International Journal of Computing Algorithm
Volume: 07, Issue: 01, June 2018, Page No.68-76
ISSN: 2278-2397

Encroachment in Data Processing using Big Data Technology

Tamilselvan Manivannan[1], Susainathan Amuthan[2]
[1]Department of Computer Science, Don Bosco College, Yelagiri Hills, Tamilnadu, India
[2]Department of Computer Applications, Don Bosco College, Yelagiri Hills, Tamilnadu, India
Email : maniorg.t71@gmail.com, muthanazeez@gmail.com

Abstract — The nature of big data is now growing and information is present all around us in different kind of forms. The big data information plays crucial role and it provides business value for the firms and its benefits sectors by accumulating knowledge. This growth of big data around all the concerns is high and challenge in data processing technique because it contains variety of data in enormous volume. The tools which are built on the data mining algorithm provides efficient data processing mechanisms, but not fulfill the pattern of heterogeneous, so the emerging tools such like Hadoop MapReduce, Pig, SPARK, Cloudera, Impala and Enterprise RTQ, IBM Netezza and Apache Giraphe as computing tools and HBase, Hive, Neo4j and Apache Cassendra as storage tools useful in classifying, clustering and discovering the knowledge. This study will focused on the comparative study of different data processing tools, in big data analytics and their benefits will be tabulated.

Keywords—Big data, analysis, tools, study

I. INTRODUCTION

The big data is the concept of large spectrum of data, which is being created day by day. In recent years handling these data is the biggest challenge. This data is huge in volume and being generated exponentially from multiple sources like social media (Facebook, Twitter etc.) and forums, mail systems, scholarly as well as research articles, online transactions and company data being generated daily, various sensors data collected from multiple sources like health care systems, meteorological department[19], environmental organizations etc. The data in their native form has multiple formats too. Also, this data is no longer static in nature; rather it is changing over time at rapid speed. These features owned by huge of current data, put a lot of challenges on the storage and computation of it. As a result, the conventional data storage and management techniques as well as computing tools and algorithms have become incapable to deal with these data. Big data solutions are distinguish by real-time complex processing and data relationship, advanced analytics, and search capabilities. Now big data has improved the demand of information management specializers in Software companies, Government, Constructing, and Medical sciences.

After the big data storage, comes the analytic processing. According to there are four critical requirements for big data processing. The first requirement is fast data loading. Since the disk and network traffic interferes with the query executions during data loading, it is necessary to reduce the data loading time. The second requirement is fast query processing. In order to satisfy the requirements of heavy workloads and real-time requests, many queries are response-time critical. Thus, the data placement structure must be capable of retaining high query processing speeds as the amounts of queries rapidly increase. Additionally, the third requirement for big data processing is the highly efficient utilization of storage space. Since the rapid growth in user activities can demand scalable storage capacity and computing power, limited disk space necessitates that data storage be well managed during processing, and issues on how to store the data so that space utilization is maximized be addressed. Finally, the fourth requirement is the strong adaptively to highly dynamic workload patterns. As big data sets are analyzed by different applications and users, for different purposes, and in various ways, the underlying system should be highly adaptive to unexpected dynamics in data processing, and not specific to certain workload patterns. Map Reduce is a parallel programming model, inspired by the “Map” and “Reduce” of functional languages, which is suitable for big data processing. The contribution of this study is to provide an analysis of the various big data tools is discussed[19].

A. Importance of Big Data

The importance of Big Data consists in the potential to improve efficiency in the context of use a huge volume of data, of different type. If Big Data is defined properly and used accordingly, organizations can get a better view on their business therefore leading to efficiency in different areas like sales, improving the manufactured product and so forth. Big Data can be used effectively in the following areas:

- In information technology to improve security and troubleshooting
- In customer service by using information from call centers to enhance customer satisfaction by customizing services
- In improving services and products through the use of social media content.
- In the detection of fraud in the online transactions for any organization
- In risk assessment by analyzing information from the transactions on the financial market[20].

