ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

A Hadoop Framework for Addressing Big Data Challenges: A Survey

K.Srikanth[1], Dr. S.Zahoor-Ul-Huq[2], P.N.V.S. Pavan Kumar[3]

Assistant Professor, Dept. of CSE., GPREC, JNTUA University, Kurnool, Andhra Pradesh, India[1,3]

Professor, Dept. of CSE., GPREC, JNTUA University, Kurnool, Andhra Pradesh, India [2]

ABSTRACT: Today's technological advancements have led to a surge of data from diverse fields like health care, scientific sensors, user-generated data, Internet, supply chain systems and financial companies over last few years. Data becomes a powerful raw material for all interdisciplinary research areas and even for government and business performance. Now, the entire world employs the phrase "Big Data" to capture the meaning of evolving trend. As compared with traditional data, Big Data has its own characteristics like it is unstructured and need more time to analyze. To understand this unstructured data, new system architectures designed for data storage, transmission, acquisition and processing mechanisms. In this paper, our aim is to provide an overview of big data and its challenges for non-expert readers to find their unique solutions for big-data problems. Here, we present a framework to divide big data system in to four modules namely data generation, data acquisition, data storage and data analytics. Next, we present the Hadoop framework for facing big data challenges. Finally, we summarize different potential research directions for big data systems.

KEYWORDS: Big Data; stuructured data; unstructured data; challenges of big data; Hadoop framework

I. Introduction

In today's environment, Big data is one of the "smoldering" word used everywhere in the society. Influence of Big data is observed in every field like science, business, industry, government, society, etc. The process of big data includes collection, storage, transportation and exploitation. The core of big data processing is to collect, store, and transport which are important stages for data analytics. Big data can be effectively defined by using four V's – Volume, Velocity, Veracity and Variety. Volume represents the size of the data, which might be too big to be handled by the current algorithms. Velocity represents data streaming at rates faster than that can be handled by traditional algorithms. Veracity measures the quality of data that we cannot assume. The quality issues can be tackled either at data-processing stage or by using learning algorithm. Variety presents data of different types and modalities for a given object in consideration. These four V's are certainly not new, the machine learning and data mining researchers have been dealing with these issues for decades. The Internet based companies challenging many of the traditional process oriented companies, for which they now need to become knowledge-based companies driven by data rather than by process.

As we know that, present generation is living in a data surge era, where variety of data generated from different sources and its generation rate. For instance, an IDC report [1] predicts that, from 2005 to 2020, the global data rate will grow by a factor of 300, from 130 Exabyte's to 40,000 Exabyte's, representing a double growth every two years. The vast potential associated with big data has led to an emerging research field that has rapidly attracted incredible curiosity from various sectors, for example, industry, government and research community. [1] discusses on scalable big-data systems, which include a set of tools and mechanisms to load, extract, and improve data. Scalable big-data system faces a series of technical challenges, including:

- Due to the array of distinct data sources and the total volume, it is very difficult to accumulate and combine data with scalability from scattered locations. For instance, more than 175 million tweets containing text, image, video, social relationship are generated by millions of accounts distributed globally [3].
- The challenge of big data systems is to store and manage the collected huge and varied datasets and to provide quality and performance guarantee, in terms of fast retrieval, scalability, and privacy protection. For example, facebook requires storing, accessing and analyzing over 30 peta bytes of user collected data.
- Big data analytics must efficiently manage huge datasets at varied levels in real time – including modelling, visualization, prediction and optimization – such that to improve decision making and acquire further advantages.

DOI: 10.15680/IJIRCCE.2015. 0405095

However, traditional data management systems based on Relational DataBase Management Systems (RDBMS), are inadequate to face the challenges of big data mentioned earlier. Specifically, the difference between the traditional RDBMS and Big data falls into the following two aspects.

- From the perspective of data structure, RDBMSs can only support structured data, but offer modest support for semi-structured or unstructured data.
- From the perspective of scalability, RDBMSs scale up with expensive hardware and cannot scale out with commodity hardware in parallel, which is not suitable for ever growing data volume.

To deal with these challenges, the industry and research community have recommended various solutions for big data systems. Cloud computing can be set up as the infrastructure layer for big data systems to meet certain infrastructure requirements such as cost-effectiveness, elasticity and ability to scale up or down. To maintain persistent storage and the management of huge data, distributed file systems [27] and NoSQL [5] databases are suitable. In processing group-aggregation tasks, such as website ranking, MapReduce [6], a programming framework has achieved great success. To handle big data challenges, Hadoop [8] integrates data storage, data processing, system management and other modules to form a powerful system-level solution. Based on these innovative technologies and platforms, we can construct various big data applications.

