International Journal of Interdisciplinary Innovative Research &Development (IJIIRD)
ISSN: 2456-236X
Vol. 01Special Issue 02|2017

Comparision between Big Data and Hadoop

Ms. NehaBhagwat, Ms.SwatiMandurkar,MandarDeshmukh,Tushar Sangole

Manav School of Engineering, Akola

Abstract

As a result of tremendous rise in internet usage like social media and forums, mail systems, scholarly and research articles, daily online transactions from multiple sources like health care systems, meteorological and environmental organizations etc., the data collected has shoot up exponentially. This vast collection of data, called Big Data, has caused the traditional tools incompetent for managing it from either of storage, computing or analytical perspective. There is an immense need of architectures, platforms, tools, techniques and algorithms to handle Big Data. The available technologies deal with two broad aspects related to Big Data that are Big Data Storage Management and Big Data Computing, focused to overcome various challenges such as scalability, faster processing speed, multiple format data processing, availability, faster response time and analytics etc. This paper reviews recent trends of storage and computing tools with their relative capabilities, limitations and environment they are suitable to work with.

Keywords*: Big Data Computing, Big Data Computing Tools, Big Data Storage Tools,Big Data Analytics*

1. Introduction

Current world is the world of data. We have data all around us. 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 [1], meteorological department, 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 bulk 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. Despite of so many challenges associated with these data, we cannot ignore the potentials and possibilities lying in it that can support for analytics and for hidden patterns identification. These analytics can be very effective in making business strategies and predicting effective decisions, finding various hidden patterns associated with several diseases and their attributes, in genomics to analyze thousands of genes and their associated roles in biological systems, in climate monitoring and prediction, GPS and other satellite parameters mining *etc.*

1.1 Big Data Formats and its Sources

Big data is a huge collection of data over a time frame that is so complex and difficult to process and manage using conventional database management tools [2]. Big Data and its sources can be categorized into following categories:

- Structured Data - generated from various researches efforts, CRM (Customer Relationship Management) and other such traditional databases.
- Semi-structured Data - such as XML formatted data.
- Unstructured Data – These data can be generated by humans such as social media, discussion forums and customer feedback, comments, emails *etc.* or may be generated by machine such as online transactional, satellite and environmental data collected through various sensors, web -logs, call records *etc.*

1.2 Big Data Characteristics and Big Data Challenge

There are four basic characteristics depicted in Figure 1 that Big Data shows always. These are Volume, Variety and Velocity [3-4]. Each aspect puts a challenge in handling and processing this data to extract some meaningful implications. These challenges could be in collection, integration, storage, sorting, searching, retrieval, analysis, and visualization from the various aforementioned key aspects of the Big Data.

Velocity: This aspect of Big Data is associated with the speed at which data is being produced and processed. When we look for the real time processing and response the speed of data production becomes a critical challenge for analytical and visualization tools. If the response time of the analytical tools is not capable to cope up with speed of data arriving, the result becomes useless.

Figure 1. Varying Characteristics of Big Data Over a Period of Time

1.3 Need for Big Data Management and Processing

There are various purposes for handling Big Data and exploring effective management and methodologies. The Big Data can be used for following purposes:

- 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.

1.4 Organization of Paper

The organization of the paper goes the way shown in Figure 2. The first section introduces the Big Data, different sources of their generation, their characteristics and challenges associated with it. Also, it discusses the need of handling and processing of Big Data in current scenario in different areas of applications. The second section contains a detailed description about available four well known tools and techniques for storing and four for computing Big Data with along with their advantages/disadvantages and the suitable environment they are applicable to work with. The third section gives the comparison of various tools and techniques based on their capabilities and limitations associated with them. The fourth section finally concludes this paper with some useful suggestions and recommendations.

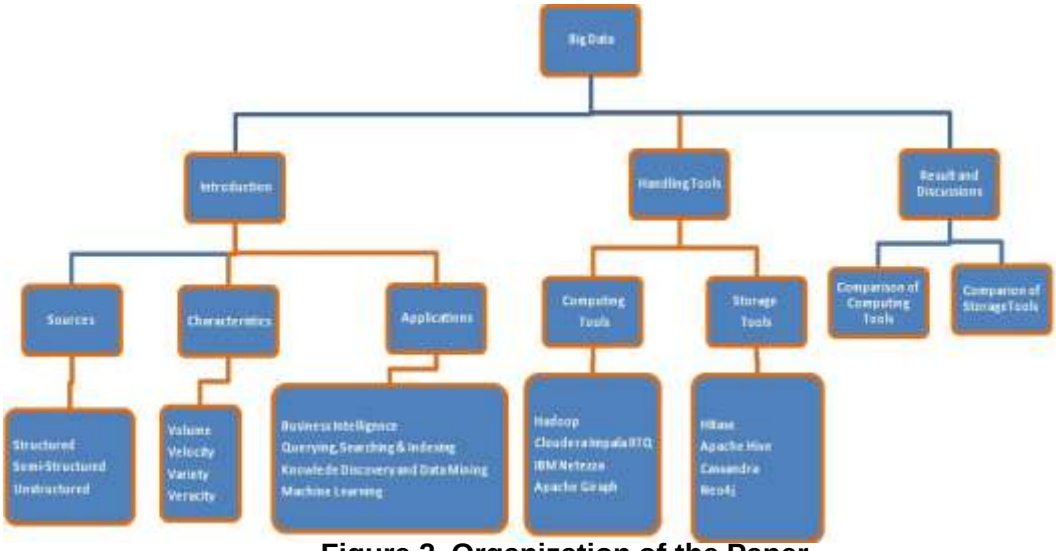


Figure 2. Organization of the Paper

2. Big Data Storage and Computing Paradigms and Tools

To draw useful implications from the Big Data, appropriate tools are required to perform data collection, data storage and processing for various analytical perspectives. The normal process flow diagram for Big Data Analytics is shown in Figure 3.