II. RELATED WORKS

As big data becomes difficult to process using on-hand data management tools or traditional data processing applications there exists use of Hadoop tool to manage big data[21]. The challenges include capture, storage, search, sharing, transfer, analysis and visualization. The study presented the specific details along with description of various open source big data computing and storage tools enlisting the area of specialization[22]. With the recent endeavors of various developers concerning the use of tools in various fields one can expect a more enhanced environment along with more technical improvements. The work can be a helping hand to provide an insight in future to develop an application with more efficiency and availability a tool can be designed which instead of supporting a specific area can be extended to more fields.

A. Need of Big Data Analytics

There are various purposes for handling Big Data and exploring effective management and methodologies. The Big Data can be used for following purposes[16]. Business Intelligence: Intelligence is incorporated in making various business strategies as listed below:

- Business alignment strategies: It is required so that the output value and strategy may be tied up closely and may give the result after appropriate decision making.
- Behavioral and organizational strategies: These strategies speed up the task performance and improve productivity.
- IT strategies: It provides improved efficiency in IT at lower cost.
- Promotion and Advertisement strategies: These are required to make intelligent and effective marketing and advertisements to raise the profit.

B. Crime/ Fraud/ Fault Detection and Prediction:

In this reference, the Big Data analytics can play a vital role in several aspects[23].

- Credit card transaction: Analytics can predict the probability of a credit card holder of being fraudulent.
- Criminal identification is possible through deep analysis of call detail record[22].
- Querying, Searching and Indexing
- Keyword based search
- Pattern matching

C. Knowledge discovery / Data Mining

- Healthcare system: In healthcare system, Big Data Analytics could play a very vital role in variety of disease pattern identification, prediction and therapy suggestions such as diabetes, heart, cancer and Parkinson disease etc. through deeply digging Big Data using various data mining techniques.
- Statistical Modeling: In various day to day life transactions.
- Climate predictions and operative suggestions can be made based on the effective analytics of huge amount of climate and environmental data.

Defect detection and prediction in software and manufacturing products.

International Journal of Computing Algorithm
Volume: 07, Issue: 01, June 2018, Page No.68-76
ISSN: 2278-2397

III. COMPARISONS OF BIG DATA ANALYTICAL TOOLS AND DISCUSSION

To draw useful implications from the Big Data, appropriate tools are required to perform data collection, data storage and processing for various analytical perspectives.

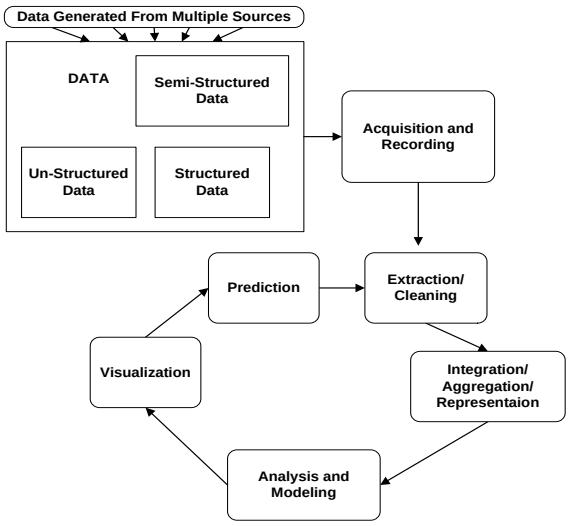

Fig. 1. Process Flow Diagram for Big Data Analytics

Fig. 2.

A. *Comparison of Computing Tools*

The comparison table it consisting of various computing tools and key features are listed below

TABLE I. COMPUTING TOOLS COMPARISON

Computing Tools	Scalability	Distributed Architecture	Parallel Computation	Fault Tolerance	Single Point Failure
Hadoop	Yes	Yes	Yes	High	Yes-At master nodes)
Cloudera Impala RTQ	Yes	Yes	Yes	Yes	Yes–If any query execution entire query is stopped
IBM Netezza	Yes	Yes	Yes- Asymmetric Massively Parallel Processing	Yes-Using redundant SMP hosts	At SMP server level
Apache Giraph	Yes	Yes	Yes–Bulk Parallel processing	Yes–by check points	No–Multiple master threads running

