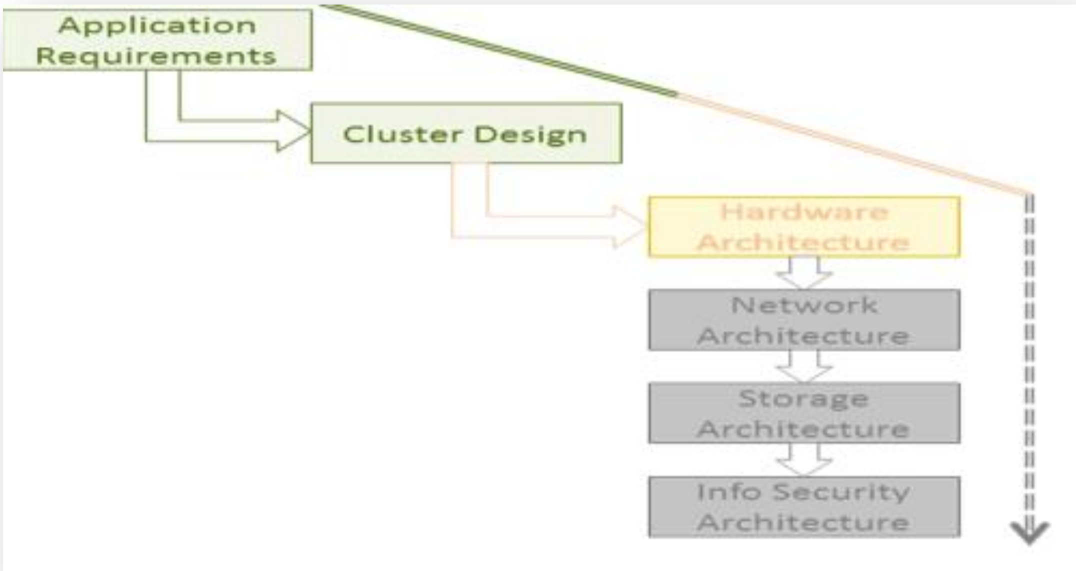
TaskTracker:
TaskTracker manages the execution of individual tasks on each slave node.

The Current Big Data Adoption Process

Huge growth in volumes, the growing variety and the pace at which data is generated are making a big impact on organizations' business decisions. Data storage requirements are becoming difficult to predict and provide for. The adoption of Big Data may be driven by two perspectives: the application perspective of analytics or the infrastructure perspective of storage.

Presently, the global Big Data adoption trend focuses primarily on application and not infrastructure. Hence, there is scope for improvement in taking a holistic view of the application and infrastructure requirements, while designing a cluster. It is imperative to understand that the underlying data processing algorithm (MapReduce or equivalent) will produce efficient output only if the data storage cluster is designed well.

The overall Big Data adoption process, as it stands today, is depicted in below diagram.



Following is a detailed explanation of the current Big Data adoption process:

1. Cluster Design: Application requirements are analyzed in terms of workload, volume and other associated parameters based on which the cluster is designed. Cluster design is not an iterative process. The initial setup is verified and validated with a sample application and sample data before being rolled out. Although Big Data cluster design allows flexibility in fine-tuning the configuration parameters, the large number of parameters and their cross-impacts introduce additional complexity.

2. Hardware Architecture: The key success factor for Hadoop clusters is the usage of high quality commodity equipment. Most Hadoop users are cost conscious and as clusters grow, their cost can be significant. In the present scenario, the hardware architecture requirements for the NameNode are higher RAM and moderate HDD. If the JobTracker is a physically separate server, it will have higher RAM and CPU speed. DataNodes are standard low-end server class machines.

3. Network Architecture: Currently, network architecture is not specifically designed for Big Data, i.e., inputs from cluster design and application requirement are not always mapped to it. Standard network setup within the existing data center is used as the backbone. In most cases, this may result in overestimated network deployment and, at times, have a negative effect on the MapReduce data processing algorithm. Hence there is significant scope for creating concrete guidelines related to designing network architecture for Big Data.

4. Storage Architecture: Most enterprises have huge investments in NAS and SAN devices. When implementing Big Data, they attempt to re-use this existing storage infrastructure even though DAS is the recommended storage for Big Data clusters. Parameters like type of disk, shared-nothing vs shared something, are often not taken into account.

