I J C T A, 9(26) 2016, pp. 301-306

An Integrated Approach for Configuring Hadoop Clusters by Ambari on Horton Sandbox

R. Kannadasan[1], G. Sivashanmugam[2], V. Vijayarajan[3], N. Prabakaran[4], A. Krishnamoorthy[5] and K. Naresh[6]

ABSTRACT

In Information Technology satisfying customer needs is still remains a milestone because of their increasing demands. When taking the comparison strategy Industry is not only altered in the way of providing solutions, also handling of techniques and resource adaptability is taken into a new level. All the business activities are moving towards Integrated Approach, so the techniques involved should also support integrated platforms this to be followed by both existing techniques as well as newly evolved techniques. In this paper a very important existing technique Hadoop as an experimental platform configuring it by a newly born tool named as Ambari, to enable integrated approach on Hadoop clusters without disturbing its applications. To do this common platform which supports Hadoop, Ambari as well as its entire component modules, for this we chosen Horton Sandbox it is very helpful to integrate Hadoop with other relevant technologies and sandbox is readymade composite software platform were makes all the modules are installed priory, which allows our work very easier. In this experimental Hadoop and its clusters on different nodes are configured and controlled by Ambari which runs on Single machine. Our results clearly showing ambari acts as a central management system to monitor Hadoop clusters bring plenty of benefits to the Hadoop Users.

Keywords: Hadoop, hortonsandbox, ambari,parallel processing and clustering.

1. INTRODUCTION

In conventional techniques, all the processes are carried with computable computational resources like processing capacity, storage, programming tools. All the organizations followed with their own processing server which includes handling of information's, implementation of techniques are done with their administration privileges itself [1]. Days on data involved in the process increases so computation requirements also increased to handle huge data process which make the organizations invent adverse techniques to provide solutions. This changed the entire behavior of the conventional system leads to new central repository system. In this system, a High end computational servers are maintained centrally and their resources are distributed to the organizations who bind with the repositories [2]. The reason behind this system birth is all new technologies are not able to adopt by the organizations. Sometimes these technologies require high cost and adopting new technology may eradicate entire existing system so vendors stand in border to adopt these technologies, due to increasing in user demands lifespan of technology also reduced day by day new things are arriving in industry which create so many changes in Interoperability of operations [3].

These modern eras are fully based on sharing of their resources in Distributed platform. Maintaining high end computational resources all the time is also not easiest one, so they formed clusters which comprise of several machines to meet the sudden increase of demand. By having clustering technique, we can meet

[1,2,3,4&5] SCOPE, VIT University, Vellore 632014, Tamilnadu, India.

computational demands by scale up with a new machine. This concept is followed in modern era, in that a famous technology plays an important role in forming clusters and sharing the resources among the clusters on distributed network is Hadoop. Later in increase in usage of Hadoop cluster, they create integrated tools to monitor Hadoop clusters for easy management. This paper describes the working nature of Hadoop with detailed process involved in it in section 2.A.

2. RELATED WORKS AND ARCHITECTURE SETUP

A. Hadoop Structure

Hadoop provides essential support to run various applications on clustered platform; it also provides an extensive volume of storage with unbelievable processing capacity. It has property to scale up from a single machine to thousands of servers and offer local computation and storage as a local resource. Hadoop was efficiently designed to distribute the workload to other servers without worrying about the magnitude of data, it used to process data order from thousands of gigabytes to petabytes also. Even we need it can extend the handling level because it was specially developed for large scaled distributed applications [4] [5].