II. PREAMBLE TO BIG DATA

A. *Definition:*

Basically, big data is not only a large volume of data but also other features that differentiate it from the concepts of "massive data" and "very large data". Of the various definitions of big data three types of definitions play a vital role in shaping how big data is viewed. IDC [9], big data technologies describe a new generation of technologies and architectures, designed to economically extract value from very large volumes of a wide variety of data, by enabling high-velocity capture, discovery, and/or analysis. This definition outlines the four salient features of big data, i.e., volume, velocity, veracity and variety. As a result, the "4Vs" definition has been widely used to describe big data. Mckinsey's report [2], big data is defined as "datasets whose size is beyond the ability of typical database software tools to capture, store, manage and analyze". This definition incorporates an evolutionary aspect in the definition of what a dataset must be considered as big data. NIST [28], "Big data is where the data volume, acquisition velocity, or data representation limits the ability to perform effective analysis using traditional relational approaches or requires the use of significant horizontal scaling for efficient processing". In fact, big data can be further classified into big data science and big data frameworks.

Big data science is the study of techniques covering the acquisition, conditioning and evaluation of big data. Big data frameworks are software libraries along with their associated algorithms that enable distributed processing and analysis of big data problems across clusters of computer units. An instantiation of one or more big data frameworks is known as big data infrastructure. The above definitions for big data provide a set of tools to compare the emerging big data with traditional data. The Table 1 lists the differences between big data and traditional data under the framework of the "4Vs".

Feature	Traditional Data	Big Data
Volume	GB	Constantly updated (PB or EB recently)
Generated rate	Per hour, day…	More rapid
Structure	Structured	Structured, semi-structured and un-structured
Data source	Centralized	Fully distributed
Data integration	Easy	Difficult
Data store	RDBMS	HDFS, NoSQL
Access	Interactive	Batch or near real-time

Table 1: Differences between big data and traditional data

B. *History:*

While discussing the history of Big Data the focus is on efficiently storing and managing bigger and bigger datasets, with size limitations increasing by orders of scaling. Based on each capability improvement, new database technologies were developed as shown in Fig. 1. Thus, the history of big data is divided into four stages.

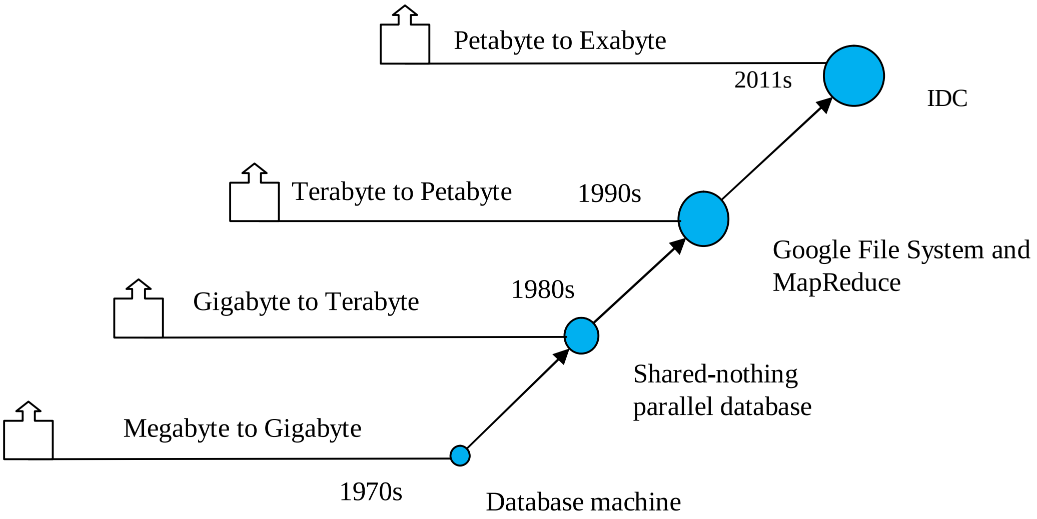

Fig. 1 History Big Data in to four stages

- In the 1970s and 1980s, business related data initiated the change in moving from megabyte to gigabyte sizes. The immediate solution at that point of time is to store that data and execute relational queries for business analyses and reporting. Database machine is the invention from research experts who integrated software and hardware to solve problems. Such type of integration will give better performance at lower cost was the analysis from the experts.
- In the late 1980s, the digital technology takes the step to generate data from gigabytes to terabytes which is beyond the storage capability of single large computer system. Data parallelization is the solution from research experts to enlarge the storage capacity and to improve performance by distributing data. Based on this idea, several types of parallel databases were built, which includes shared disk databases, shared memory databases and shared nothing databases. Among these three types of databases, the shared nothing architecture, built on a networked cluster of individual machines has observed a huge success.
- In the late 1990s, web 1.0 taken the opportunity to introduce the whole world into the internet era, which created huge semi-structured and unstructured data from terabytes to petabytes. However, parallel databases able to handle structured data well, but they offer little support for unstructured data. To face the challenge of web-scale data management and analysis, Google created Google File System (GFS) and MapReduce framework. They are able to perform automatic data parallelization and distribution of large-scale data to large clusters of servers. These systems are able to scale-up and scale-out and is therefore able to process unlimited data. To handle mixed type of huge data at runtime, NoSQL databases are introduced which are very fast, scalable and reliable.
- In coming years, the big companies are able to reach the next magnitude i.e., from petabytes to exabytes. Predicting the future, the big companies like EMC, Oracle, Microsoft, Google, Amazon and Facebook began to invest lot of money to do research in big data problems. This encouragement will make the researchers and academicians to develop innovative data management mechanisms and tools.

