International Transaction of Electrical and Computer Engineers System, 2017, Vol. 4, No. 1, 26-38
Available online at http://pubs.sciepub.com/iteces/4/1/4

DOI:10.12691/iteces-4-1-4

A Perusal of Big Data Classification and Hadoop Technology

Nikhat Akhtar[1,*], Firoj Parwej[2], Yusuf Perwej[3]

[1]DDepartment of Computer Science & Engineering, Babu Banarasi Das University, Lucknow, India
[2]Department of Computer Science & Engineering, Singhania University, Distt. Jhunjhunu, Rajasthan, India
[3]Department of Information Technology, Al Baha University, Al Baha, Kingdom of Saudi Arabia (KSA)
*Corresponding author: dr.nikhatakhtar@gmail.com

Abstract Big Data make conversant with novel technology, skills and processes to your information architecture and the people that operate, design, and utilization them. The big data delineate a holistic information management contrivance that comprise and integrates numerous new types of data and data management together conventional data. The Hadoop is an unlocked source software framework licensed under the Apache Software Foundation, render for supporting data profound applications running on huge grids and clusters, to proffer scalable, credible, and distributed computing. This is invented to scale up from single servers to thousands of machines, every proposition local computation and storage. In this paper, we have endeavored to converse about on the taxonomy for big data and Hadoop technology. Eventually, the big data technologies are necessary in providing more actual analysis, which may leadership to more concrete decision-making consequence in greater operational capacity, cost deficiency, and detect risks for the business. In this paper, we are converse about the taxonomy of the big data and components of Hadoop.

Keywords: *Big Data, storage infrastructure, hama, avro, visualization, data domains, Hadoop, JobTracker, YARN*

Cite This Article: Nikhat Akhtar, Firoj Parwej, and Yusuf Perwej, "A Perusal of Big Data Classification and Hadoop Technology." *International Transaction of Electrical and Computer Engineers System*, vol. 4, no. 1 (2017): 26-38. doi: 10.12691/iteces-4-1-4.

1. Introduction

Since last three or four years, the field of "big data" has come into view as the new outskirts in the wide spectrum of IT-enabled renascence and [1] favorable time allowed by the information revolution. Big Data is used to delineate data which are very huge in size, which makes it arduous to analyze in conventional ways. The big data are enormously valuable to generate productivity in businesses and evolutionary accomplishment in scientific areas, which give us a great quantity of favorable times [2] to make substantial progresses in many fields. The company requirement to make sure of that new big data technology address their present business requirements to see their impressibility. Big Data is not approximately handling huge data sets and distributed computing, it also acts for a perfect example of a shift in the way data is managed and analyzed. When storing big data using eternal [3] storage can be costly. Next the Hadoop is building nearly commodity hardware. Consequently, it can provide justly huge storage for an appropriate cost. The Hadoop technology has been used in the field in Peta byte dimension. Hadoop [4] is an open-source framework for distributed computing that entitle the processing of volumetric data sets via horizontal scalability. Hadoop endows storage at a suitable price, this category of data can be captured and stored. Hadoop, not only confer distributed storage, on the other hand distributed processing as well. In this paper, we first introduce the general background the taxonomy of the big data like storage infrastructure, analytics, visualization, [5] data domains, compute infrastructure, security and privacy and finally, we review the related technologies, such as Hadoop.

2. The Storage Infrastructure

The big data storage is concerned with storing and managing data in a scalable way, persuade the necessity of applications that require access to the data. The perfect big data storage system would allow storage of a virtually countless amount of data, and high rates of unsystematic write and read access, flexibly and proficiently deal with a range of various data models, endorsement both structured and unstructured data. At this time state-of-the-art in data storage technologies that are capable of handling huge amounts of data, and identifies the data store related trends. Now we are discussing differing types of storage systems that are used for big data.

2.1. Distributed File Systems

The Hadoop File System (HDFS) offers the efficiency to store huge amounts of unstructured data in a reliable way on commodity hardware. In spite of there are file

systems with preferable performance, HDFS is [6] an integral part of the Hadoop framework and has already reached the level of a beneficial standard. It has been designed for huge data files and is well suited for hastily ingesting data and agglomeration processing.

2.2. NoSQL Databases

It is the most important family of big data storage technologies are NoSQL database management systems. The NoSQL databases use data models from outside the relational world that do not inevitably adhere to the transactional properties of atomicity, consistency, isolation, and durability (ACID).

2.3. NewSQL Databases

A contemporary form of relational databases that objective for comparable scalability as NoSQL databases while maintaining the transactional guarantees made by conventional database systems.

2.4. Big Data Querying Platforms

This technic that provides query mask in front of big data stores like as distributed file systems or NoSQL databases. The main anxiety is providing a high-level interface, e.g. via SQL3 like query languages and obtain low query latencies.

2.5. Database All-in-One Machine

It clarifies the complexity of deploying and managing the infrastructure of data centers, resolve the issue of continuous expanding of basic hardware resources at the age of big data, the need of all-in-one machine, and the storage cost of mass data. International manufacturers, Oracle, EMC, IBM launch integration products and solutions for big data. Database all-in-one machine is commonly convenient for data model of complex [7] storage relations. At the moment, computing needs high transactionality and consistency. Database all-in-one machine take possession of fully distributed big data processing architecture, integrating the hardware and software in a system.

3. The Analytics

The Big data analysis provides making “sense” out of huge volumes of multifarious data that in its raw form deficiency a data model to define what every element means in the context of the others. The Big Data analytics provide tools and methodologies that objective to transform enormous quantities of raw data into the data about the data for analytical intention. It is influential algorithms that are able to detect patterns, trends, and correlations over different time horizons in the data, but also on advanced visualization techniques as “sense-making tools”. As soon as trained algorithms can help make predictions that can be used to [8] detect anomalies in the form of huge deviations from the required trends or correlation in the data. There are five key methods for analyzing big data and generating insight.

3.1. Discovery Tools

The discovery tools are useful throughout the information lifecycle for intense, intuitive observation and analysis of information from any combination of unstructured and structured sources. These tools allow analysis side by side traditional BI source systems. Therefore, there is no necessity for up-front modeling, users can draw new discernment, come to meaningful conclusions, and make informed decisions faster.

3.2. BI Tools

BI tools are vital for reporting, analysis and performance management, in the first instance with transactional data from data warehouses and production information systems. BI Tools provide extensive capabilities for business intelligence and performance management, ad-hoc analysis, scorecards, including enterprise reporting, dashboards, enterprise scale platform.

3.3. In-Database Analytics

In-Database Analytics include a heterogeneity of techniques for discovering patterns and relationships in your data. In as much as these techniques are applied directly within the database, you remove data movement to and from other analytical servers, which accelerates information cycle times and lower total cost of proprietorship.

3.4. Decision Management

The Decision Management provides predictive modeling, business rules, and self-learning to take informed expedition based on the present context. This type of analysis enables individual recommendations across many channels, maximizing the value of each client interaction.

4. The Visualization

Data visualization is acting for data in some orderly form, including attributes and variables for the unit of information [9]. Visualization is a mental image or a visual act for an object or scene or person or abstraction that is alike to visual perception. Visualization has numerous definition, but the most designated one, which is found in literature is “the use of computer-supported, interactive, visual depiction [10] of data to amplify cognition”, where cognition means the ability of human perception or in simple words use of knowledge or the acquisition. The extension of conventional visualization approaches have already been become visible, but far from enough. In huge-scale data visualization, several scientist use feature extraction [11] and geometric modeling to most decrease the data size before authentic data rendering. The visualization is a graphical

representation that denotes the complicated ideas clearly, precisely, and expeditiously [12]. The most generally used big data visualization techniques can be broadly classified into the following three types.

