International Journal of Application or Innovation in Engineering & Management (IJAIEM)
Web Site: www.ijaiem.org Email: editor@ijaiem.org
Volume 6, Issue 7, July 2017 **ISSN 2319 - 4847**

Hadoop : A Framework for Big Data Processing & Storage

Yojna Arora[1], Dr Dinesh Goyal[2]

[1] PhD Scholar, Department of Computer Science & Engineering Gyan Vihar School of Engineering & Technology Suresh Gyan Vihar University, Jaipur, India

[2] Professor, Department of Computer Science & Engineering Gyan Vihar School of Engineering & Technology Suresg Gyan Vihar University, Jaipur, India

ABSTRACT

Big Data has become big enough to be handled by traditional technologies and algorithms. Rather than bringing the computation to data it is the other way round, it is cheaper to bring the computation algorithm to place where data is residing. Many techniques to deal with this Big Data have been implemented in recent past, Hadoop is one such technology. It is an oen source software library that provides distributed computing of large data sets. This paper contains a detailed description of Hadoop, its architecture components, the applied Hadoop stack used by industries and processing framework

Keywords: Hadoop, Map Reduce, HDFS, YARN,

1. INTRODUCTION

There has been a tremendous increase in the amount of data generated. In the past decade average size of a corporate database was in the range of Gigabytes (GB) which has exponentially increased up to Terabytes (TB) or even Petabytes (PB). The inability to handle such huge amounts of data gave rise to many computational challenges [1], [2]. In order to overcome these computational challenges and to analyse this Big Data ina better way various techniques such as High Performance Computing [3], Grid Computing [4], Mobile Agent Based [5], Data Mining [6], Hadoop[7] [8] and Map Reduce [9] are implemented.

The paper further includes following sections. Section 1 gives an overview about Hadoop & its evolution. Section II explains about the detailed architecture of Hadoop. Further Section III is about the practical implementation of Hadoop stack in various corporate organizations. Section IV explains about processing frameworks of Hadoop available in literature. Lastly, Section V concludes the paper.

2. HADOOP [10], [11], [12]

Hadoop is an open source software library which provides distributed processing of large data sets across clusters of computers. It offers analytics at a low cost and high speed as it processes large data sets at a granular level. The components of Hadoop are designed keeping in mind the frequent Hardware failure. Hadoop framework guarantees automatic recovery from Hardware failures thus providing high performance, reliability and scalability.

Developed in January of 2008, it got the contribution from both academia and corporate. Most of its code is written in Java with some code has been written in C and shell scripts. Since Map reduce Java code is common so it can be used with any programming language to implement Map and Reduce.

2.1 Evolution of Hadoop

The idea of Hadoop came from the first implementation of Google File system in [38] and later Map Reduce in [39] Finally, Hadoop 1.0 was introduced in April 2006 which included the Data storage component HDFS and computation component Map Reduce and it further continues to evolve by many contributors. HDFS stands for Hadoop distributed file system, which splits and replicates the large file onto 3 servers thus allowing fault tolerance. The other part Map Reduce is the computing algorithm which splits request into small parts and send each to different servers.

Later on Hadoop 2.0 was released. The fundamental idea of Hadoop 2.0 was to split up two major functionalities i.e. job tracker resource manager and job scheduling and monitoring. The new component YARN (Yet another Resource Negotiator) allows multiple data processing engines such as SQL, real time streaming, batch processing to handle data stored in a single platform.