C. *Paradigms:*

i. **Streaming Processing:** This paradigm, the hypothesis is that the potential value of data depends on data freshness. Thus, this paradigm analyzes data as soon as it arrives to obtain the results and the data appears in a stream, in its continuous flow, because the stream is fast and carries massive size, only a little portion of the stream is stored in limited memory. One or few passes over the flow are made to obtain rough results. This type is used for online applications, commonly at the second, or even millisecond, level.

ii. **Batch processing:** In this paradigm data are first stored and then processed. MapReduce [6] has become the foremost batch-processing model. The main idea behind MapReduce is to divide the huge data in to small chunks then these chunks are analyzed in parallel to obtain intermediate results. The final result is obtained by aggregating all the intermediate results. In bioinformatics, web mining and machine learning, MapReduce paradigm is widely used.

The differences between these two paradigms are listed in Table 2.

Characteristic	Stream processing	Batch processing
Input	Stream of new data	Data chunks
Data size	Infinite or unknown in advance	Known and finite
Hardware	Typical single limited amount of memory	Multiple CPUs, memories
Storage	Not store or store non-trivial portion in memory	Store
Processing	A single or few pass over data	Processed in multiple rounds
Time	A few seconds or even milliseconds	Much longer
Applications	Web mining, sensor networks, traffic monitoring	Widely adopted in almost every domain like bioinformatics, web mining, machine learning etc..,

Table 2. Major differences between stream and batch processing paradigms

Some research experts make an effort to integrate the advantages of these two models. Big data platforms are able to use alternative paradigms based on big data applications because of architectural differences between those two paradigms. Example, batch processing model deals with complex data storage and management systems, and streaming processing model do not. As the batch processing model is widely adopted in almost every domain, here we only focus on batch processing based big data platforms.

III. ARCHITECTURE OF BIG DATA

The four stages generation, acquisition, storage and processing are shown in Fig. 2. From beginning to ending, big data system deals with various phases in the digital data life cycle. Generally, based on applications, the big data system involves in multiple distinct phases. To overcome this, we adopt a systems-engineering approach which is worldwide recognized. In this approach we decompose a big data system into four phases, namely data generation, data acquisition, data storage and data analytics are explained clearly in below section.

- Data generation phase is totally focus on how data is generated. The phrase “big data” reveals the meaning large, diverse and complex datasets that are generated from diverse fields like sensors, video, click streams, and other available digital sources. These datasets are related to various domain specific values. In this paper, we concentrate on three predominant domains for datasets which include business, internet and scientific research. However, to squeeze the latest advances in the information and communications technology (ICT) domain, these datasets play an important role in collecting, processing, and analyzing for innovative solutions.
- Data acquisition is the second stage where after obtaining the information to be divided in to data collection, data transmission, and data pre-processing. First, we know that data comes from variety of sources like websites in which different forms of data available like images, videos and host formatted text. Here we need a data collection technology that obtains raw data from specific data production environment. After collecting