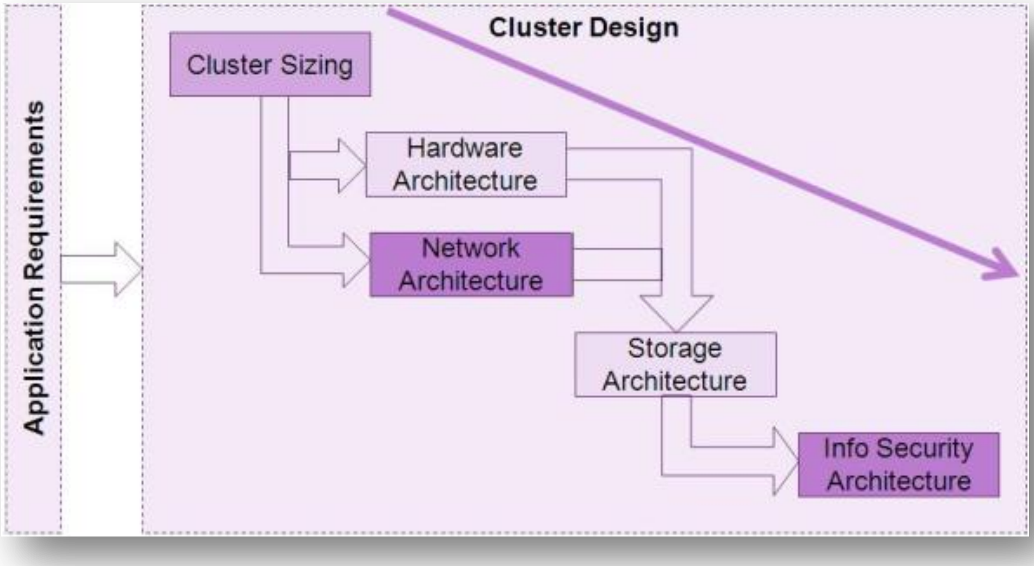
5. Information Security Architecture: General examination of different Big Data implementations shows that security features are sparse and aftermarket security offerings are not fully tailored to these clusters. Findings show these deployments to be largely insecure and wholly reliant on network and perimeter security support.

Big Data Adoption Process - Recommended

While designing the Big Data architecture for an enterprise setup, it is necessary to take a comprehensive approach.

- Application requirements should drive the overall cluster design activity including cluster sizing, hardware architecture, network architecture, storage architecture and information security architecture.
- Hardware architecture should be based on application requirements and cluster sizing.
- Network architecture should also be derived from application requirements and cluster sizing. This can be worked out in parallel to hardware architecture design.
- Storage architecture should depend upon cluster sizing, hardware architecture and network architecture. Application requirements should help fine tune the storage architecture.
- Information security architecture should depend upon hardware architecture, network architecture and storage architecture. Application requirements should validate the security architecture.

A view of the recommended adoption process is depicted in below diagram. The darker boxes indicate the complex steps.



1. Application Requirements and Cluster Sizing: There are a number of important cluster configuration parameters to be derived from the application requirements.

- Primary and Secondary NameNode
- JobTracker as NameNode or separate
- Number of DataNodes
- Number of Mappers per node
- Number of Reducers per node
- Replication factor
- Block size
- NameNode server thread count

- Number of parallel copies run by Reducers
- Heap size of child JVMs of Mappers
- Heap size of child JVMs of Reducers
- Memory allocated for the in-memory file system at the reducers
- Sort factor
- Memory limit for sorting
- Size of Read/Write buffer used in Sequence files

2. Cluster Design and Hardware Architecture: Hadoop is built to handle component failure well and to scale out on low cost gear; thereby eliminating the need for RAID cards, redundant power supplies and other per-component reliability features. Error-correcting RAM and SATA drives with good MTBF numbers should be used, as this assures reliable computations. Hard drives are the largest source of failures. Therefore, care must be taken in making the right choice of hard drives with focus on reliability and efficiency. Hardware architecture for each cluster component should be decided in line with the application requirements and cluster design.

3. Network Architecture: The underlying network architecture impacts the efficiency of the MapReduce algorithm. In fact, each of the networking components affects the performance of a cluster and this becomes the toughest variable to nail down. Since Hadoop workloads vary a lot, the key is to use adequate network capacity to allow all nodes in the cluster to communicate with each other at reasonable speed and cost.

4. Storage Architecture: While considering storage for Big Data, the disk size is more important than the seek time. The architectural parameters to be considered for storage are as follows:

Characteristics of Big Data Storage

Scalable**:** Storage should be scalable in terms of size, throughput and speed of access.
Provides tiered storage: It is critical for the storage system to be able to manage the "tiering" of data across the range of media types: flash, fast disk, slower disk and tape.
Makes content widely accessible: Storage should distribute data geographically so that it is closer to users.