TABLE II. COMPUTING TOOLS COMPARISON

Computing Tools	Query Speed	Real-Time Analytics / Response Time	Streaming Query Support	ETL Required	Data Format for Analytics
Hadoop	Slow	No	No	No	Structured/ Unstructured
Cloudera Impala RTQ	High	Yes/in seconds	No	No	Structured /Unstructured
IBM Netezza	High	Yes/in seconds	Yes	No	Structured (RDBMS)
Apache Giraph	High	Ye / very less	No	No	Graph Database

TABLE III. COMPUTING TOOLS COMPARISON

Computing Tools / Paradigm	I/O Optimization	Optimized Query Plans	Efficiency	Latency Time for Query
Hadoop MapReduce	No	Not Applicable	Low for real time application(High for batch processing)	Not Applicable
Cloudera Impala RTQ	Yes	Yes	Higher	Low Latency due to use of dedicated distributed query engine
IBM Netezza	Not Required	Yes	High	Very Low- in seconds, due to in-memory data base processing and parallelism
Apache Giraph	Not Required	Yes–In terms of graph query/ algorithms	High	Low-In memory computation

Based on the comparison we identified following set of categories on which we would like to evaluate the above tools and computing paradigm in subsequent sub-sections:

B. Distributed Computation, Scalability and Parallel Computation

As we can see from the comparison tables, all computing tools provide these facilities. Hadoop distributes data as well as computing via transferring it to various storage nodes. Also, it linearly scales by adding a number of nodes to computing clusters but shows a single point failure. Cloudera Impala also quits execution of the entire query if a single part of it stops. IBM Netezza and Apache Giraph whereas does not have single point failure. In terms of parallel computation IBM Netezza is fastest due to hardware built parallelism[3][11].

C. Real Time Query, Query Speed, Latency Time

The Hadoop employs MapReduce paradigm of computing which targets batch-job processing. It does not directly support the real time query execution i.e OLTP. Hadoop can be integrated with Apache Hive that supports HiveQL query language which supports query firing, but still not provide OLTP tasks (such as updates and deletion at row level) and has late response time (in minutes) due to absence of pipeline parallelism and run-time scheduling of task assignment to distributed nodes.

All other three computing tools support the real time query execution very well and have early response time in seconds. However, Cloudera executes queries at least 10 times faster than Hive/MapReduce. Hadoop has comparatively higher latency time as it targets batch-job processing. The Cloudera Impala has low latency time as it uses a dedicated distributed engine to access data. IBM Netezza and Apache Giraph also achieve very low latency time due to in-memory database processing and computation. Apart from these tools there are other frameworks that are dedicated only to big data stream computing and mining for supporting real time analytics too but they are not discussed in this paper[12].

D. I/O and Query Optimization, Efficiency & Performance

Hadoop does not generate optimized query execution plans thus offers low efficiency for queries whereas Cloudera, IBM Netezza and Giraph have provision of I/O and query execution plans optimizations which results in higher efficiency and high performance in query execution. In Cloudera, purely I/O bound queries achieve approximately 3-4 times, queries of join or multiple Map Reduces achieves approximately 7-45 and simple

aggregation queries achieve 20-90 times performance gain over Hive/MapReduce. Giraph also provides high performance in terms of large scale graph processing for even trillion of edges[13].

E. ETL Requirement, Cost Effectiveness, Fault Tolerance

Since Hadoop, Giraph and Cloudera RTQ are open sourced, hence are a cost effective solution whereas IBM Netezza is proprietary to IBM, hence a costly solution for handling BigData. Also, since Clouera and Giraph perform in memory computation they do not require data input and data output that saves a lot of processing cost involved in I/O. None of the tools require the ETL (Extract, Transform and Load) service, thereby they save a major cost involved in data preprocessing. Hadoop is highly fault tolerant that is achieved by maintaining multiple replicas of data sets, and its architecture that facilitates dealing with frequent hardware malfunctions. Giraph achieves fault tolerance using barrier checkpoints[5][6].