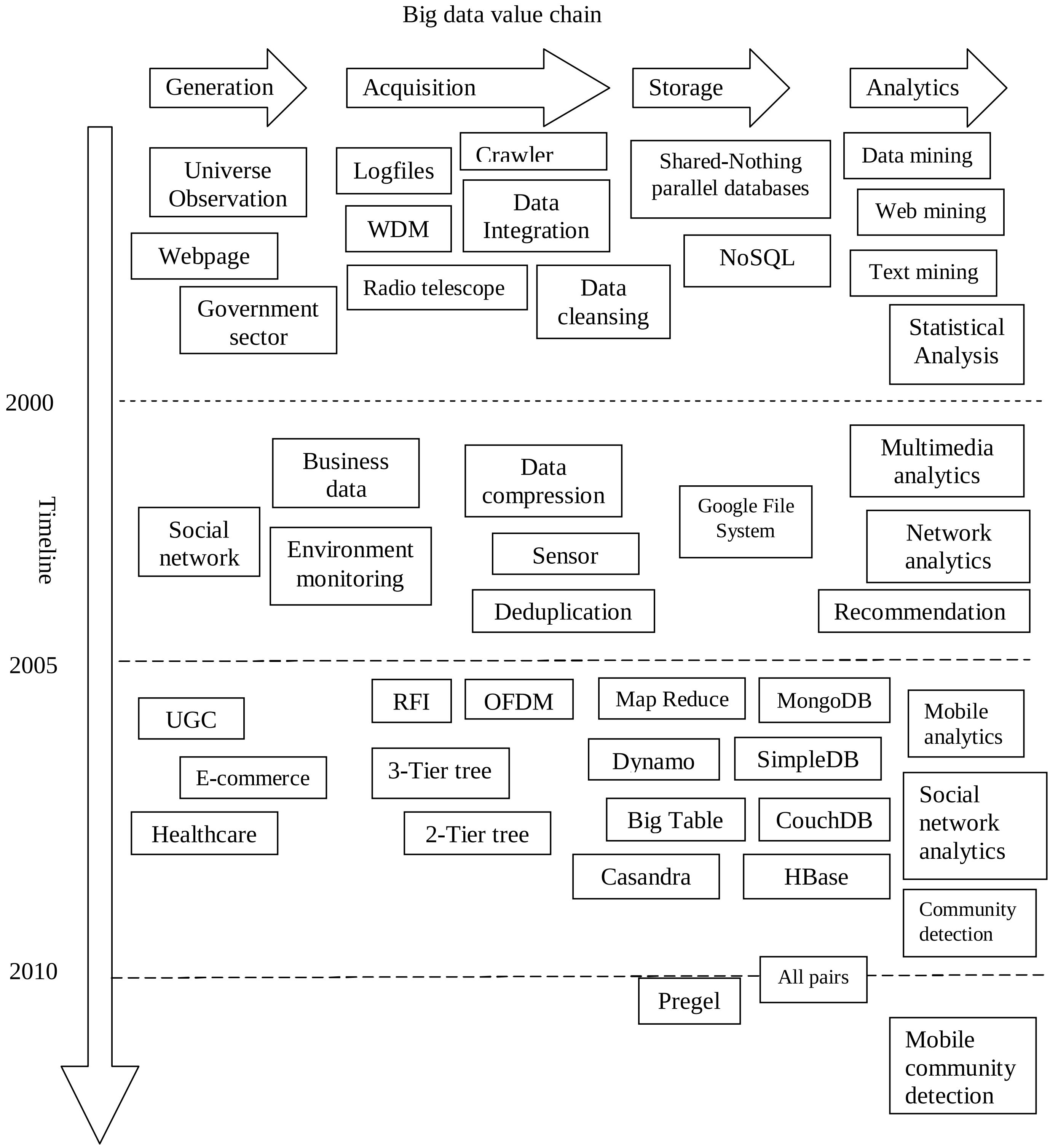

Fig 2. Architecture of big data

we need to send raw data to the proper storage sustaining system for various type of analytical issues, for this we need high-speed transmission technology. After that, we come to know that datasets will contain largely meaningless data, which unnecessarily increase the storage space and it affects the data analysis also. In general, redundancy is very common in most of the datasets, so to overcome it we need to apply data compression technology. Thus, we need to perform pre-processing to the data for efficient storage and mining.

- Data storage mainly concentrates on storing and managing large-scale datasets. A data storage system can be divided into two parts: hardware infrastructure and data management. The instantaneous demand of various tasks of a pool of shared resources is handled by hardware infrastructure. The hardware infrastructure is able to scale up and out and is also able to support different types of application environments. To maintain huge scale datasets, data management software is set up on hardware infrastructure. To analyze stored data, storage systems must provide several programming models, fast querying and interface functions.
- Data analysis is the phase where we need to extract the required results from analytical methods. There are many domain-specific analytical methods and tools to derive the analytical expectations. Emerging analytics research can be divided into six technical areas namely structured data analytics, text analytics, multimedia analytics, web analytics, network analytics, and mobile analytics. This classification is done based on data characteristics of each area.

IV. CONFRONTATION OF BIG DATA

A. Challenges of Big Data:

It is not an easy task to design and deploy a big data analytics system because big data is beyond the potential of current hardware and software platforms. The novel hardware and software platforms in turn demand new infrastructure to address the extensive challenges of big data. Here, we put our effort to classify the challenges of big data into three categories namely data collection and management, data analytics and system issues. The following challenges of big data must be met.