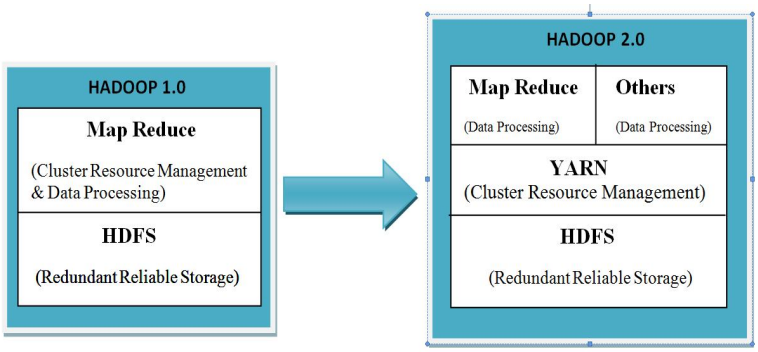

Fig 1. Evolution of Hadoop

2.2 Hadoop Architecture

Hadoop includes following modules:

- **Hadoop Core:** It is a common utility that supports other Hadoop modules.
- **Hadoop Distributed File System:** A distributed file system that is used for storing files on clusters and provides high throughput access to data.
- **HBase (Hadoop Database) :** A distributed database for random Read/Write access
- **YARN:** YARN stands for "Yet another resource negotiator". It is a framework for job scheduling and cluster resource management.
- **Pig (Programming Tool) :** A high level data processing system for parallel computing

2.2.1 Hadoop Distributed File System [13] [14] [15]

HDFS is a very large distributed file system that stores files as a series of block and replicate them to provide fault tolerance. The application data is stored on "Data Node". However, the metadata is stored on a distributed server, called "Name Node"

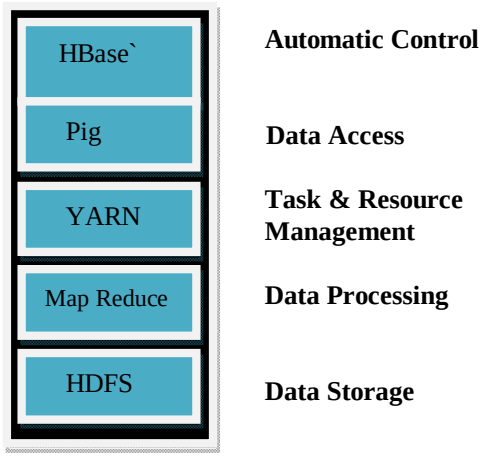

Fig 2 Hadoop Architecture

Name Node: Name Node manages all metadata and file system actions. It handles the file system namespace operations like open, close, and rename for both file and directory It also makes all decisions regarding replication of blocks. NameNode maintains the tree of namespace and maps the file blocks to DataNode (i.e. the physical location of file's data).

Data Node: Data node stores the data on the behalf of local or remote clients in the Hadoop file system. The process begins with a handshake which is performed between NameNode & DataNode in order to verify the namespace ID and software version of Data Node. If the credentials do not match, the DataNode shuts down. Otherwise, the Data Node registers with NameNode & stores their unique storage ID's, which are assigned when they first register and then not changed later. After this, the DataNode service the read, write & replication request based on directions from Name Node

Job Tracker: The job tracker helps in individual maps & reduces processes. It tracks the success and failure of these tasks. If a task is failed, it is launched again on another node for a limited no of tries.

Task Tracker: The task tracker manages the execution of individual task on each slave node. They also work in parallel using multiple java Virtual Machines. They also transmit a heartbeat message to notify the Job Tracker that it is still alive.

2.2.2 Map Reduce

In Map Reduce in [16], the first step 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 smaller tasks (key value pair). The reduce jobs then takes the output from a map as input and combine these into a smaller set of results. The map function can run independently on each key al pair, allowing parallelism. Reduce function can also implement on each intermediate key. It is a linearly scalable programming model. Mathematically it can be explained as a one to one mapping of Key Value pair

$$\text{Map: } (k1, v1) \longrightarrow [(k2, v2)]$$

$$\text{Reduce: } (k2, [v2]) \longrightarrow [(k1, v1)]$$

2.2.3 YARN [17]

YARN was born as a need to enable a broader way of interaction patterns for data stored in HDFS beyond the Map Reduce kind of framework.