F. Data Format, Language Support and Application Development

Hadoop HDFS is purpose built for supporting multi-structure data unlike the relational data bases whereas IBM Netezza deals strictly with the relational database. The Cloudera Impala RTQ supports both structured as well as unstructured data store. Apache Giraph is designed specially to work on graph data base such as Neo4j. Hadoop itself work on simply MapReduce paradigm but may support a range of languages for application development when integrated with other technologies such as Apache PIG that supports Python, JavaScript and JRuby languages. Cloudera can be successfully integrated with various BI tools supporting various languages. IBM Netezza directly supports a wide range of languages (C, C++, Fortran, Java, Lua, Perl, PythonR) for application development. Apache Giraph builds applications implemented using Java libraries[11].

G. Comparison of Storage Paradigms/Tools

The comparison table it consisting of various storage tools and key features are listed below.

TABLE IV. STORAGE TOOLS COMPARISON

Reference	Storage Tools	Open Source	Distributed	Scalable	Data Storage Format	ETL Required?
[17]	HBase	Yes	Yes	Yes	Structured	Yes
[8][17]	Apache Hive	Yes	Yes	Yes - Good	Structured/ Unstructured	Yes - Hence a bit higher latency in minutes
[17]	Neo4j	Yes	Yes	Yes	Non-relational	No
[17]	Apache Cassandra	Yes	Yes	Yes -vast	Structured / Semi-structured / unstructured (schema less)	No

Based on the comparison tables, storage tools can be categorized and evaluated based on following subsets of characteristics that provides some insights of applicability of various tools in different application domains:

H. Distributed, Scalability and Data Format Flexibility

All storage tools provide distributed data storage and querying facility and scalable in nature. HBase can be easily scaled-up with new records up to millions of rows and billions of columns. HBase and Hive run on Hadoop data node clusters, hence exploits its scalable property to further expand the database through data partitioning over multiple cluster data nodes. Neo4j supports scalability in terms of parallel readings on multiple nodes. Cassandra is highly scalable NoSQL database whose throughput and query response scales linearly with machine nodes. HBase has tabular data structure format, but it is wide-column and key-value-based data store capable of supporting a huge number of columns and flexible schema architecture. Hive also supports the

unstructured database whereas Cassandra supports a full range of structured and unstructured data formats and the dynamic changes in data structure can be accommodated easily[4]. Neo4j is extremely flexible schema less database solutions that solves graph modeled problems.

I. Availability, Fault Tolerance, Fault Recovery

The HBase and Hive run on Hadoop master/slave architecture of nodes that cause a single point failure at the master level failure. In HBase, Region Server that manages the partitioned data into a cluster region becomes single-point-failure. Similarly, Neo4j's write-master is single point failure. All these are fault tolerant but HBase and Hive offers a bit low availability. Neo4j has a much better availability whereas Cassandra is highly available due to the absence of master/slave paradigm[14].

J. Real Time and Streaming Query Support, Query Performance

HBase is optimized for read operations and hence not much efficient for writes. On the other hand, Hive does not suit for OLTP due to absence of row level updates and deletes. Neo4j supports real time queries, but in the form of graph traversals. Cassandra supports OLTP very well on a full range of data formats. For stream query analysis, Cassandra is the best solution. The HBase stores data in Map Files (that are indexed Sequence Files) thus becomes a suitable choice for streaming analysis of a MapReduce kind of style that involves occasional random look ups. The performance of query in Hive is increased through meta-store, data partitioning and external level table support that is not required to be pushed on HDFS. Neo4j overcomes the performance degradation problem in traditional RDBMS queries with several joins because a graph traversal is performed which works at the same speed, no matter how much data constitutes it. Cassandra provides very high throughput for write operation queries[15].

K. Open Source, Access Control, Language Support

All storage tools discussed in this paper are open source, hence free available and cost effective. Out of them, Cassandra and Neo4j does not have a requirement of ETL services that causes no extra processing cost overhead and become the cheapest choices among them. HBase, Neo4j and Apache Cassandra fully support an access control mechanism that provides security, authorized access and modification to the database whereas Hive does not provide such efficient control. HBase is java centric hence directly supports lesser languages but through REST and Thrift Gateways interface support languages. Neo4j supports several languages through various Neo4j language clients and REST APIs whereas Cassandra supports all key languages that are required to develop a variety of applications without need of any intermediate gateways[16].