- Data representation: Most of the datasets are heterogeneous in structure, semantics, type, granularity, organization and accessibility. A proficient data presentation should be designed to replicate the structure, hierarchy and diversity of the data and an integration technique should be designed to enable efficient operations across different datasets.
- Redundancy reduction and Data compression: Huge number of redundant data is available in raw datasets. So, redundancy reduction and data compression techniques are efficient to lessen overall system overhead.
- Data life-cycle management: One of the most important challenges of big data is that current storage system cannot host the massive data. To address this challenge we need to set up the data importance principle associated with the analysis value to decide what parts of the data should be archived and what parts should be discarded.
- Data privacy and security: With the explosion of online services and mobile phones, privacy and security concerns regarding accessing and analyzing personal information are growing. It is difficult to understand what support for privacy must be provided at the platform level to eliminate privacy leakage.
- Approximate analytics: Analysis of entire dataset is becoming more difficult as data sets grow and the real time requirements become stricter. One solution to solve this problem is to provide approximate results by means of approximation query. The approximation query has two dimensions namely the accuracy of the result and the groups omitted from the output.
- Connecting social media: Social media have unique properties such as vastness, statistical redundancy and the availability of user feedback. To identify references from social media to specific product names, locations or people on websites, various extraction techniques have been successfully used. Applications can achieve high levels of precision and distinct points of view by connecting inter-field data with social media.
- Deep analytics: One of the drivers of excitement around big data is the expectation of gaining novel insights. Machine learning is one of the sophisticated analytical technologies to unlock such insights. However, it requires an understanding of probability and statistics to effectively leveraging these analysis toolkits.
- Energy management: Data transmission, storage and processing will certainly consume more energy, as data volume and analytics demand increases. The energy consumption of large scale computing systems has

attracted greater concern from economic and environmental perspectives. To provide extensibility and accessibility system level power control and management mechanisms must be considered in a big data system.

- Scalability: Very large data sets are supported by big data analytics. To scale the ever-growing size of complex data sets, big data systems are the only solution.
- Collaboration: Big data analytics is an interdisciplinary research field that requires experts from various professional fields collaborating to extract hidden values. To accomplish the goals of analysis, a comprehensive big data cyber infrastructure is necessary to allow broad communities of scientists and engineers to access the diverse data and apply their respective expertise.

B. Facing Challenges of Big Data Using Hadoop Framework

In handling massive data processing, Google's distributed file system and the MapReduce computation model had a great success. Its clone, Hadoop, has attracted substantial attention from both industry and scholars alike. For big data movement Hadoop has long been the mainstay. Apache Hadoop is an open-source software framework that supports massive data storage and processing. In old tradition it is very expensive to purchase hardware to store data and process data, Hadoop overcomes it with distributed processing of large amounts of data on large clusters of commodity servers. The following features make Hadoop suitable for big data management and analysis:

- Scalability: Hadoop provides a hardware infrastructure which is capable of scaled up and down with no need to change data formats. The system will automatically redistribute data and computation jobs to accommodate hardware changes.
- Cost efficiency: Hadoop carry enormously parallel computation to commodity servers, leading to a sizeable decrease in cost per petabyte of storage, which makes enormously parallel computation affordable for the ever growing quantity of big data.
- Flexibility: Hadoop is capable of tackling any type of data from any number of sources. In Hadoop, for further analysis different types of data from multiple sources can be aggregated. Thus, many challenges of big data can be addressed and solved.
- Fault tolerance: Missing data and computation failures are common in big data analytics. Hadoop is capable of recovering data and computation failures caused by node break-down or network congestion.

Hadoop Software Library:

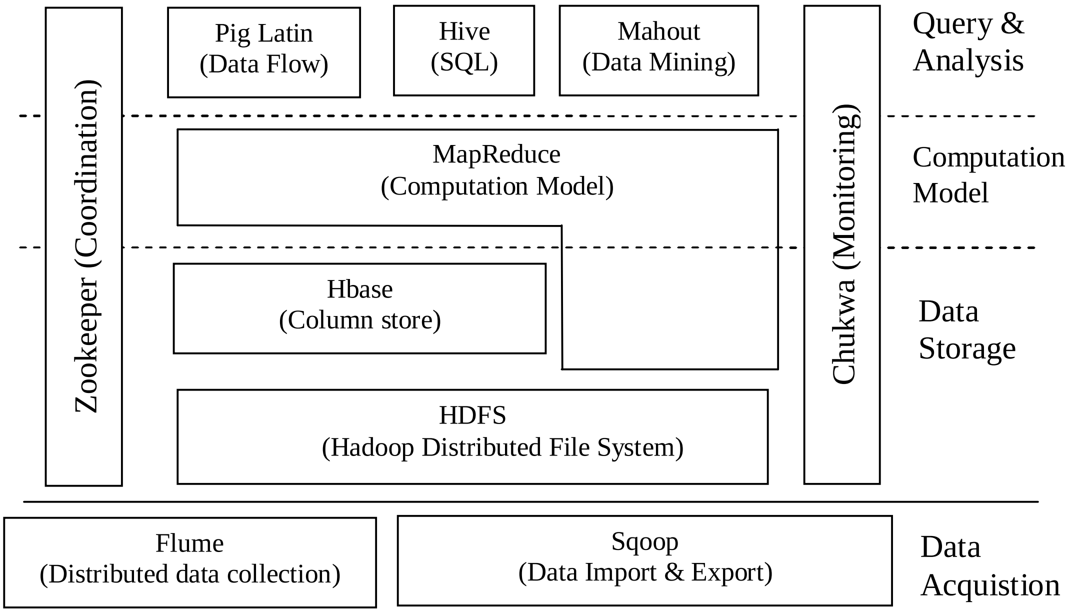

Fig 3. Architecture of Hadoop core software library

DOI: 10.15680/IJIRCCE.2015. 0405095

The Apache Hadoop software library is a huge computing framework consisting of several modules, including HDFS, Hadoop MapReduce, HBase, and Chukwa. The layered architecture of the core library is shown in Fig.3.Here we try to introduce different modules from bottom to up in examining the structure of the big data value chain. To achieve the data acquisition of the big data value chain, Apache Flume and Sqoop are two data integration tools. Flume is a distributed system that effectively collects, aggregates and transfers huge amounts of log data from different sources to a centralized system. Sqoop allows easy import and export of data among structured data stores and Hadoop. To store data Hadoop HDFS and HBase are useful. HDFS is a distributed file system build up to execute on commodity hardware that references the GFS design. HDFS is the primary data storage to represent Hadoop applications. A HDFS cluster consists of a single NameNode that is able to manage the file system, and collections of DataNodes that store the actual data. In general, a file is split into one or more blocks, and these blocks are stored in a set of DataNodes. To prevent missing data, each block has several duplications distributed in different DataNodes. Apache HBase is a column-oriented store designed based on Google's Bigtable. Thus, Apache HBase provides Bigtable-like capabilities above on top of HDFS. HBase can supply as input and output for MapReduce jobs execute in Hadoop and may be used through Java API, REST, Avor or Thrift APIs.

Based on Google's MapReduce, Hadoop MapReduce is built to analyze huge data. The MapReduce structure consists of a single master JobTracker and single slave TaskTracker per cluster node. The responsibility of master is to schedule jobs for the slaves, monitor them, and re-executing the failed tasks. The slaves completes the tasks as directed by the master. The MapReduce structure and HDFS execute on the same set of nodes which allows tasks to be scheduled on the DataNodes in which data already present. To analyze data set in MapReduce programs, Pig Latin and Hive are two SQL-like high-level declarative languages are used. For data flow tasks and to produce sequences of MapReduce programs PigLatin is appropriate whereas to summarize data and adhoc queries Hive is suitable. Mahout is a data mining library implemented on top of Hadoop that uses the MapReduce paradigm. It contains many core algorithms for classification, clustering and batch-based collaborative filtering.

To manage and monitor distributed applications that run on Hadoop, Zookeeper and Chukwa are used. Zookeeper is a centralized service for maintaining configuration, naming, providing distributed synchronization and providing group services. Chukwa is responsible for monitoring the system status and can display, monitor and analyze the data collected. Table 3 presents a quick summary of classification of Hadoop modules. Under this classification, Flume and Sqoop fulfil the function of data acquisition, HDFS and Hbase are responsible for data storage, MapReduce, Pig Latin, Hive and Mahout perform data processing and query functions, and ZooKeeper and Chukwa coordinate different modules being run in the big data platform. Hadoop is now extensively adopted industrially for variety of applications including spam filtering, web search, click stream analysis and social network recommendation.

Function	Module	Description
Data Acquisition	Flume	Data collection from disparate sources to a centralized store
	Sqoop	Data import and export between structured stores and Hadoop
Data Storage	HDFS	Distributed file system
	Hbase	Column-based data store
Computation	MapReduce	Group-aggregation computation framework
Query & Analysis	Pig Latin	SQL-like language for data flow tasks
	Hive	SQL-like language for data query
	Mahout	Data mining library
Management	Zokeeper	Service configuration, synchronization, etc.,
	Chukwa	System monitoring

Table 3. Hadoop module classification

In addition, much academic research is built upon Hadoop. The following surveys shows how the well known companies adopting Hadoop for their products and projects. As announced in June 2012, Yahoo! runs Hadoop on 42,000 servers in four data centers to support Yahoo! products and projects, such as Yahoo! search and spam filtering. Its largest Hadoop cluster holds 4,000 nodes but will increase to 10,000 with the release of Apache Hadoop 2.0. In the same month, Facebook announced that their Hadoop cluster processed 100PB data, and this volume grew by roughly half a PB per day in November 2012. Some notable organizations that use Hadoop to run large distributed computations found in [4]. In addition, there are a number of companies offering commercial implementation and/or providing support for Hadoop, including Cloudera, IBM, MapR, EMC and Oracle.