4.1. Spatial Layout Visualization

This class of visualization techniques mentions to formulations that distinctive map a data object to a specific point in the coordinate space. The incipient motivation of such techniques is the cognitive ability of humans to effortlessly interpret information organized as a spatial substrate. The normally used spatial layout visualization techniques include bar charts, scatter plots, line charts, etc. However, these graphics are habitually limited by their inability to visualize complex relationships in data.

4.2. Abstract/Summary Visualization

Usually big data analytics require data to be processed at scale before any meaningful correlations can be searched. The scaling existing visualization techniques at this level becomes a non-trivial work. A current class of visualization techniques has been proposed recently that process and abstract or summarize such huge-scale data before rendering it to visualization routines. These techniques topple under Interactive visualization.

4.3. Interactive/Real-Time Visualization

This is a more recent class of techniques topple under interactive visualization that have to adapt to user interactions in actual time. Above-mentioned techniques necessitate that even complex visualization mechanisms take less than a second for a real-time navigation of data by a user. These techniques are quite strong in the sense that they allow users to expeditiously explore important insights in the data and prove or disprove various data science theories on top of such insights. Above-mentioned techniques are also crucial to industries that rely significantly on data-driven insights.

4.4. Other Visualization Tools

In this section we are discussing the other big data visualization tools.

• **Statistical Analysis System (SAS):** It uses intelligent auto charting to create optimal possible visual data we choose.

• **Polymaps:** It is a free Java script library for making interactive maps in present web browsers.

• **Flot:** It is a Java script plotting for JQuery. It endorsement visualization for interactive charts, zooming, data points and stacked charts.

• **Google Maps:** This tool empowers developers to build visual mapping programs for any website or application.

• **Microsoft Excel:** It is also platform for data analysis and creates visualization and act of data in the form of charts and graphs.

• **R-Project:** R is a free software atmosphere for statistical and graphics. It assistance in data manipulation, analysis and calculation.

• **Tableau Public:** It is an effortless and user-friendly tool to create an iterative data visualization rapidly and puts them on the website [13].

5. The Data Domains

The Big data technologies sort out the issue by allowing us to cost efficaciously capture and store many new types of data in their raw format, later allowing us to analyze these new forms in our analytic systems. There are three types of data we need to consider, structured, unstructured, and semi-structured.

5.1. The Structured Data

The term structured data normally refer to data that has a defined length and format for big data. It's structured because when you placed it in your relational database [14] system a structure was apply for it, therefore we know where it is, what it means, and how it relates to other pieces of data in there. Structured data is data that are divided into making consistent pieces that are identical [4] and accessible by both humans and computers. The granularity of these slices can range from an individual data point, such as a number, date or text to data that includes multiple individual data points. The sources of structured data are separate into two categories first machine-generated data and second human-generated data.

The Machine-generated structured data can include the following.

- **Sensor Data:** Examples include radio frequency ID tags, Global Positioning System data smart meters, and medical devices,.
- **Web Log Data:** When servers, applications, networks, and so on operate, they capture all variants of data about their activity. This can amount to large volumes of data that can be advantageous, for example, to deal with service-level accord or to predict security breaches.
- **Point-of-Sale Data:** When the cashier swipes the bar code of any item that you are buying, all that data related to the product are generated.

The Human-generated structured data can include the following.

- **Input Data:** This is any piece of data that a human might input into a computer, such as name, income, age, non-free-form survey responses, and so on. This data can be advantageous to comprehend basic customer behavior.
- **Click-Stream Data:** Data is procreate every time you click a link on a website. This data can be analyzed to determine clientele behavior and purchase patterns.
- **Gaming-Related Data:** Every move you make in a game can be recorded. This can be advantageous in discerning how end users move through a gaming portfolio.

5.2. Unstructured Data

The unstructured data include free text like as documents produced in your company, audio files, images

and videos, and some types of social media. If the object to be stored carries no tags and has no instituted schema, ontology, [4] glossary, or coherent organization this is unstructured. In spite of, in the same category as unstructured data there are numerous types of data that do have at least some organization. Unstructured data is ubiquitously. Actually, most individuals and organizations conduct their lives around unstructured data [15]. The sources of unstructured data are separate into two categories either machine generated or human generated.

Here are some examples of machine-generated unstructured data.

- **Satellite Images:** This contains climate data or the data that the government captures in its satellite surveillance imagery. Like Google Earth, and you get the picture.
- **Scientific Data:** This contains seismic imagery, atmospheric data, and high energy physics.
- **Photographs and Video:** This contains surveillance, security, and traffic video.
- **Radar or Sonar Data:** This contains vehicular, meteorological, and oceanographic seismic profiles.

The following list shows a few examples of human-generated unstructured data.

- **Text Internal to Your Company:** The consider of all the text within documents, logs, survey results, and e-mails. Enterprise information, in fact represents a huge percent of the text information in the world today.
- **Social Media Data:** This data are initiated from the social media platforms such as Twitter, LinkedIn, YouTube, Facebook, and Flickr.
- **Mobile Data:** This contains data such as text messages and location information.
- **Website Content:** This comes from any site delivering unstructured content, like Instagram, YouTube, and Flickr.

5.3. Semi-Structured Data

The line between semi-structured and unstructured data is a slight fuzzy. If the data [4] has any organizational structure or carries a tag (like XML extensible markup language used for documents on the web) then it is somewhat simple to organize and analyze, and it is more reachable for analysis may make it more valuable [14]. This type of data that comes into view to be unstructured, but in fact semi-structured include

- **Text:** The XML, email or electronic data reciprocate messages (EDI). These lacks formal structure, but do contain tags or a known structure that isolated semantic elements. Today social media sources, a hot topic for analysis, come in this category. Twitter, Flickr, Facebook, and others offer data access through an application programming interface (API).
- **Web Server Logs and Search Patterns:** An individual's journey through a web site, whether discovery, consuming content, or shopping records in elaboration in electronic web server logs.
- **Sensor Data:** There is a large explosion in the number of sensors generate streams of data all around us. In consideration of sensors as only being found in industrial control systems or major transportation systems. Presently this includes RFIDs, infrared and wireless technology, and GPS location signals among others.

6. The Compute Infrastructure

Infrastructure is the foundation of Big Data architecture. Possessing the proper tools for storing, processing and analyzing your data is vital in any Big Data project. In this section, we are closely examining Infrastructure approaches- what they are, how they work and what each outlook is best used for.

6.1. Hadoop

To recap, Hadoop is really an open-source framework for processing, storing and analyzing data. The basic principle behind Hadoop is rather than tackling one monolithic block of data all in one go, it's more believable to break [16] up & distribute data into numerous parts, allowing processing and analyzing of different parts at the same time. In faithfulness, Hadoop is a whole ecosystem of various products, hugely presided over by the Apache [4] Software Foundation. Hadoop is superior suited to high-throughput, in-depth analysis in retrospect, where the larger number or all of the data is harnessed.

6.2. HDFS

The Hadoop Distributed File System (HDFS) is a distributed file system that runs on standard or minimum end hardware. HDFS is a distributed file system that confers high-performance access to data across Hadoop clusters. HDFS has become a key tool for managing pools of big data and helpful big data analytics applications. The HDFS stores a huge amount of data placed across several machines, usually in the hundreds [17] and thousands of simultaneously connected nodes, and provides data reliability by replicating each data example as three different copies two in one group and one in another. These copies may be replaced in the event of unsuccess.

6.3. MapReduce

In MapReduce executes a wide range of analytic functions by analyzing datasets in parallel before decreasing the outcome. The Map job distributes a query to different nodes, [18] and the Reduce gathers the outcome and solution them into a single value.

6.4. Yarn

Yarn is a basic component of Hadoop, managing access to all resources in a cluster. Yarn brokers access to cluster compute resources on behalf of several applications, using selectable standard such as fairness or capacity, allowing for a more general-purpose experience. Yarn combines a central resource manager that accepts the way applications use Hadoop system resources with node manager agents that monitor the processing operations of particular cluster nodes. It can run on Linux and Windows.