a) It is more general processing platform not constraint to Map Reduce.
b) YARN enhances the power of Hadoop, compute cluster not limited to Map Reduce. Thus, it provides great processing power and good scalability.
c) Improved cluster utilisation according to criteria such as capacity, guarantee, fairness, service level agreements, supports other workloads (Graph processing, machine learning, iterative processing)
d) YARN allows multiple access engines either open source or prepared for Hadoop, to use for batch, interactive and real time data

2.2.4 PIG [20], [22] [22]

a) PIG is a high level programming on top of Hadoop Map Reduce
b) It can do all required manipulation in Hadoop
c) It uses java python & ruby and connects to other utilities.
d) It tackles real problems
e) Ex ETL model

2.2.5 HBase [18] [19]

a) HBase is a distributed fault tolerant, highly scalable, no-SQL distributed database build on top of Hadoop Distributed File System
b) HBase contains the information about location of data available in HDFS
c) It is column oriented database management systems
d) It provides scalable and high performance platform for dealing with heterogeneous data.
e) Data is stored in a key value pair structure
f) HBase adopts master/ slave architecture. Master is responsible for assigning regions to slave and also to recover from slave failures. Whereas, slave is responsible for managing client request for reading and writing.
g) Data is stored based on Google's Big table concept
h) It can hold extremely large amount of data

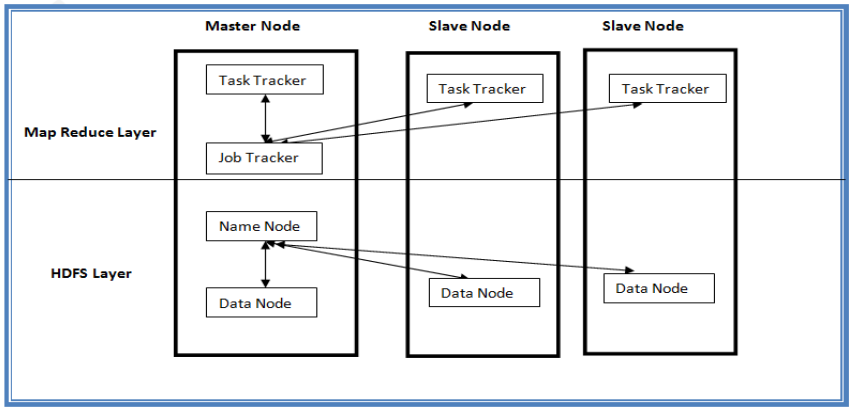

Fig 3 Architecture of Hadoop Distributed File System

Other Components
Apache hive [21], [23] is data warehouse software that facilitates cleaning and managing large data set residing in distributed file system. It provides a mechanism to project structure on all of this data and allows using sql like queries to access data that has been stored

2.2.6 Apache Sqoop [24]

a) Sqoop stands for SQL to hadoop
b) It is a tool designated for efficiently transferring bulk data between Hadoop and structured data store
c) It is a command line tool, import individual table or entire database into HDFS and generate java classes to allow user to interact and import data
d) It also provides ability to import sql database straight into hive data warehouse which is in HDFS system
e) It supports parallelism of tasks and thus fast performance

2.2.7 Oosie [25]

a) Oosie is a work flow schedule system that manages all Apache Hadoop jobs
b) Oosie workflow jobs are also called DAGs
c) Oosie coordinated jobs are triggered by frequency or data availability. It is integrated with rest of Hadoop stack supporting rest of the Hadoop jobs. Example : java Map Reduce, streaming Map Reduce, Pig Hive, Sqoop or others

2.2.8 Zookeeper [26] [27]

`

a) It provides operational services for Hadoop cluster
b) It has a centralized service for maintaining configuration information , naming and providing distributed synchronization.
c) Zookeeper also provides distributed configuration service, synchronized service and naming registry for entire distributed system Distributed applications use Zookeeper to store and a medium to update important information on cluster itself