In spite of many improvements, Hadoop still require certain features found in DBMS, which is over 40 years old. As Hadoop doesn't support schema and index, it must traverse each item when reading the input and transform the input into data objects, which leads to performance degradation. The following are the some of the approaches that are currently used to recover the pitfalls of the Hadoop framework.

- Flexible Data Flow: Several algorithms cannot directly map into MapReduce functions, including loop-type algorithms that need state information for implementation and termination. Researchers taken a step to extend Hadoop to hold flexible data flow; HaLoop [10] and Twister [11] are examples that hold loop programs in MapReduce.
- Blocking Operators: The Map and Reduce functions are blocking operations, i.e., a task cannot go forward to the next step until all tasks are completed at the unique stage. This feature causes performance degradation and makes Hadoop inappropriate for on-line processing. Logothetis et al. [12] built MapReduce abstraction onto their distributed engine for ad hoc data processing. MapReduce Online [13] is proposed to support online aggregation and continuous queries. Li et al. [14] and Jiang et al. [15] utilized hash tables for improved performance and incremental processing.
- Scheduling: The Hadoop scheduler employs a simple heuristic scheduling strategy that evaluates the progress of each task to the average progress to determine re-execution tasks. This method is not appropriate for heterogeneous environments. Longest Approximation Time to End (LATE) scheduling has been proposed to improve the response time of Hadoop in heterogeneous environments. In a multi-user environment in which users immediately execute their jobs in a cluster, Hadoop implements two scheduling schemes namely fair scheduling and capacity scheduling. These two methods direct to poor resource utilization. Many researchers are working to improve the scheduling policies in Hadoop, such as the delay scheduler [16], dynamic proportional scheduler [17], deadline constraint scheduler [18], and resource-aware scheduler [19].
- Joins: MapReduce is devised for processing a single input. The extension of the holding join operator allows Hadoop to dispose several inputs. Join methods can be roughly classified into two groups: Map-side join [20] and Reduce-side join [21].
- Performance Tuning: Hadoop presents a general frame-work to hold a mixture of applications, but the default configuration scheme does not assure that all the applications run the best. Babu et al. [22] proposed an automatic tuning approach to find optimal system parameters for the given input data. Jahani et al. [29] presented a static analysis method for the automatic optimization of a single MapReduce job.
- Energy Optimization: A Hadoop cluster commonly consists of a huge collection of commodity servers, which get through a substantial quantity of energy. An energy efficient method for controlling nodes in a Hadoop cluster must be created. The Covering-Set approach [23] designates certain nodes to host at least a duplicate of each data block, and other nodes are powered off during low-utilization periods. The All-In strategy [24] saves energy by powering off all nodes until the job queue exceeds a predetermined threshold.

Hadoop is intended for batch-type application. In many real-time applications, Storm [26] is an excellent contender for processing unbounded streams of data. Storm can be employed for real-time analytics, online machine learning, continuous computation, and distributed RPC. Recently, Twitter disclosed their open project, called Summingbird [25], which integrates Hadoop and Storm.

V. Conclusion and Future Work

The age of big data is upon us, bringing with it an urgent need for advanced data acquisition, management, and analysis mechanisms. In this paper, we have presented a brief introduction to big data and explained its history and paradigms. The architecture of big data consists of four phases namely data generation, data acquisition, data storage, and data analysis. In architecture of big data, we tried to present the entire big data lifecycle. Then we discussed the challenges of big data in today's environment and how to overcome it. Then we introduce Hadoop framework to face the challenges of big data. Finally, we introduced the pitfalls of Hadoop framework and what are the solutions to overcome it. Many challenges in the big data system need further research attention. Below, we list the open issues where we can extend our research for future improvements: Big Data Platform, Processing Model, Big Data Application.