6.5. Spark

Used on top of HDFS, and assurance speeds up to 100 times faster than the two-step MapReduce function in certain applications. That allows data to load in-memory and queried frequently, making it especially apt for machine learning algorithms

6.6. NoSQL

In NoSQL, which stands for Not Only SQL, is a term used to cover a range of various database technologies. This is relational predecessors, NoSQL databases are adept at processing dynamic, semi-structured data with minimum latency, making them better tailored to a Big Data environment. The NoSQL are commonly described as operational and analytical. NoSQL is preferable suited for operational tasks interactive workloads based on a selective standard where data can be processed in close in real-time. Other big names in NoSQL field include Apache Cassandra, Oracle NoSQL and MongoDB.

6.7. Massively Parallel Processing (MPP)

In MPP technologies, process vast amounts of data in parallel. The hundreds of processors, each with their own operating system and memory, work on various parts of the same program. MPP is a complicated process in need of a certain database function to [19] be shared between all involved processors. Messages are reciprocated between processors via an interconnection of data paths during MPP. An MPP system is considered superior than a symmetrically parallel system (SMP) for applications that allow a number of databases to be discovered in parallel. These contain decision support system and data warehouse applications. The supercomputers are also examples of MPP architecture.

6.8. Cloud

The cloud computing is the delivery of computing facility servers, databases, networking, software, storage, analytics and more over the Internet (the cloud). Cloud computing notifies to a broad set of products that are sold as a facility and delivered over a network. For example, in other infrastructural approaches when setting up your big architecture you necessity to buy hardware and software for each person involved with the processing and analyzing of your data. In cloud computing provide, your analysts only require access to application a web-based service where all of the essential resources and programs are hosted. The cloud computing also has a benefit in terms of delivering rapidly insights.

7. The Security and Privacy in Big Data

Today scenario the enormous amount of data being collected continues to rapidly grow, more and more companies are building big data repositories to gather, aggregate and extract meaning from their data. The main point of Big data is to access [20] data from several and various domains, security and privacy will play a necessary role in big data research and technology. We provide a brief description of security and privacy challenges in this section.

7.1. Protect Computations in Distributed Programming Frameworks

In distributed programming frameworks uses parallelism in computation and storage to process large amount of data. MapReduce is an example of the distributed programming framework. It is used [21] for data intensive computation in a huge cluster environment. MapReduce has become famous for analyzing the huge data sets. In the first stage of MapReduce, a Mapper for every chunk reads the data, performs some computation, and outputs a list of key value pairs. In the next stage, a Reducer combines the values related to each distinct key and outputs the outcome. The tasks which involve highly parallel computations over huge data sets are especially suited for MapReduce frameworks such as Hadoop. In spite of, the data mappers may contain intentional or unintentional leakages. The unbelief mappers could return incorrect results, which will in turn generate incorrect aggregate outcome.

7.2. Account Observe and Control

The challenging task to manage accounts for big data users. Need strong passwords, deactivate inactive accounts, and impose a maximum permitted number of failed log-in attempts to help stop attacks from getting access to a cluster. It's essential to [22] note that the enemy isn't always outside of the organization. The monitoring account access can assist decrease the probability of a successful compromise from the inside

7.3. Anonymization

The Anonymization is the procedure of altering and masking personal data in such a way that individuals cannot be re-identified and no information about them can be learned. This is a normal technique to do this is using hashing. A hashing algorithm maps the actual value (e.g. Person name) to a string of seemingly random characters. Hashing algorithms have the property that for the same input, they always have similar output. Furthermore, when you only know the output, you cannot retrieve back the input. This can remain both data utility and data privacy in the [23] datasets, because data, scientists will still be able to link various data records based on hash identifiers, meanwhile at the same time not directly being able to identify which person's information they are processing.

7.4. Protect Data Storage and Transactions Logs

They protect data storage and transaction logs are stored in multi-tiered storage devices. Manually moving data between tiers assistance the IT manager to control what data is moved and when. Furthermore, as the volume of data set continue to increase exponentially, scalability and availability have obligate auto-tiering for big data storage management. Auto tiering solutions does not

retain information about where the data is stored, which creates new challenges to protect data storage.

7.5. Protect Configurations for Software and Hardware

When make servers based on secure images for all systems in your organization's big data architecture. Make certain patching is up to date on these machines and that administrative privileges are limited to a miniature number of users. Utilization automation frameworks, such as Puppet, to automate system configuration and make sure that all big data servers in the enterprise are uniform and safe.

8. The Hadoop Technologies

Today circumstances, we live in the age of big data, where the data volumes we need to work with on a day-to-day basis have outgrown the storage and processing ability of a single host. It is not effortless to measure the total volume [2] of data stored electronically, but a recent analyst estimate puts the size of the "The Digital Universe" at 4.5 zettabytes in 2013. The same firm is forecasting a tenfold growth by 2020 to 46 zettabytes. Big data brings with it two basic challenge firstly how to store and work with voluminous data sizes, and secondly, how to comprehend data and turn it into a competitive advantage. In this context Hadoop fills a gap in the market by efficaciously storing and providing computational [4] capabilities for substantial amounts of data. It's a distributed system made up of a distributed filesystem, and it puts forward a way to parallelize and execute programs on a cluster of machines.

8.1. What is Hadoop?

The Apache Hadoop is an open-source software framework that confer massive data storage and distributed processing of huge amounts of data. The Hadoop confers the tools needed to develop and run software applications. The data is divided into blocks and [3] stored across several connected nodes (e.g. Computers) that work together. This setup is indicated to a cluster. A Hadoop cluster can span thousands of nodes. The Computations run in parallel across the cluster, which means that the work is split among the nodes in the cluster. The Hadoop framework is written in Java, it permits developers to deploy custom written programs coded in Java or any different language to process data in a parallel fashion across hundreds or thousands of [4] commodity servers. The complexity of the process and the volume of data, reaction time can vary from minutes to hours. Hadoop leverages a cluster of nodes to run MapReduce programs, comprehensively in [1] parallel. In MapReduce program consists of two steps, first the Map step processes input data and second Reduce step assembles intermediate results into a final outcome. Every cluster node has a local CPU and local file system on which to run the MapReduce programs. The data are fragmentary into data blocks, stored across the local files of various nodes, and replicated for credibility. The local files constitute the file system called Hadoop Distributed File System (HDFS). The number of nodes in every cluster transformation from hundreds of thousands of machines shown in Figure 1. Hadoop can also enjoin for a certain set of failover scenarios.

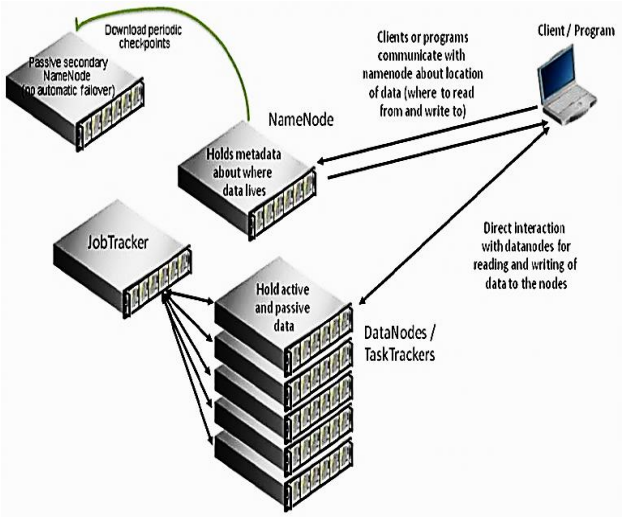

Figure 1. The Overview Of Hadoop Cluster

8.2. The Genesis of Hadoop

The Hadoop was created by Doug Cutting who had created the Apache Lucene (Text Search), which is originated in Apache Nutch (Open source search Engine). Hadoop is a part of the Apache Lucene Project. In fact Apache Nutch started in 2002 for working crawler and discover system. Nutch Architecture would not scale up to billions of pages on the web. Since 2003 Google had published Architecture called Google Distributed File System (GFS), which was resolved the storage requirement for the very huge files generated as a part of the web crawler and indexing process.