2.2.9 Flume [21]

a) Flume is a distributed, reliable and available service for efficiently collecting and moving large amount of data.
b) It has a simple and flexible architecture based on streaming data flows.
c) It is robust, fault tolerant, tuneable to enhance real ability mechanism, fail over and recovery that keeps the cluster safe and reliable.
d) It is usually required for collecting data from social media and log files from all the machine in the cluster

Its architecture contains three components:
- Flume Source: It collect the data from the external web servers which is to be moved
- Flume Channel: The channel keeps the data temporarily until it is consumed by the sink
- Flume Sink: It receives the data from the channel and store it into the file system such as GFS or HDFS

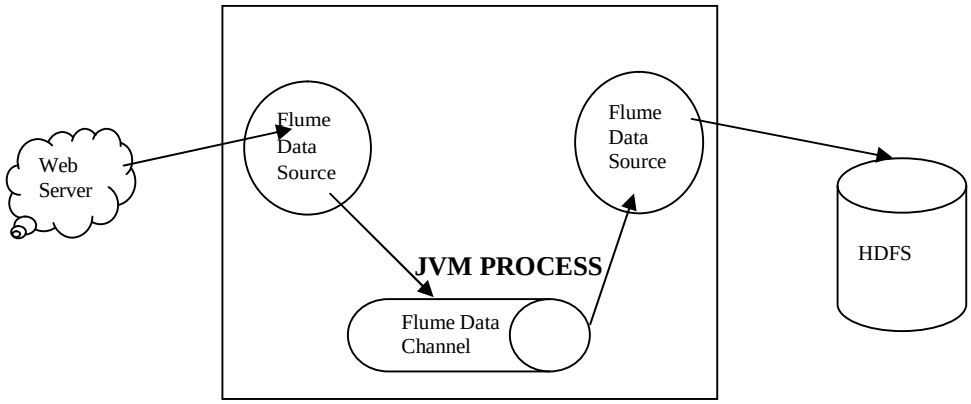

Fig 4. Working of Flume

2.2.10 IMPALA

a) IMPALA is a scalable parallel database technology on top of Hadoop stack.
b) It is designed to bring scalable parallel database technology to Hadoop and allows user to submit low latency queries to the data which is stored within the HDFS or HBase without requiring data movement or manipulation.
c) It allows to submit sql like queries with very less latency

2.2.11 SPARK

a) SPARK is fast and general engine for large scale data processing.
b) It provides Scalable data analytics platform for in memory computing and therefore allowing performance advantages over traditional Hadoop cluster system approach
c) It also implements and support SCALA language and provide unique environment for data processing. It is great for complex analytics and machine learning analytics

3 Hadoop Implementation

With the advent of Social web and Semantic web, more and more data in the form of images, audio, video, text etc which further increased the problem of extracting actual content.
This problem of handling large data can only be solved with the implementation of new models for storing and computing. Many of such models have been developed in the recent past. Big Table implementation by Google [28], dyanamo by Amazon[29], Linkedin[30], Apache casandra[31] and many other[32]. These organizations have also customised the standardized Hadoop architecture according to their work process and requirements. IBM, Joost, New York timesPower set, veoh are among the others which are using Hadoop
Hadoop is used for different purposes by different organization :

1. Search : Yahoo, Amazon
2. Log Processing : Facebook, Yahoo
3. Recommendation System ; Facebook
4. Data Warehouse : Facebook
5. Video & Image Analysis : New York Times

The modified Hadoop Stack used by big organizations are explained as follows :

3.3 Google Stack

- Roots of Hadoop go back to Big Tables and different kinds of applications that revolve around the idea of how do we scale and distribute very large amount of data
- Google file system : It distribute large amount into cheap storage and try to put a lot of data and come up with framework which can process the data
- Mysql gateway : It is able to access data in sql like language . To ingest data in Map Reduce cluster and abele to correlates some of the data as well
- Sawzell : High level specific language to access Map Reduce in cluster and submit some of the jobs
- Evenflow : It allows to chain together complex workload and coordinate services and events across the framework or specific cluster they have at that time.
- Drammel : A columnar storage and a meta data manager that allows to manage data and process very large amount of unstructured data
- Chubby :Chubby is a coordination system that would manage all of products in one ecosystem that will process the large amount of data seamlessly