REFERENCES

1. J. Gantz and D. Reinsel, ``The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east," in *Proc. IDC iView, IDC Anal. Future*, 2012.
2. J. Manyika *et al*., *Big data: The Next Frontier for Innovation, Competition, and Productivity*. San Francisco, CA, USA: McKinsey Global Institute, pp. 1-137, 2011.
3. R. Cattell, ``Scalable SQL and NoSQL data stores," *SIGMOD Rec.*, vol. 39, no. 4, pp. 12-27, 2011.
4. J. Dean and S. Ghemawat, ``Mapreduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107-113, 2008.
5. J. H. Howard *et al*., ``Scale and performance in a distributed file system," *ACM Trans. Comput. Syst.*, vol. 6, no. 1, pp. 51 81, 1988.
6. J. Gantz and D. Reinsel, ``Extracting value from chaos," in *Proc. IDC iView*, pp. 1-12, 2011.
7. Y. Bu, B. Howe, M. Balazinska, and M. D. Ernst, ``Haloop: Efficient iterative data processing on large clusters," *Proc. VLDB Endowment*, vol. 3, nos. 1-2, pp. 285-296, 2010.
8. J. Ekanayake *et al*., ``Twister: A runtime for iterative mapreduce," in *Proc. 19th Assoc. Comput. Mach. (ACM) Int. Symp. High Perform. Distrib. Comput.*, pp. 810-818, 2010.
12. Logothetis and K. Yocum, ``Ad-hoc data processing in the cloud," *Proc. VLDB Endowment*, vol. 1, no. 2, pp. 1472-1475, 2008.
13. T. Condie, N. Conway, P. Alvaro, J. M. Hellerstein, K. Elmeleegy, and R. Sears, ``Mapreduce online," in *Proc. 7th USENIX Conf. Netw. Syst. Des. Implement.*, pp. 21, 2010.
14. B. Li, E. Mazur, Y. Diao, A. McGregor, and P. Shenoy, ``A platform for scalable one-pass analytics using mapreduce," in *Proc. Assoc. Comput. Mach. (ACM) SIGMOD Int. Conf. Manag. Data*, pp. 985-996, 2010.
15. D. Jiang, B. C. Ooi, L. Shi, and S. Wu, ``The performance of mapreduce: An in-depth study," *Proc. VLDB Endowment*, vol. 3, nos. 1-2, pp. 472-483, 2010.
16. M. Zaharia, D. Borthakur, J. Sen Sarma, K. Elmeleegy, S. Shenker, and I. Stoica, ``Delay scheduling: A simple technique for achieving locality and fairness in cluster scheduling," in *Proc. 5th Eur. Conf. Comput. Syst.*, pp. 265-278, 2010.
17. T. Sandholm and K. Lai, ``Dynamic proportional share scheduling in Hadoop," in *Job Scheduling Strategies for Parallel Processing*. Berlin, Germany: Springer-Verlag, pp. 110-131, 2010.
18. K. Kc and K. Anyanwu, ``Scheduling hadoop jobs to meet deadlines," in *Proc. IEEE 2nd Int. Conf. Cloud Comput. Technol. Sci. (CloudCom)*, pp. 388-392, Nov./Dec. 2010.
19. M. Yong, N. Garegrat, and S. Mohan, ``Towards a resource aware scheduler in Hadoop," in *Proc. Int. Conf. Web Services (ICWS)*, pp. 102-109, 2009.
20. S. Blanas, J. M. Patel, V. Ercegovac, J. Rao, E. J. Shekita, and Y. Tian, ``A comparison of join algorithms for log processing in mapreduce," in *Proc. Assoc. Comput. Mach. (ACM) (SIGMOD) Int. Conf. Manag. Data*, pp. 975-986, 2010.
21. J. Lin and C. Dyer, ``Data-intensive text processing with mapreduce," *Synthesis Lect. Human Lang. Technol.*, vol. 3, no. 1, pp. 1-177, 2010.
22. S. Babu, ``Towards automatic optimization of mapreduce programs," in *Proc. 1st Assoc. Comput. Mach. (ACM) Symp. Cloud Comput.*, pp. 137-142, 2010.
23. J. Leverich and C. Kozyrakis, ``On the energy (in) efficiency of hadoop clusters," *Assoc. Comput. Mach. (ACM) SIGOPS Operat. Syst. Rev.*, vol. 44, no. 1, pp. 61-65, 2010.
24. W. Lang and J.M. Patel, “Energy management for mapreduce clusters,” *Proc. VLDB Endowment*, vol. 3, nos. 1-2, pp. 129-139, 2010.
25. J.H. Howard et al., “Scale and performance in a distributed file system,” *ACM Trans. Comput. Syst.*, vol. 6, no. 1, pp. 51-81, 1988.
26. E. Jahani, M.J. Cafarella, and C. Re, “Automatic optimization for mapreduce programs,” *Proc. VLDB Endowment*, vol. 4, no. 6, pp. 385-396, 2011.

BIOGRAPHY

K.Srikanth is currently working as Assistant Professor in Computer Science and Engineering Department, G.Pulla Reddy Engineering College, JNTU Anantapur, Kurnool, Andhra Pradesh. He received Master of Technology (M.Tech) degree in 2011 from JNTUA, Anantapur, AP, India. His research interests are Data Mining, Big Data.

Dr.S.Zahoor-Ul-Huq is currently working as Professor in Computer Science and Engineering Department, G.Pulla Reddy Engineering College, JNTU Anantapur, Kurnool, Andhra Pradesh. He received PhD degree in 2012 from SKU, Anantapur, AP, India. His research interests are Networks, Cloud Computing, Big Data.

P.N.V.S.Pavan Kumar is currently working as Assistant Professor in Computer Science and Engineering Department, G.Pulla Reddy Engineering College, JNTU Anantapur, Kurnool, Andhra Pradesh. He received Master of Technology (M.Tech) degree in 2010 from JNTUA, Anantapur, AP, India. His research interests are Data Mining and Warehousing, Big Data.