In 2004 based on GFS architecture Nutch was implementing open source called the Nutch Distributed File System (NDFS). In 2004 google was published Mapreduce, In 2005 Nutch developers had worked on Mapreduce in Nutch Project. The majority of Algorithms had been ported to run using Mapreduce and NDFS. In February 2006 they introduced Nutch to form an independent subproject of Lucene called Hadoop. At the same time, Doug Cutting joined Yahoo, which provided a dedicated team and the resources to turn Hadoop into a system that ran at web scale. This was exhibited in February 2008 when Yahoo declares that its production search index was being generated by a 10,000-core Hadoop cluster.

In January 2008, Hadoop was built up its own top-level project at Apache, confirming its success and its diverse, effectual community. Furthermore, Hadoop was being used by many other companies besides Yahoo, such as Facebook, and the *New York Times*. In April 2008, Hadoop broke a world record to become the intense system to sort a terabyte of data and running on a 910-node cluster, Hadoop sorted one terabyte in just 209 seconds, beating the previous year's winner of 297 seconds.

8.3. Benefits of Storing Data in Hadoop

The Hadoop accomplishes two tasks first massive data storage and second distributed processing. Hadoop is a low-cost alternative for data storage over conventional data storage options. Hadoop uses commodity hardware to credibly store huge quantities of data [4]. Other benefits if data and application processing are protected against hardware failure and node goes down, data are not vanished because a minimum of three copies of the data exists in the Hadoop cluster. Moreover, jobs are automatically redirected to working machines in the cluster. The distributed Hadoop model is conformation to effortlessly and economically scale up from single servers to thousands of nodes, each offering local computation and storage. Unlike conventional relational databases, you do not have to pre-process data prior to storing it in Hadoop. You can comfortably store unstructured data. Finally, you can use Hadoop to stage huge amounts of raw data for subsequent loading into an enterprise data warehouse or to create an analytical store for high-value agility like as advanced analytics, querying, and reporting.

8.4. The Building Blocks of Hadoop

The running Hadoop means running a set of daemons, or resident programs, on the multitudinous servers in your network. These daemons have conspicuous roles some exist only on one server, some exist across multitudinous servers. In this paper, we discuss each one and its role within Hadoop.

8.4.1. NameNode

The NameNode is the centerpiece of an HDFS file system and the master of HDFS that directs the slave DataNode daemons to carry out the small level I/O tasks. The NameNode is the bookkeeper of HDFS, it sustains track of how your files are fragmentary down into file blocks, which nodes store those blocks, and the collective health [24] of the distributed file system. The function of the NameNode is memory and I/O intensive. The server hosting the NameNode generally does not collect any consumer data or adhere any computations for a MapReduce program to less the workload on the machine. In other words the NameNode server doesn't double as a DataNode or a TaskTracker. The customer applications speak to the NameNode while they desire to locate a file, or they want to add, move, delete, and copy a file. The NameNode responds the prosperous requests by returning a list of episodic DataNode servers where the data lives. The NameNode is a single point of discomfiture for the HDFS Cluster. The HDFS is not currently an advanced level of availability system. When the NameNode toward a lower position, the file system goes offline.

8.4.2. DataNode

The DataNode stores data in the Hadoop File System. Every slave machine in your cluster will host a DataNode daemon to adhere the grunt work of the distributed filesystem reading and writing HDFS blocks to genuine files on the local file system. When you want to read or write an HDFS file, the file is split into blocks and the NameNode will tell your client which DataNode each block live in. Assume that your client communicates directly with the DataNode daemons to process the local files commensurate to the blocks. Moreover, a DataNode may communicate [24] with other DataNodes to replicate its data blocks for redundancy. The DataNodes are incessantly reporting to the NameNode shown in Figure 2. The DataNodes informs the NameNode of the blocks it's currently storing. Subsequently, that is mapping [25] thorough the DataNodes successive poll the NameNode to confer information concerning local modification as well as receive instructions to create, move, or erase blocks from the local disk.

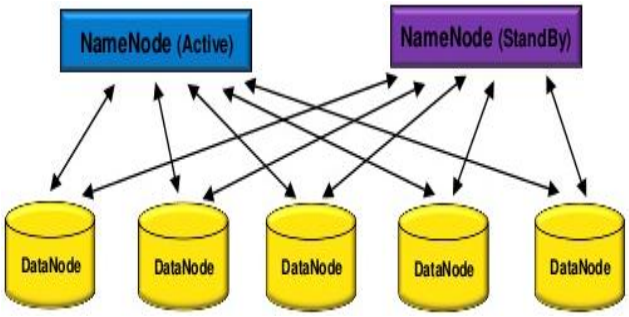

Figure 2. The NameNode

8.4.3. Secondary NameNode

Secondary Namenode is the most bewildering words for Hadoop beginner, people generally think that secondary Namenode is a replacement for Namenode when Namenode get decayed, but the truth is it is not, [26]. The working of secondary Namenode is apart and it is not a replacement for Namenode. Furthermore, Secondary NameNode (SNN) is an assistant daemon for monitoring the state of the cluster HDFS shown in Figure 3. The NameNode, each cluster has one SNN, and it generically resides on its own machine as well. The no DataNode or TaskTracker daemons run on the common server. The SNN distinguishes from the NameNode in that this process doesn't receive or record any real-time transformation to HDFS. Alternatively, [25] it communicates with the NameNode to take snapshots of the HDFS metadata at intervals defined by the cluster configuration [26]. At the moment we understood all Secondary Namenode does puts a checkpoint in a file system which will help Namenode to function superior. It's not the backup for the Namenode.

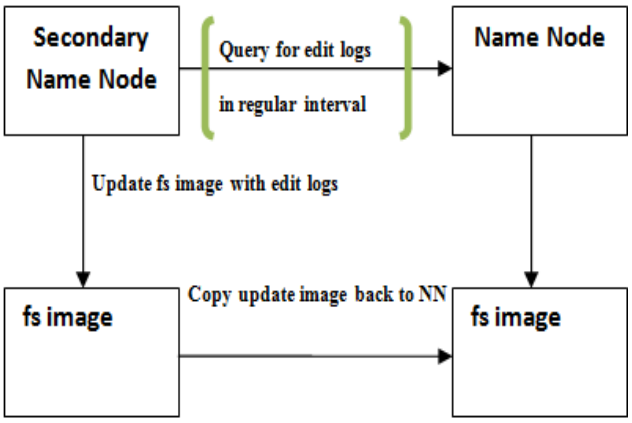

Figure 3. The Secondary NameNode

8.4.4. JobTracker

The JobTracker is the service within Hadoop that is accountable for taking client requests. It's allot them to TaskTrackers on DataNodes where the data required are locally present. Whether that is not possible, JobTracker attempt to allot the tasks to TaskTrackers within the same rack where the data is locally present. Whether for some reason this also lapse, again JobTracker allot the task to a TaskTracker where a replica of the data exists. One time you submit your code to [27] your cluster, the JobTracker determines the execution plan by determining which files to process, allot nodes to apart tasks, and monitors all tasks as they are running shows in Figure 4. In Hadoop, data blocks are replicated across DataNodes to make sure redundancy, so whether one node in the cluster fails, the job does not lapse as well. There is only one JobTracker daemon in pursuance of Hadoop cluster. It's normally run on a server as a master node of the cluster.