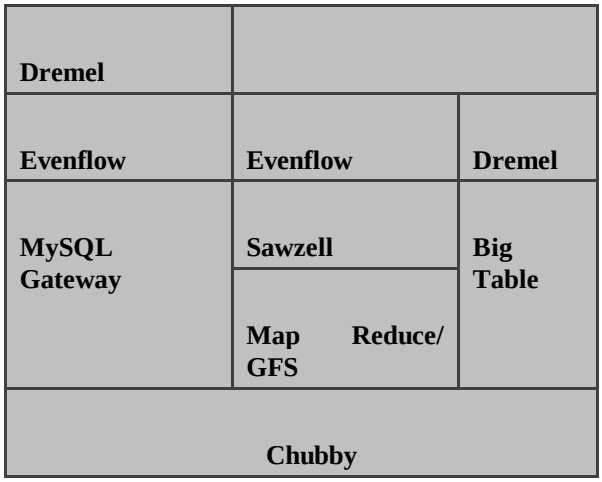

Fig 5 Google's Hadoop Stack

4 Processing Frameworks in Hadoop

Processing frameworks in Hadoop can be broadly classified

4.2 General-purpose processing frameworks[33]

These frameworks are batch frameworks which follow different programming model and the data is processed in Hadoop using low level API. It is useful for the cluster because all other framework only solve a specific problem and not generalized i.e. only work of a few set of use cases.

Example :MapReduce, Spark, and Tez

Map Reduce is the most defined framework but it is on contrary slowest too.

Spark and Tez are both DAG frameworks and don't have the overhead of always running a Map followed by a Reduce job; both are more flexible than Map Reduce.

4.3 Abstraction frameworks

These frameworks are build on top of General purpose processing framework and allow users to process data using higher level of abstraction. For example :

a. Pig is an abstraction framework that can run on MapReduce, Spark, or Tez.
b. Apache Crunch provides a higher level API used to run MapReduce or Spark jobs.
c. Cascading is an abstraction framework that can run on MapReduce or Tez

.

4.4 SQL frameworks

These frameworks are built on top of general purpose framework. They allow the users to run queries in Hadoop using SQL. This framework has its high demand and various options and categories. Example :

a. Hive can run on top of MapReduce or Tez [34]
b. Special-purpose SQL engines includes Impala, Presto and Apache Drill.

4.5 Graph processing frameworks —

These frameworks are also build on top of general purpose framework. They allow the capabilities of graph processing to be present in Hadoop. Example: Giraph, GraphLab.[35], [36]

a. Apache Giraph is a library that runs on top of MapReduce.
b. GraphX is a library for graph processing on Spark.
c. GraphLab was a stand-alone, special-purpose graph processing framework that can handle tabular data.

4.6 Machine learning frameworks

Build on top of general purpose framework, these frameworks enable machine learning analysis on Hadoop data. Mahout, MLlib, Oryx, and H2O are commonly used machine learning frameworks.

a. Mahout is a library on top of MapReduce,
b. MLlib is a machine learning library for Spark.
c. Oryx and H2O are stand-alone, special-purpose machine learning engines.

4.7 Real-time/streaming frameworks[37]

These frameworks provide near real-time processing for data in the Hadoop ecosystem. They can be built on top of a generic framework, such as Spark Streaming (on Spark), or as a stand-alone, special-purpose framework, such as Storm.

a. Spark Streaming is a library for doing micro-batch streaming analysis, built on top of Spark.
b. Apache Storm is a special-purpose, distributed, real-time computation engine with Trident used as an abstraction engine on top of it.

Frameworks that *do not* build on top of a general-purpose framework are called *special-purpose frameworks. Example:* Impala, Drill, and GraphLab.