Processing the task with a large volume of data requires parallel applications on multiple machines, performing this type of large scale system are very difficult because these types of applications have high failure probability factors. It also requires a better programming designs to bring cooperation among the machines whenever they involved in processing the task [6]. Generally, if we talk about failure they affect the system design during crash occurs, that time recovery the program and bringing back the system stability is impossible. However, the system is tightly coupled or loosely coupled partial failures are unexpected due to sudden network break down. Sometimes failures may occur due to overheat, system crash, driver faults, congested disk spaces that affect data passage between the nodes. The other problems like improper presentation, invalid data because of protocol variation among multiple cluster systems, lock files due to wrong clock synchronizations, atomic distributed transactions may lose their network connectivity, etc, cannot be rectified easily. In additional to that component and progress starvation causes failures to the system. In order to process the Distributed system successfully without any collapsing is one of the challenging jobs [7]. The engineers found a solution to these types of failures through a software paradigm called as Hadoop: it was designed robustly to handle issues mainly raised from hardware and data congestions. Initially, the components of Hadoop are help to manipulate processor strength, memory capacity, storage requirements and adopting bandwidths to balance network structures. After this large scale distributed applications are managed efficiently [8] [9]. During these days, the Hadoop system achieved several improvements but faces challenges on creating synchronization between multiple machines. To overcome this, they simplified the programming model and add functionalities like automatic distribution of data and work among multiple cluster nodes, applying parallelism on CPU cores and utilize it, this makes Hadoop reach next level were all the vendors started to use Hadoop for their distributed applications. Maximum Data center continues their business with the help of Hadoop only. Next we shown the approach used in the Hadoop and demonstrated how Hadoop extracts complexities present in distributed system [10].

B. Architecture Design

The above mentioned building blocks play a vital for Hadoop success let we see them and the entry level architecture design of Hadoop is shown below.

Two important basic modules present in the Hadoop technology are,

- *Hadoop Distributed File System (HDFS):* Stores data on multiple clusters and provide high data flow path between the machines present in the cluster.

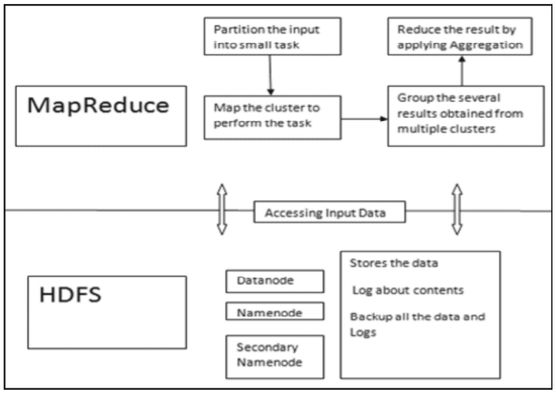

Figure 1: Entry level Architecture of Hadoop

- *Map Reduce Framework:* It is a programming based design used to process the data stored on the cluster machines from that result are computed.

C. HDFS

It is java based Hadoop framework which gives distributed, scalable and portable file system access to Hadoop users. The function of HDFS component to store large files across multiple clusters, access of files can be achieved by programming API. It uses Thrift API to convert the file output to user language it supports maximum formats used in the industry, it supports both command line interface as well HDFS-UI web app over Hyper Text Protocols [9]. The terminology of HDFS is done by Name node and Data node only. The work carried by HDFS is simply refer as Data Distribution the structure of HDFS file system is remains same as Unix file system except the idea used to store the data on multiple machines. The containers of HDFS and its purpose are explained below.

Data node

It creates a Data node on working machines and stores the data on it.

Name node

This type of node will present only on master machines it holds the full control over accessing the stored data. It maintains a log that contains information about data input stored in which Data node and corresponding machine addresses.

One more Node is existing in HDFS structure it doesn’t involve in any functionality, but it maintains separate service that keeps the copy of both the data images and edit log take place on data node named as Secondary Name node.

D. Mad Reduce

This is another important framework in Hadoop fundamental is MapReduce layer. This layer workflow is done by two components are follows,

1. Programmed API is created to follow the workflow.
2. A set of services is created to execute the workflow.

The input from HDFS is taken as an assignment then it starts the handling of the task by two processes namely Mapper and Reducer. Mapper is used to partitioning the job and map portioned job to the clusters after processing over it sort and group those results and pass to Reducer which perform aggregation to reduce the results obtained.