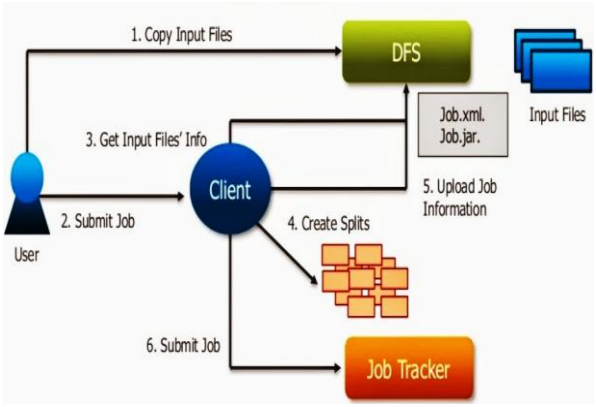

Figure 4. The JobTracker

8.4.5. TaskTracker

In TaskTracker is a node in the cluster that concede tasks Mapreduce and go together operations from a JobTracker. The TaskTrackers manage the execution of personal tasks on each slave node. Every TaskTracker is accountable for executing the individual tasks that the JobTracker allot. In spite of the fact that, there is a single TaskTracker per slave node and each TaskTracker can spawn several JVMs to handle many map or decrease tasks in parallel. The accountability of the TaskTracker is to persistently communicate with [27] the JobTracker. The TaskTracker sustain sending a heartbeat message to the JobTracker to inform that it is alive. Furthermore the JobTracker unsuccessfully to obtain a heartbeat from a TaskTracker within a particular period. The TaskTracker has destroyed and will re-adduce the commensurate tasks to additional nodes in the bunch.

8.4.6. Scheduler

In scheduler beginning and maintain jobs automatically by manipulating a prepared job control language algorithm or via communication with a human consumer. A scheduler in Hadoop is for sharing the cluster between various jobs, users for superior utilization of the cluster resources. Besides, without a scheduler a Hadoop job might consume all the resources in the cluster and other jobs have to wait for it to finish shown in Figure 5. Accompanied by scheduler jobs can execute in parallel consuming a part of the cluster. The Hadoop has a pluggable interface for schedulers. Entire implementations of the scheduler should extend the abstract class Task scheduler and the scheduler class should be specified in the Mapreduce, Jobtracker, Task scheduler property.

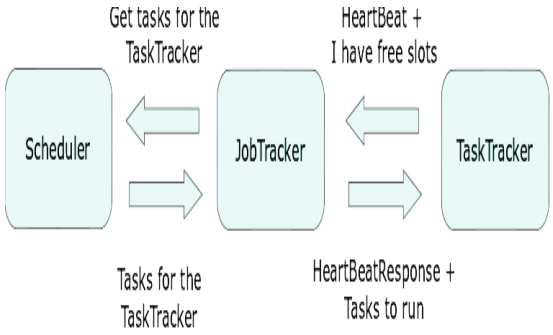

Figure 5. The Scheduler

8.5. The Data Cleaning of Hadoop

The Data Cleaner has this ability to help you with the quality of data, ingestion of data, standardizing and monitoring of data. We can leverage the computing power of your Hadoop cluster to vanquish infrastructure and performance hurdles. Data Cleaner can connect to the Hadoop Distributed File System (HDFS) and write, read your data, likewise any other file system. Additionally, you can submit your Data Cleaner jobs to in fact execute on the Hadoop cluster itself. Data Cleaner is a mighty tool for data profiling and for processing of huge data. Make sure that your assumptions about the data are accurate, before you spend a lot of time on the processing flow and copy data from and to various sources, be they big, relational, key-value based, small, document oriented or even file based. Data Cleaner can help you discover data matter, but also to extract value out the data you want to believe.

9. The Components of Hadoop

In this paper, we're going to explore the core components of Hadoop and find out their role and some of the more well-known and advantageous add-ons shown in Figure 6.

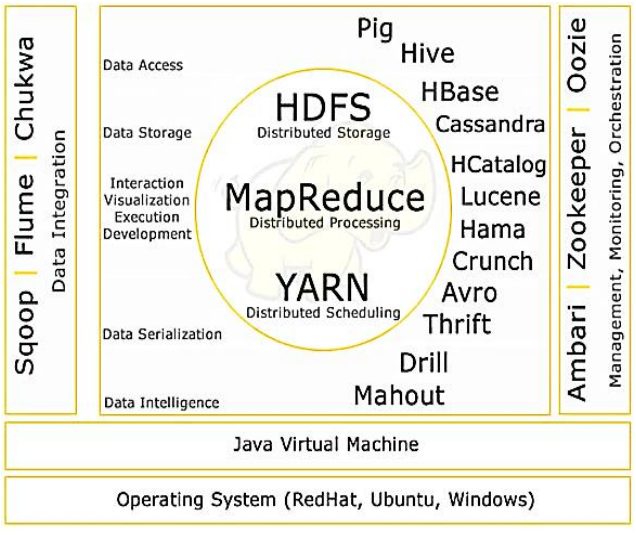

Figure 6. The Core Components of Hadoop

9.1. Hadoop Distributed File System (HDFS)

As you know that each physical system has its personal storage limit. When it comes to store huge volumes of data, then we may need more than one system, principally a network of systems. So that the data can be separate among several machines which are connected to each other via a network. This type of management in order to store huge volumes of data is known as a distributed file system. Hadoop has its personal distributed file system which is known as HDFS. This is the distributed file-system which stores data on the commodity machines [17]. This is the core of the Hadoop framework. This also endows a very high aggregate bandwidth across the cluster. The Java-based distributed file system that can store all variants of data without earlier organization. Throughout software upgrades [4] the possibility of corrupting the filesystem due to software bugs or human inaccuracy rise. The main intention of creating snapshots in HDFS is to reduce probable damage to the data stored in the system during upgrades.

The core of HDFS containing the two types of components. These primary two components are subsystems that run as separate processes. At the moment, we will just introduce the main accountability of these two components. The Namenode can be believed as the brain of HDFS shown in Figure 7. This component is aware how the directory structure looks like, how the access rights to each file and directory are configured, which users exists and it also be aware where each block of each file is stored. Entire this information is referred to as Namespace. The Datanodes store the blocks of the files you hold in HDFS. The standard size of one block is 64 MB per standard, but can be configured for every file at the time the file is created. Finally, the blocks end up as a general file on the local file system on one of the data nodes.

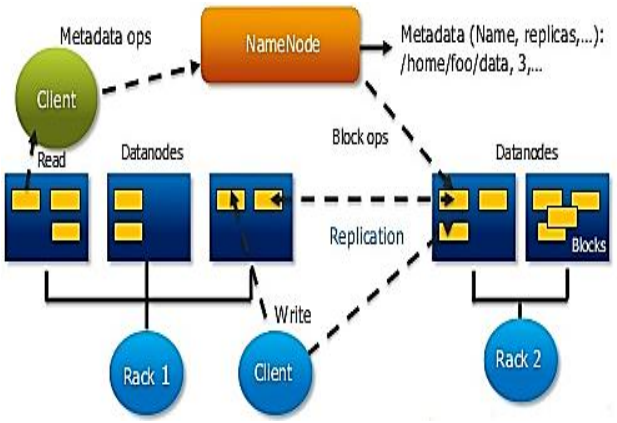

Figure 7. The Hadoop Distributed File System (HDFS)

9.2. Hadoop YARN

The Hadoop YARN (Yet Another Resource Negotiator) is a distinguished component of the open source Hadoop platform for big data analytics, licensed by the non-profit Apache software foundation. The vital components of Hadoop include a central library system, a Hadoop HDFS file handling system, and Hadoop, MapReduce, which is a batch data keeping resource. Therewith Hadoop YARN, which is described as a clustering platform that assistance to manage resources and schedule tasks. We belief the YARN involves setting up both ubiquitous and application specific resource management components. This aid to allocate resources to distinctive applications and manage other kinds of resource monitoring tasks. In YARN, an application submission client submits an application to the YARN resource manager. YARN schedules applications in order to concentrate on tasks and maintain big data analytics systems. YARN permits multiple access engines to use Hadoop as the common standard for batch, responsive and real-time engines that can concurrently access the same data set. YARN's dynamic allocation of bunch resources makes better usage over more stable MapReduce rules used in olden versions of Hadoop. The existing MapReduce applications developed for Hadoop 1 can run YARN without any disintegration to existing processes that previously work.