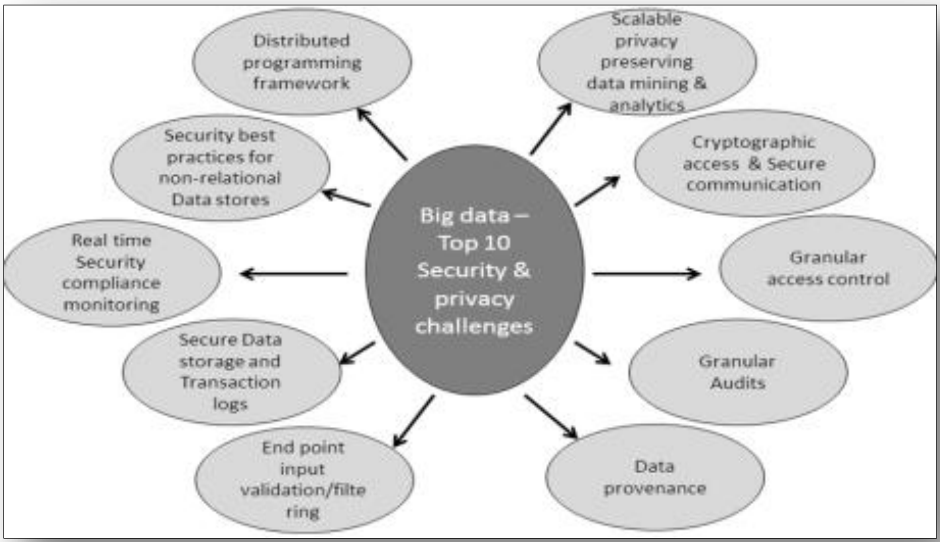
Supports workflow automation: Big Data must be delivered to users in the context of a workflow. For this reason, Big Data storage architecture must support easy integration of workflow.

Supports integration with legacy systems, and analytical and content applications: A well designed Big Data storage system should be heterogeneous and flexible. It should offer interfaces that allow direct access to the Big Data storage functionality.

Supports integration with cloud ecosystems: An ideal Big Data storage system must be built from the ground up, to be cloud enabled.

Self-managing: The Big Data storage system should have the built-in ability to handle failures; it must accommodate component failures and repair itself without intervention.

5. Information Security Architecture: To handle information security of Big Data, the architectural and operational security aspects need to be looked into. By its very nature, Big Data security has inherent challenges to tackle. And many of these challenges still need appropriately devised mechanisms to address them. The top 10 challenges of handling Security within Big Data environment are illustrated below.



Big Data clusters share most of the same vulnerabilities as web applications and traditional data warehouses. Concerns over how nodes and client applications are vetted before joining the cluster, how data at rest is protected from unwanted inspection, privacy of network communications and how nodes are managed, remain in focus. The security of the web applications that front end Big Data clusters is equally important.

As many clusters are being deployed within virtual and cloud environments, they can leverage vendor supplied management tools to address operational security issues. While these measures cannot provide fail-proof security, a reasonable amount of effort can make it considerably more difficult to subvert systems or steal information.

Big Data in Cloud

With the cloud computing and Big Data trends converging, the era of Big Data in the cloud commences. From technology perspective, focus will shift away from the software powering Big Data projects towards the infrastructure necessary to support it.

For many Big Data scenarios, information comes from outside the company, for e.g., social media, demographic data, web data, events, feeds, etc. The elasticity of the cloud makes it ideal for Big Data analytics -- the practice of rapidly processing large volumes of unstructured data to identify patterns and improve business strategies.

Making the storage perform at a level that enables the kind of data analysis Big Data needs is important. Also, the cloud service provider's inability to provide infrastructure from the same physical location results in network latency and proves detrimental to the overall performance of MapReduce. These factors prove to be the biggest detriments to the use of cloud for Big Data processing.

Conclusion:

Looking at architecture design for Big Data only from the application perspective gives an isolated view. Similarly, taking the infrastructure perspective alone into consideration may negate the advantages of Big Data. A comprehensive strategy covering application and infrastructure aspects while architecting Big Data is most desirable. The commonly followed process for Big Data implementation today still has scope for improvement in this regard.

Big Data implementation is a multi-skilled discipline. It has major dependency on the underlying infrastructure and the deployment architecture. Once a decision to use Big Data has been made by an organization, the network, storage and security architecture associated with the deployment architecture must be worked out in an iterative manner before finalizing it. The rule of thumb is to exploit the benefit of low total cost of ownership (TCO) by using commodity hardware and get almost real-time application performance.

References

[1] Apache Foundation, Hadoop overview,
http://hadoop.apache.org/

[2] MapReduce.org, information about MapReduce Framework,
http://www.mapreduce.org/

[3] Forbes, Ten properties of the Perfect Big Data storage architecture,
http://www.forbes.com/sites/danwoods/2012/07/23/ten-properties-of-the-perfect-big-data-storage-architecture/2/

[4] Securosis, Securing Big Data: Security Recommendations for Hadoop and NoSQL Environments,
https://securosis.com/Research/Publication/securing-big-data-security-recommendations-for-hadoop-and-nosql-environment

[5] Forbes, BigData meets cloud,
http://www.forbes.com/sites/forrester/2012/08/15/big-data-meets-cloud/

[6] Virginia State University, Evaluating MapReduce System Performance: A Simulation Approach,
http://scholar.lib.vt.edu/theses/available/etd-08282012-152556/unrestricted/Wang_G_D_2012.pdf