The overall management of Big Data involves storing, processing and analyzing it for various purposes; hence we can visualize the infrastructure, to handle Big Data related tasks, as a layered architecture.

TABLE V. STORAGE TOOLS COMPARISON

Reference	Storage Tools	ACID Transaction Support	Real Time Query/ OLTP	Stream Query Support	Range of SQL Supported queries	Single Point Failure
[17]	HBase	Yes – Rollback support	No	No- partially	No Support of SQL Can support when integrated with Hive	Yes- At Region Server level
[8][17]	Apache Hive	Yes	No	No	Limited – through HiveQL that has been extended through writing	Yes – At master node of underlying hadoop framework

International Journal of Computing Algorithm
Volume: 07, Issue: 01, June 2018, Page No.68-76
ISSN: 2278-2397

					custom function	
[17]	Neo4j	Yes	Yes – in form of graph traversal and deletion of node	No	Queries in the form of Graph Traversals	Yes – At Master level responsible for write replicas
[17]	Apache Cassandra	Yes – provides AID only	Yes	Yes	Yes – through CQL whereas JOINs and most SQL search are supported by defining schema	No – Hence high Availability

TABLE VI. STORAGE TOOLS COMPARISON

Reference	**Storage Tools**	**Failover Recovery**	**Fault Tolerance**	**Meta Data Store**	**Language Interface Support**	**Access Control**
[17][18]	HBase	Long time at node level failure whereas 10 to 15 minutes at Region Server level	Yes	Yes	Less - Java Centric, Non-java clients are supported through REST and Thrift gateways	Yes
[8][17][18]	Apache Hive	Yes - supports node level recovery	Yes - replication mechanism to have synced with metastore	Yes	Clojure, Go, Groovy, Java JavaScript, Perl, PHP, Python, JRuby, Scala	No in-built security provisions
[17][18]	Neo4j	Yes - Select the new master	Yes - Supported by ACID Transaction system	Yes - Optional schema	Java, PHP, .Net, Python, Clojure, Ruby, Scala, etc.	Yes
[17][18]	Apache Cassandra	Yes - Optimized for the Recovery performance	Yes - Optional	Yes -Due to flexible schema support	Java, Python, Node.JS, etc.	Yes - Provided by the DataStax Enterprise

International Journal of Computing Algorithm
Volume: 07, Issue: 01, June 2018, Page No.68-76
ISSN: 2278-2397