9.3. The Hadoop MapReduce

The Hadoop MapReduce provides a software infrastructure for effortlessly writing applications. This is processed in massive amounts of data in parallel on a large number of cluster commodity hardware in a fault-tolerant, credible manner. A MapReduce job normally partitioned the input data set into independent chunks which are processed by the map tasks in a perfectly parallel manner. The framework sorts the outputs of the maps, which are [18] then input to the decrease work. Generally, both the input and the output of the job are stored in a file system. The framework takes supervision of scheduling tasks, monitoring them and re-executes the irremeable tasks. Normally the compute nodes and the storage nodes are the same, that is, the MapReduce infrastructure and the Hadoop Distributed File System are running on the identical set of nodes shown in Figure 8. This configuration permits the infrastructure [2] to efficaciously schedule tasks on the nodes where the data is already present, achieve in very high aggregate bandwidth across the cluster. The term MapReduce in fact refers to two segregated and distinct tasks that Hadoop programs perform. The first is the map job, which occupy a set of data and metamorphose it into another set of data, where personal elements are split down into tuples. After that the second is the lower job occupy the output from a map as input and integrate those data tuples into a smaller set of tuples. As the sequence of the name MapReduce indicates, the lower job is consistently performed after the map job. The benefit of the MapReduce framework confers its cost effectiveness, ductility as well as scalability because of its inherent parallel processing architecture.

9.4. Pig

Pig is a procedural language for developing parallel processing applications for huge data sets in the Hadoop environment. The Pig is a substitute to Java programming for MapReduce, and automatically produce MapReduce functions. The Pig is normally used for complex use cases that expect multiple data operations. It is more of a processing language than a query language. Pig assistance, develop applications that aggregate and sort data and supports various inputs and exports. It is highly customizable, therefore users can write their own functions using their preferential scripting language.

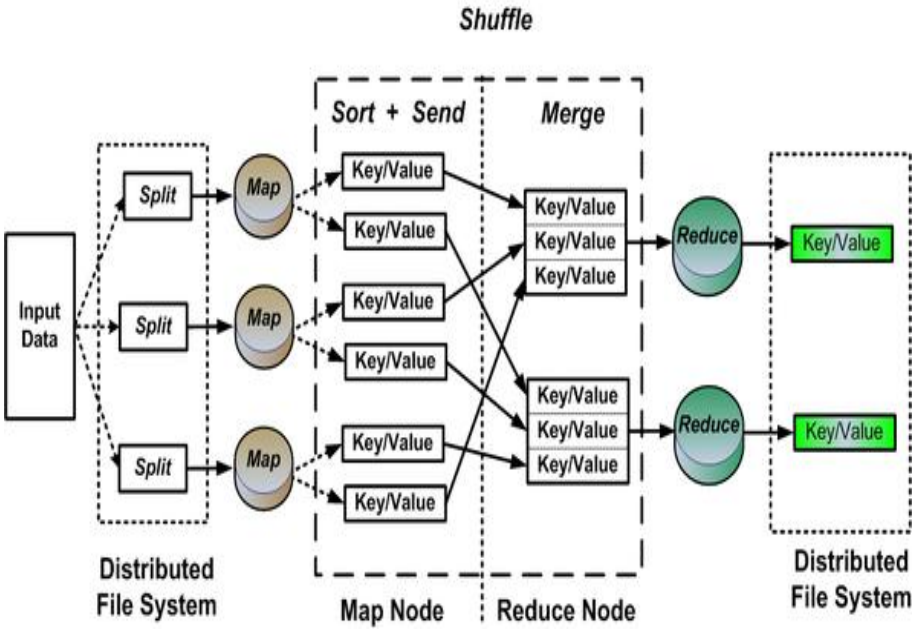

Figure 8. The MapReduce

9.5. Hive

In Hive is data warehousing software that addresses how data is structured and queried in distributed Hadoop clusters. Hive is also a famous development environment that is used to write queries for data about the Hadoop environment. It endows tools for ETL operations and brings some SQL-like potential to the environment. Hive is a declarative language that is used to develop applications for the Hadoop environment, although it does not support real-time queries.

9.6. HBase

The HBase is a distributed, dexterity, scalable, NoSQL database that sits atop the HFDS. It was invented to store structured data in tables that could have billions of rows and billions of columns. It has been deployed to power historical discovery through huge data sets, especially when the desired data is contained within a huge amount of insignificant or inconsequential data. HBase is not a relational [30] database and wasn't designed for endorsement transactional and other real-time applications. It is obtainable through a Java API and has ODBC and JDBC drivers.

9.7. Cassandra

The Cassandra is a liberated and open-source distributed NoSQL database management system designed to control huge amounts of data across many commodity servers, provide for high availability with no single point of negligence. In Cassandra is a distributed NoSQL database designed to manage large amounts of structured data across an array of commodity servers [28]. Cassandra boasts a distinctive architecture that delivers advanced distribution, linear scale performance, and is competent of keeping large amounts of data while providing sustained availability and uptime to thousands of concurrent users. Cassandra has been always up, always on, and produce very consistent performance in an imperfection environment. This makes Cassandra perfect for processing online workloads of a [4] transactional character, where Cassandra is dealing huge numbers of interactions and simultaneous traffic with each interaction yielding miniature amounts of data.

9.8. HCatalog

In HCatalog is a table and storage management service for Hadoop data and that presents a table abstraction so that you do not necessarily know where or how your data is stored. You can modify how you write data, while still supporting existing data in chronic formats. The HCatalog conceals additional layers around the Hive metadata store to provide an enlarged metadata service that includes functions for both Pig and MapReduce. Therefore HCatalog uses the Hive data model, you can use these functions to interact straightly with a Pig and MapReduce without translating the data type.

9.9. Lucene

This is high-performance text quest engine library that is written entirely in Java. When your quest within a collection of text, Lucene split the documents into text fields and builds an index from them. The index is the main component of Lucene that forms the basis of fast text quest potential. You use the explore methods within the Lucene libraries to discover text components.

9.10. Hama

The Apache Hama is a normal intention Bulk Synchronous Parallel (BSP) computing engine on top of Hadoop. It provides a parallel processing framework for comprehensive scientific and iterative algorithms. BSP is a convenient and flexible programming model, as compared with conventional models of message passing shown in Figure 9. Hama performs a series of perfect steps based on BSP. A perfect step consists of three phase

first local computations, second message communication, and third barrier synchronization [29]. Hama is appropriate for iterative computations since it is possible that input data which can be saved in memory is able to transfer between perfect steps. In spite of, MapReduce must scan the input data in each iteration, and then output data must be saved in the file system, like as HDFS. Consequently, Hama can resolve the problems which MapReduce cannot handle effortlessly. Furthermore, It provides not only real BSP programming model. This is also vertex and neuron basic programming models, influence by DistBelief and Google's Pregel.