E. Dilemma Statement

In this section, we have shown the importance of Hadoop Technology and we presented complication present in handling Hadoop and needed tools developed to deploy Hadoop in the integrated approach. The automatic allocation of the job is done by job tracker which assigns jobs to the nearest cluster which contains enough data inputs. This process lags on considering current load execution of particular machine and hence, this affects the entire process. If one Task tracker is slow then it makes another task tracker to wait for particular single task execution, this degrades the entire performance of Hadoop [10], this has been completely overhaul in upcoming Hadoop version is shown below. Another new tool added to Hadoop framework is YARN is used to separate the process into resource management and processing components. By this scheduling of user's application is carried out. Another variant added with Hadoop is a common library and utilities needed for Hadoop modules done by Hadoop common.

After achieving several shapes inside Hadoop framework practicing entire modern Hadoop architecture within the individual platform become tedious, comprising all the modules I a single machine and deploying all these tools remains a headache to the users. In this paper, we experienced Horton sandbox which provides a better platform to practice entire Hadoop framework. By Horton sandbox, we achieved an integrated solution for practicing Hadoop frameworks.

E. Hortonworks Monitoring

Horton comprises most recent innovations of Hadoop framework that provides an open environment for distributed data processing across clusters of machines. It comprises several products maximum it deals with the handling of large data sets from any formats. Inside Horton work, a framework called Horton work [8] Data Platform is an open source platform promote integrated approach for Hadoop with existing other data architectures. Hence, organizations can deploy Modern Data architecture easily with the help of Horton Data platform, with YARN as its centralized architect helps to support several data workloads concurrently. The other needed tools to manage Hadoop below,

Ambari

It afford essential environment to manage and helps to keep continuous track on Hadoop clusters. Ambari settled with readymade operational tools and APIs to simplify the complex operations on Hadoop clusters.

Sandbox

It is a virtual application helps to run Horton work Data Platform (HDP) on any machine at any place. We can analyze sandbox data with many intelligence techniques. It provides fully controlled environment so all the tools for running applications are composited inside the sandbox. Sandbox was preconfigured to support portable environment for Hadoop machines. The representation of modern data structure is shown below with the working flow of Ambari and Map Reduce frameworks.

3. EXPERIMENTAL MONITORING SETUP

A. Configuring

We have undergone configuring Hadoop with the help of Ambari inside the sandbox and we excited by seeing the working results. By working on Sandbox we don't want to worry about the number of Hadoop clusters extent. Any number of Hadoop clusters [10] can be organized and managed within the single unit of the sandbox. The configuration structure of ambari and monitoring Hadoop clusters inside the sandbox was

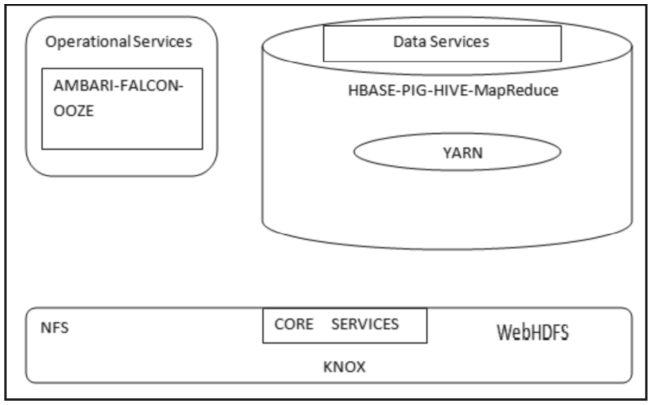

Figure 2: Modern Hadoop Architecture

shown below. Hadoop comprises several components like HDFS, MapReduce, YARN, Tez, Nagios, Ganglia, Hive, HBase, Pig, Sqoop, Oozie, Zookeeper, Falcon, Slider, Storm and Flume. Installing all these component is very crucial activity. With the help of ambari, all these components are deployed as a single piece of software in use.