The processing frameworks can also be classified based on their architecture
1. Engines: Those frameworks which have active components like a server are classified as engines. Example Hive
2. Libraries: Those frameworks which do not have any active components are classified as Libraries. Example MLib.

This distinction, however, does not impact end users. Users who need a solid machine learning framework usually don't care whether it's architecturally considered a library or an engine

5 Conclusion

It is very difficult to handle Big Data using traditional methods and techniques like RDBMS, and Data Mining. So a highly parallel software is used to handle Big Data. This paper includes all the necessary information about Haddop. Hadoop is an open source software for distributed computing of large database clusters. All its components are used to handle the Big Data. HDFS is the distributed file system which stores the data. Map Reduce is the programming paradigm which helps in computation. YARN is the resource manager which helps in scheduling and management of request and resource. Apart from these there are many supporting components which runs on top of these providing additional features. This Haddop stack is modified and used by organizations such as Google, Facebook and Linkedin according to their business requirements Many processing frameworks have also been implemented such as general purpose, abstraction, graph based etc. There are certain drawbacks and issues which needs to be considered in future for better analytics results.

References

[1]. Editorial: "Gathering clouds and a sequencing storm". Nature Biotechnology, Volume 28, no. 1, Jan 2010
[2]. Monya Baker, "Next-generation sequencing: adjusting to data overload",Nature Methods 20
[3]. Avita Katal, Mohammad Wazid and R H Goudar, "Big Data : Issues, Challenges, Tools and good Practices", in IEEE 2013
[4]. Dan Garlasu, "Data Implementation Based on Grid Computing",
[5].] Youseef M Essa, "Mobile Agent Based New Framework for improving Big Data Analysis", in International Conference on Cloud Computing and Big Data, IEEE, 2013
[6]. Sung Hwan Kim, Nam UK Kim and Tai Myoung Chung, "Attribute Relationship Evaluation Methodology for Big Data Security", in IEEE, 2013
[7]. Lam C, Warren J: Hadoop in Action. Manning Publications; 2010.
[8]. Venner J: Pro Hadoop. New York: A Press; 2009
[9]. J. Dean and S. Ghemawat. MapReduce: Simpli_ed data processing on large clusters. In Proc. OSDI, 2004.
[10]. Leons Petrazickis and Marius Butuc, "Crunching Big Data with Hadoop and BigInsights in the Cloud", pg 241-242, Information Management Technologies
[11]. Ronald C Taylor, "An overview of the Hadoop / MapReduce /HBase framework and its current applications in bioinformatics", 11th Annual Bioinformatics Open SourceConference (BOSC) 2010, Boston, MA, USA. July 2010
[12]. Hadoop: Open-source implementation of MapReduce. http://hadoop.apache.org
[13]. Parth Chandarana and M Vijayalakshmi, "Big Data Analytics Framework", in International Conference on Circuits, System, Communication and Information Technology Applications",IEEE, 2014
[14]. Edmund Kohlwey, Abel Sussman, Jason Trost and Amber Maurer, "Leveraging the cloud for Big data Biometrics", in World Congress in Services, IEEE,2011
[15]. Youseef M Essa, "Mobile Agent Based New Framework for improving Big Data Analysis", in International Conference on Cloud Computing and Big Data, IEEE, 2013
[16]. Dean, S Ghemawat, " Map reduce : A flexible data processing tool",Communications of the ACM , Vol 53, Number 1, pp 72-77, January 2010.