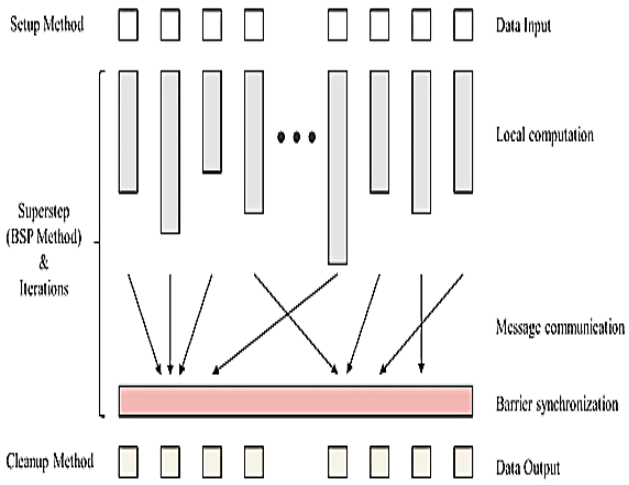

Figure 9. The MapReduceBulk Synchronous Parallel (BSP) Model

9.11. Crunch

The Crunch, a Java library that objective to make testing, writing, and running MapReduce pipelines convenient, efficient, and even fun. In Crunch design is modeled afterward Google's FlumeJava, focusing on a miniature set of normal primitive operations and ingenious user-defined functions that can be assorted to create arduous, multi-stage pipelines. Consequently, in runtime Crunch compiles the pipeline into a sequence of MapReduce jobs and mastermind their execution. Crunch adherence reading and writing data that are stored using Hadoop's writable format or Apache Avro records. The Crunch can be used to create the glue code that alters raw data into the structured input that a machine learning algorithm expects.

9.12. Avro

The Avro provides a remote procedure call and a data sequencing framework developed within Apache's Hadoop project. It was developed by Doug Cutting, the father of Hadoop. Avro becomes entirely helpful, as it deals with data formats that can be processed by various languages. Avro is a privileged tool to serialize data in Hadoop. Avro becomes language-independent schema is related to its read and write operations. Avro serializes the data into a concrete binary format, which can be deserialized by any application. Avro customer JSON format to declare the data structures. Nowadays, it supports languages such as Java, C, C++, C#, Python, and Ruby. Avro depends heavily on its schema. It allows every [31] data to be written with no prior knowledge of the schema. It serializes quickly and the outcome serialized data is lesser in size.

9.13. Thrift

The Apache Thrift software framework, provides for scalable cross language services development, concatenate a software stack with a code generation engine to build services. The Thrift work efficiently and seamlessly between C++, Erlang, Perl, Haskell, C#, Cocoa, Java, Python, PHP, Ruby, JavaScript, Smalltalk, OCaml and Delphi languages. Apache Thrift permits you to define data types and service interfaces in a simple definition file. That file as input, the compiler procreate code to be used to comfortably build RPC clients and servers that communicate seamlessly across programming languages.

9.14. Drill

Apache Drill is an open-source software framework that supports data-intensive distributed applications for interactive analysis of massive amounts of datasets. In Drill includes a distributed execution environment, purpose built for massive amounts of data processing. A Drillbit service can be installed and run on all of the expected node in a Hadoop cluster to form a distributed cluster environment. When a Drillbit runs on each data node in the cluster, Drill can maximize data locality during query execution without moving data over the network or between nodes. Drill uses ZooKeeper to preserve cluster membership and health-check information. The Drill endows a powerful distributed execution engine for processing queries. Users can submit requests to any node in the cluster. You can normally add new nodes to the cluster to scale for huge volumes of data, support more users or to make better performance.

9.15. Mahout

The Apache Mahout is also a project of the Apache Software Foundation to produce free implementations of distributed or scalable machine learning algorithms focused principally in the areas of collaborative filtering, clustering and classification. Mahout provides the data science tools to automatically discover meaningful patterns in those big data sets. The Apache Mahout project objective to make it faster and convenient to turn big data into big information.

9.16. Ambari

The Ambari main objective of making Hadoop administration effortless by developing software for provisioning, managing, and monitoring Apache Hadoop clusters. Ambari provides an instinctive, easy-to-use Hadoop management. Ambari decrease the complexity to administer and configure cluster security across the entire platform. Ambari is web-based tool for endorsement for Hadoop HDFS, Hadoop, HCatalog, HBase, ZooKeeper, MapReduce, Hive, Oozie, Pig, and Sqoop.

9.17. ZooKeeper

The ZooKeeper maintains common objects that are needed in large cluster environments, such as configuration information, distributed synchronization, and group services. It is centralized infrastructure and set of services that allow synchronization across a cluster. Many other open source projects that use Hadoop clusters require these cross-cluster services. Having these services available in ZooKeeper make sure that every project can embed ZooKeeper without having to build new synchronization services into each project.

9.18. Oozie

The Oozie is the workflow scheduler that was developed as part of the Apache Hadoop project. It manages how workflows start and execute, and also monitoring the execution path. In Oozie provides users with the capability to define actions and dependencies between actions. Oozie is scheduled actions to run when the required dependencies are met. Oozie is a server-based Java web application that uses workflow definitions written in hPDL, which is an XML Process Definition Language similar to JBOSS, JBPM, jPDL. Again Oozie only supports specific workflow types, so other workload schedulers are normally used instead of or in addition to Oozie in Hadoop environments.

9.19. Sqoop

The Sqoop as a front-end loader for big data. Sqoop is a command-line interface that make easier in moving bulk data from Hadoop into relational databases and other structured data stores. Using Sqoop replaces the necessity to develop scripts to export and import data. In common use case is to rehabilation data from an enterprise data warehouse to a Hadoop cluster for ETL processing. Functional ETL on the commodity Hadoop cluster is resource efficient, while Sqoop provides a practical transfer method.

9.20. Flume

The Flume is a distributed, credible, and available service for efficiently collecting, aggregating, and moving huge amounts of log data. It has a convenient and resilient architecture based on streaming data flows. It is powerful and fault tolerant with tunable credibility mechanisms and numerous failover and recovery mechanisms. It uses a simple, extensible data model that allows for online analytic application. Flume NG uses channel-based transactions to guarantee authentic message delivery. When a message moves from one agent to another, two transactions are started, first on the agent that delivers the event and the second on the agent that receives the event [30]. This ensures guaranteed delivery semantics Flume is used to log manufacturing operations. When one run of product comes off the line, it generates a log file about that run. Even if this occurs hundreds or thousands of times per day, the huge volume logs file data can stream. The Flume tool for identical day analysis with Apache Storm or months or years of production runs can be stored in HDFS. The analyzed by a quality assurance engineer using Apache Hive.

9.21. Chukwa

The Chukwa is a Hadoop subproject devoted to large-scale log collection and analysis. In Chukwa built on top of the Hadoop distributed file system (HDFS) and MapReduce framework and inherits Hadoop's scalability and robustness. In Chukwa also provide a dynamic and a robust toolkit for displaying monitoring and analyzing outcome. Again order to make the optimal use of this collected data. A Chukwa is structured as a pipeline of collection and processing stages, with clean and narrow interfaces between stages. This will make possible future innovation without breaking existing code.

10. Limitations of Hadoop

Hadoop is an impressive platform for processing massive volumes of data with remarkable speed on low-cost commodity hardware, but it does have some momentous limitations. For example, Hadoop, Distributed File System or HDFS cannot read miniature data files randomly. Consequently, Hadoop is not the optimal solution for organizations which deal with less amount of data. Next the time delay is more noticeable with huge data sets, which means it is less practicable for more scalable projects that require data to be analyzed in real-time. Another disadvantage Hadoop is also absent encryption at the storage and network levels. Which is a considerable selling point for government organization and others that choose to keep their data under conceal. This is the major obstacle is in terms of application. Again the MapReduce is appropriate for batch processing jobs [32]. It does not do well for graph, iterative, incremental and many other variant formats. However, so many of the models have been in particular built for these kinds of processing either on top of MapReduce or independently. Example for, Pregel, Incoop, S4, Haloop, CIEL, and many more. Another limitation Implementing iterative map alleviate jobs is expensive due to the large space using up by each job. The rack-aware capacity of Hadoop, while strong for their intended purpose, do not support scalable network topologies such as multidimensional toruses and meshes. They maintain Clos-style network topologies, period. Finally the framework of Hadoop is developed in Java, the programming language known for its fame to be the most vulnerable one among cyber criminals. It means Hadoop is completely vulnerable to data breaches automatic.