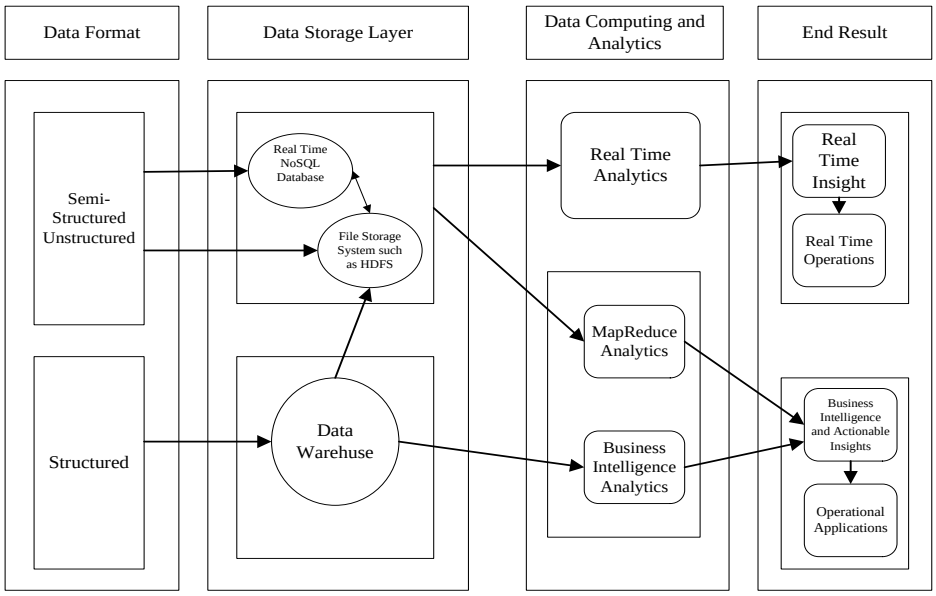

Fig. 3. Layered Architecture for Big Data Handling

IV. CONCLUSIONS

This comparison aims to find some available computing and storage tools that are being used in current scenario to address challenges of Big Data processing. We have categorized the survey into two streams. One stream contains study and survey of existing computing paradigms and tools used to perform computation on Big Data and the other stream gives a detailed survey of storage mechanisms and tools available today. In this reference, we focused on Apache Hadoop, Cloudera Impala and Enterprise RTQ, IBM Netezza and Apache Giraph as computing tools and HBase, Hive, Neo4j and Apache Cassandra as storage tools. Based on deep and detailed analysis of their features, relative advantages and disadvantages we have made a critical comparison among these tools. The comparison is made on the most striking attributes that one looks for before choosing these tools for its application domain to handle Big Data. We have discussed various issues associated with various tools and compared them accordingly and gave critical review on the suitability and applicability of different storage and computing tools with respect to a variety of situations, domains, users and requirements.

V. FUTURE ENHANCEMENTS

As we know big data is an emerging field. In this study we have discussed the various analytical tools and its features. We can implement one of the tools in future. The hope is to implement better and better technique which can robustly resolve the drawbacks and find the best solution. Future work will focus on performance of Hadoop on cloud platforms.

References

[1] Gandomi, A,Haider, M. Beyond the hype: Big data concepts, methods and analytics,International Journal of Information Technology,35(2)- 2011.
[2] Seema Acharya, Subhasini chellapan, “Big data Analytics”,Willey,edition-2015.
[3] M. Kendrick, Big data, Big Challenges, Big opportunities: IOUG Big data Strategies Survey- 2015
[4] Shavachko K, Kuang H, Radia and chansler R, The hadoop distributed file system, Proccedings of IEEE Conference-2010.
[5] Fransico P, The Netezza data appliance architecture: A platform for high performance data warehousing and analytics,IBM Redbooks-2011.
[6] DataStax Corporation-Introduction to cassendra-A White Paper(2013)
[7] Katal,A wazid ,m,”big data:Issues,callenges,tools and good pratcices”.
[8] Baski,k,”Considerations for big data:Architecture and approach”-2012.
[9] http://www.gise.iitb.ac.in/wiki/images/2/26/Hive.pdf

[10] http://en.wikipedia.org/wiki/big-data ret retrived 2015-03-12
[11] http://en.wikipedia.org/wiki/Apache-Hadoop retrieved 2015-03-12
[12] http://en.wikipedia.org/wiki/Map Reduce retrieved 2015-02-27
[13] http://en.wikipedia.org/wiki/pig retrieved 2015-02-27
[14] Tom white "hadoop the definitive guide" proc O'Reilly Media,Edition 3,May 2012.
[15] Donald Minner,Adam Shook "mapreduce design pattens" proc O'Reilly Media,Edition November 2012.
[16] https://www.researchgate.net/publication/264555968_Big_Data_Analytics
[17] http://www-users.cs.umn.edu/%7Ekumar/dmbook/ch4.pdf
[18] http://www.oracle.com/us/corporate/analystreports/infrastructure/ioug-big-data-survey-1912835.pdf
[19] https://www.researchgate.net/publication/299423030Comparative_StudBig_Data_Computing_and_Storage_Tools
[20] http://www.dbjournal.ro/archive/10/10.pdf
[21] https://www.edureka.co/blog/videos/introduction-hadoop-administration
[22] An efficient Parkinson disease diagnosis system based on Least_Squares Twin_Support Vector Machine and Particle Swarm Optimization
[23] http://www.dailymail.co.uk/indiahome/indianews/article-3004655/India-play-vital-role-dealing-threat-Boko-Haram.htm