B. VM Machine

All these setups are executed on the virtual machine with the help of VAGRANT environment. After configuring VM setup next we have to activate NTP service of the machine. From the VM machine, we can control the master machine and other networking machines.

C. L and I Ambari

Launching and Installing Ambari [9] can be downloaded directly from the repository using the command above. yum install ambari-server: Once installation over we can setup Ambari with password initialization for control privileges.

D. Configuring Hadoop Cluster

Once logged inside the Ambari by filling the details of host machines it will show the configured wizard with all complete tools that are shown in fig 5. This is our own Hadoop machine which configuring two machines installed by the virtual machine and master machine. By having the virtual machine, we monitored the host machine with the help of Ambari dashboard. The dashboard showing number machines configured under Hadoop, CPU usage of virtual machines, cluster node process, name node process. From this, we presented the integrated approach to monitoring any number of Hadoop clusters from a single ambari node.

Figure 3: Installing VMe

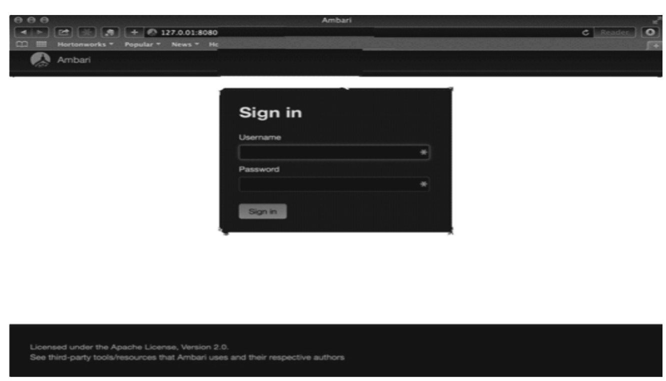

Figure 4: Ambari setup Entry

Figure 5: Dash board view of ambari

4. CONCLUSION

Hadoop stands an everlasting application on providing distributed applications, by having sandbox tool it leverages the power of open source to all enterprises. By these models, we are able to gain robust control of Hadoop applications. The consolidated tools help us to provide multiple heterogeneous access methods for different users at once, it also provides scaling of servers up to any extent. From this type open source tool instead of "bulk on trade" small enterprises also able to practice Hadoop for their applications.

REFERENCES

[1] Z. Ren, X. Xu, J. Wan, W. Shi, and M. Zhou, "Workload characterization on a production Hadoop cluster: A case study on Taobao," in *Workload Characterization (IISWC)*, 2012 IEEE International Symposium on, 2012, 3–13.

[2] T. White, Hadoop: The definitive guide. "O'Reilly Media, Inc.," 2012.

[3] G. Horn, "Mobile hadoop clusters." *Google Patents*, 2012.

[4] G. Wang, A. R. Butt, P. Pandey, and K. Gupta, "A simulation approach to evaluating design decisions in MapReduce setups.," *in MASCOTS*, 9, 1-11, 2009.

[5] M. Ding, L. Zheng, Y. Lu, L. Li, S. Guo, and M. Guo, "More convenient more overhead: the performance evaluation of hadoop streaming," *in Proceedings of the 2011 ACM Symposium on Research in Applied Computation*, 307–313, 2011.

[6] C. Dai, Y. Ye, T. J. Liu, and J. J. Zheng, "Design of High Performance Cloud Storage Platform Based on Cheap PC Clusters Using Mongo DB and Hadoop," in Applied Mechanics and Materials, 380, 2050–2053, 2013.

[7] D. Eadline, Hadoop 2 Quick-Start Guide: Learn the Essentials of Big Data Computing in the Apache Hadoop 2 Ecosystem. Addison-Wesley Professional, 2015.

[8] D. Zburivsky, Hadoop Cluster Deployment. Packt Publishing Ltd, 2013.

[9] S. Wadkar and M. Siddalingaiah, "Apache ambari," *in Pro Apache Hadoop, Springer*, 399–401, 2014.

[10] G. Singh, Monitoring Hadoop. Packt Publishing Ltd, 2015.