11. Conclusion

In the real world we have entered an era of big data which is the next frontier of competition, innovation, and productivity, a new wave of scientific revolution is about to begin. The big data are a field committed to the processing, analysis and storage of spacious collections of data that successive originate from disparate sources. In this paper, systematically investigates and analyzes the following aspects, storage infrastructure, analytics, data

domains, compute infrastructure, security and privacy, visualization in big data. These discussions aim to confer an extensive overview and big-picture to readers of this exciting area. Again, Hadoop can run from single servers to thousands of machines, every one provide local computation and storage. Furthermore, Hadoop makes it possible to run applications on systems with thousands of commodity hardware nodes, and controlled thousands of terabytes of data. Its distributed file system makes possible acute data transfer rates among various nodes and permit the system to sustain operating in case of a node lack of success. This paper explains the significance of the Hadoop technology which will give us the summary of the several components of Hadoop framework and also examine the what are the limitations of Hadoop. Finally, we hope this paper can confer researchers and practicing professionals with the state-of-the-art knowledge on Big data and Hadoop technology.

References

[1] Gandomi, A., & Haider, M. Beyond, "The hype: big data concepts, methods, and analytics," International Journal of Information Management, 35(2), 137-144, (2015).

[2] Heudecker, Nick. 2013. "Hype Cycle for Big Data." Gartner G00252431.

[3] B. Elser and A. Montresor, "An evaluation study of bigdata frameworks for graph processing," in Proc. IEEE Int. Conf. Big Data, Oct. 2013, pp. 6067.

[4] Yusuf Perwej, "An Experiential Study of the Big Data", International Transaction of Electrical and Computer Engineers System, ISSN (Print): 2373-1273, ISSN (Online) 2373-1281, USA, Vol. 4, No. 1, Page 14-25, 2017.

[5] F. H. Gebara, H. P. Hofstee, and K. J. Nowka, "Second-generation big data systems," Computer, vol. 48, no. 1, pp. 3641, 2015.

[6] K. Shvachko, H. Kuang, S. Radia, R. Chansle, "The Hadoop Distributed File System", Proceeding MSST 10 Proceedings of the 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST), Pages 1-10, May 03 - 07, 2010.

[7] Hinshaw F D, Meyers D L, Zane B M. Programmable streaming data processor for database appliance having multiple processing unit groups: US, US7577667, 2009.

[8] Bollier, David. The Promise and Peril of Big Data. The Aspen Institute, 2010.

[9] M. Khan, S.S. Khan, Data and Information Visualization Methods and Interactive Mechanisms: A Survey, International Journal of Computer Applications, 34(1), 2011, pp. 1-14.

[10] D. Tang, C. Stolte, P. Hanrahan, "Polaris: A System for Query Analysis and Visualization of Multidimensional Relational Databases", IEEE Trans. Visualization and Computer Graphics, vol. 8, no. 1, pp. 52-65, Jan.-Mar. 2002.

[11] S. Card, J. MacKinlay, and B. Shneiderman, (1998). "Readings in Information Visualization: Using Vision to Think". Morgan Kaufmann.

[12] Alfredo R. Teyseyre and Marcelo R. Campo, "An Overview of 3D Software Visualization", IEEE Transactions on Visualization and Computer Graphics, vol.15, No.1, 2009.

[13] C.L. P. Chen, C.-Y. Zhang, Data-intensive applications, challenges, techniques and technologies: A survey on Big Data, Information Sciences, 275 (10), pp. 314-347, 2014.

[14] J. Fan, F. Han, H. Liu, " Challenges of big data analysis", National Science Review, 1 (2) (2014), pp. 293-314.

[15] K. Bakshi, "Considerations for Big Data: Architecture and Approach", Aerospace Conference IEEE, Big Sky Montana, March 2012

[16] Basili, V.R., Carver, J.C., Cruzes, D., Hochstein, L.M., Hollingsworth, J.K., Shull, F. and Zelkowitz, M.V. 2008. Understanding the High-Performance-Computing Community: A Software Engineer's Perspective. IEEE Software

[17] Maltzahn, C., Molina- Estolano, E., Khurana, A., Nelson, A.J., Brandt, S.A. and Weil, S. 2010. Ceph as a scalable alternative to the Hadoop Distributed File System. login: The USENIX Magazine. (2010).

[18] Xu, C., Goldstone, R.J., Liu, Z., Chen, H., Neitzel, B. and Yu, W. 2015. Exploiting Analytics Shipping with Virtualized MapReduce on HPC Backend Storage Servers. IEEE Transactions on Parallel and Distributed Systems. PP, 99 (2015).

[19] Peng Hong, Du Nan. Research of parallel technology in massive commerce data management system. Application Research of Computers. Vol. 26 No. 2 Feb. 2009.

[20] T. Omer, P. Jules, "Big Data for All: Privacy and User Control in the Age of Analytics", Northwestern Journal of Technology and Intellectual Property, article 1, vol. 11, issue 5, 2013.

[21] A.A. Cardenas, P.K. Manadhata, S.P. Rajan, "Big Data Analytics for Security", IEEE Security & Privacy, vol. 11, issue 6, pp. 74-76, 2013.

[22] De Cristofaro, E., Soriente, C., Tsudik, G., & Williams, A. (2012). Hummingbird: Privacy at the time of twitter. In Security and Privacy (SP), 2012 IEEE Symposium on (pp. 285-299).

[23] Mohammadian E, Noferesti M, Jalili R. FAST: fast anonymization of big data streams. In: ACM proceedings of the 2014 international conference on big data science and computing, article 1. 2014.

[24] Konstantin Shvachko, Hairong Kuang, Sanjay Radia, Robert Chansler Yahoo!, "The Hadoop Distributed File System," IEEE NASA storage conference, May 2010.

[25] Cristina L. Abad, Huong Luu, Nathan Roberts, Kihwal Lee, Yi Lu, Roy H. Campbell, "Metadata Traces and Workload Models for Evaluating Big Storage Systems", Proceedings of IEEE 5th International Conference on Utility and Cloud Computing (UCC), pp. 125-132, 2012.

[26] Mohammad Asif Khan, Zulfiqar A. Memon, Sajid Khan, "Highly Available Hadoop NameNode Architecture", International Conference on Advanced Computer Science Applications and Technologies 2012, pp. 167-172, 2012.

[27] Jian Wan, Minggang Liu, Xixiang Hu "Dual-JT: Toward the high availability of JobTracker in Hadoop", Cloud Computing Technology and Science (CloudCom), IEEE 4th International Conference on, 2012.

[28] Rabl, Tilmann; Sadoghi, Mohammad; Jacobsen, Hans-Arno; Villamor, Sergio Gomez-; Mulero -, Victor Muntes; Mankovskii, Serge (2012-08-27). "Solving Big Data Challenges for Enterprise Application Performance Management" VLDB.

[29] Leslie G. Valiant, A bridging model for parallel computation, Communications of the ACM, Volume 33 Issue 8, Aug. 1990.

[30] W. Shang, Z. M. Jiang, H. Hemmati, B. Adams, A.E. Hassan, P. Martin, "Assisting developers of big data analytics applications when deploying on Hadoop clouds", the Proceeding of the international conference on software engineering, vol. 203, pp. 402-411, 2013.

[31] OnurSavas, YalinSagduyu, Julia Deng, and Jason Li, Tactical Big Data Analytics: Challenges, Use Cases and Solutions, Big Data Analytics Workshop in conjunction with ACM Sigmetrics 2013, June 21, 2013.

[32] Ke Wang, Ning Liu, Iman Sadooghi, "Overcoming Hadoop Scaling Limitations through Distributed Task Execution", Cluster Computing (CLUSTER), IEEE International Conference on, 2015.