[17]. Vinod Kumar, Arun C Murthy, Chris Douglas, "Apache Hadoop YARN : Yet Another Resource Negotiator", ACM, Oct 2013
[18]. Mehul Nalin Vora, "Hadoop HBase fro large scale data", International Conference on Computer Science & Network Technology, IEEE, 2011
[19]. Dorin Carstoiu, Elena Lepadatu, Mihai Gaspar, "Hbase – non SQL Database, Performances Evaluation", International Journalof Advancements in Computing Technology Volume 2, Number 5, December 2010
[20]. Gates, A .F , Natkovich, O. Chopra, S Kamath, P Narayanmurty, & Srivastava, "Building a high level data flow system on top of Map Reduce : the Pig experience", in Proceeding VLDB
[21]. E. Laxmi Lydia & Dr. M. Ben Swarup, " Big data analysis using Hadoop components like Flume, Map Reduce, Pig and Hive", IJCSET, Vol 5, Issue 11, Nov 2015
[22]. Alan F. Gates, Olga Natkovich, Shubham Chopra, Pradeep Kamath, Shravan M. Narayanamurthy, Christopher Olston, Benjamin Reed,Santhosh Srinivasan, Utkarsh Srivastava, "Building a High LevelDataflow Systemon top of MapReduce:The Pig Experience", VLDB Endowment, ACM, 2009
[23]. Thusoo, A,.Sarma, J.,Jain, P. Murthy, "Hive : A warehousing solution over a Ma Reduce framework", Proceedings of the VLDB endowment, 2009
[24]. Mr S. S Aravinth & Ms. A Hasenah Begam, "An efficient Hadoop FrameworkSqoop and Ambari for Big Data Prcessing", International Journal of innovative research and in Sience and Technology, Vol 1, Isssue 10, March 2015
[25]. Mohammed Islam, Angelo K Huang, etal "Oozie : Toward a scalable workflow management system for Hadoop". ACM, 2012
[26]. Hunt, p Konar, M Junqueira, "Zookeeper : Wait free coordination for Internet scale system", Proceeding USENIX Annual Technical Conference, 2010
[27]. Patrick Hunt, Mahadev Konar, Flavio P. Junqueira and Banjanin Reed, "Zookeeper : Wait free coordination for Internet scale systems", USENIX Annual Technical Conference, 2012
[28]. Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach Mike Burrows, Tushar Chandra, Andrew Fikes, Robert E. Gruber, "Bigtable: A Distributed StorageSystem for Structured Data", Seventh Symposium on Operating System Design and Implementation (OSDI), Seattle, WA: Usenix Association; November 2006.
[29]. Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lakshman, Alex Pilchin, Swami Sivasubramanian, Peter Vosshall and Werner Vogels, "Dynamo: Amazon's Highly Available Key-Value Store", in the Proceedings of the 21st ACM Symposium on Operating Systems Principles, Stevenson, WA, October 2007.
[30]. Linkdln Project Voldemort Home page - http://projectvoldemort.com/
[31]. Apache Cassendra Homepage - http://cassandra.apache.org
[32]. Mark Grover, Ted Malaska, and Jonathan Seidman, " Hadoop Application Architecture", in O'Really Media, June 2015
[33]. Linning Zhu, Xiiwei Xu, Sherif Sakr and Daniel Sun, " A pipeline framework for heterogenous execution environment for Big Data Processing", in IEEE, 2016-05-02
[34]. Ashish Thusoo, Nmit Jain, Zheng Shao and Joydeep Sen Sharma, "Hive : A warehousing solution over a Map Reduce framework", in Proceeding of VLDB Endowement, Volume 2, Issue 2, August 2012
[35]. Yucheng Low, "Distributed GraphLab: a Framework for machine learning and data mining in cloud", in ACM, 2012
[36]. Yucheng Low, Joseph E Gonzalez and Appo Kyrola, " Graphlab : A new framework for parallel machine learning", in ACM 2014
[37]. V S Agneshwaran, "Big Data Analytics Beeyond Hadoop : Real time applications with Storm, Spark and more hadoopp alternatives", in FT Press 2014-15
[38]. Sanjay Ghemawat, Howard Gobioff & Shun Tak Leung, "The Google File System", 19th ACM Symposium on Operating Systems Principles, October 2003 12
[39]. Jeffery Dean & Sanjay Ghemawat, "Map Reduce : Simplified Data Processing on Large Clusters", 6th Symposium on Operating System Design and Implementation, Dec 2004