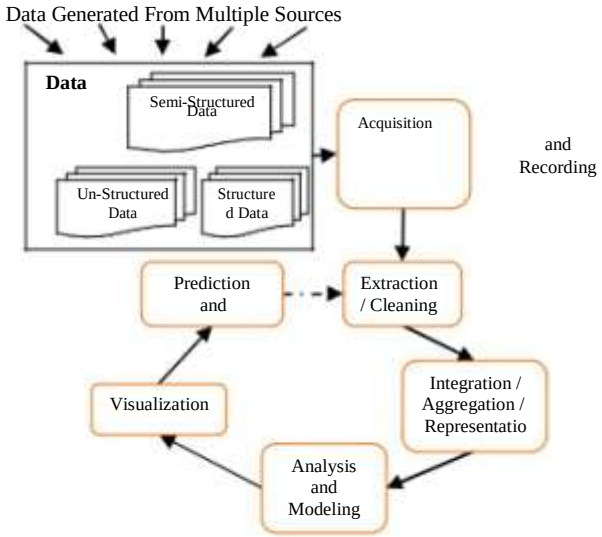


Figure 3. Process Flow Diagram for Big Data Analytics

2.1 Big Data Computing / Processing Tools

2.1.1. HADOOP MapReduce

Hadoop also known as Apache Hadoop [13-15] is an open source framework that has been provided by Apache. This framework is developed to deal with distributed and scalable computing as well as storage management of huge data, the Big Data. Hadoop platform includes two core layers; one is the distributed storage layer that is built on the HDFS (Hadoop Distributed File System) [16] inspired by the Google File System [17] and the second layer is the distributed computing layer whose key idea is MapReduce computing paradigm, initially, developed by Google.

i. Advantages of Hadoop

- Open source: Being an open source, Hadoop is freely available [13].
- Cost Effective: Hadoop saves cost as it employs cheaper low end cluster of commodity of machines instead of costlier high end server. Also, distributed storage of data and transfer of computing code rather than data saves high transfer costs for large data sets [13].
- Scalable: To handle larger data, the Hadoop is capable to scale linearly by putting additional nodes in clusters [13], [14].

2.1.2. Cloudera Impala and Cloudera Enterprise RTQ

Cloudera Enterprise RTQ driven by Cloudera Impala enables enterprises to exploit advantageous features of SQL tools to achieve real-time analytics potentials when working with large volumes of data, whose nature may be structured and unstructured
Various business analysts and IT industries can use it over a wide range of supported data types as well as huge data volumes to interact in real time with aHBase or a HDFS data store for the sake of analytics. The Cloudera Impala's position in Hadoop stack is depicted,

2.1.3. IBM Netezza

Netezza can be placed in both storage and computing category as it provides data warehouse as well as analytics appliance. Netezza is based on Asymmetric Massively Parallel Processing (AMPP) shared-nothing architecture which is basically a two-tier architecture [31,33] shown in Figure 6 and Figure 7 which handle large complex queries very quickly. The first tier employs a high performance Linux based Symmetric Multi-Processing host. This tier is responsible for compiling data query jobs and accordingly generating execution plans. It breaks down the original query task into sub-tasks suitable for parallel execution. Afterwards, these subtasks are distributed over the second tier [32].

2.1.4. Apache Giraph

Apache Giraph, running on top of Hadoop framework, is the open sourced version of Google's proprietary product Google Pregel [38]. It also has distributed processing structure suitable basically for large scale graph processing [39,43-44] such as in analysis of the interconnected web (for Page Ranking) or social media (Facebooks, Twitters, LinkedIn *etc.*) interaction that are nothing but a graph of interconnected vertices which may be a web page linked to another page through the edge (hyperlink) or it may be users in social media connected with each other through edges representing friendship or some kind fan or business following *etc.* The Giraph basically based on the Valient model [40] of Bulk Synchronous Parallel computation model. Usually, the Giraph is used in combination with well-known graph databases such as Infinite Graph or Neo4j or with Hadoop.

i. Advantages of Apache Giraph

- Scalable: It is used for large scale graphs' analysis involving up to trillion of edges. Giraph computing is based on the Valiant model of Bulk Synchronous Parallel computation [40].
- Fault Tolerant: It achieves fault tolerance by employing check -points technique [41].

Id = 1
Key =10

Edge < Id = 5, value =10>

Id = 5
Key = 3

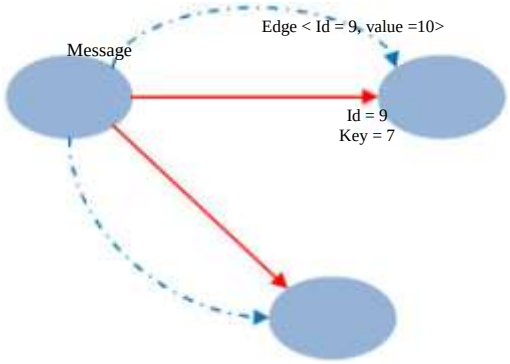


Figure 8. Vertex Centric Computing Model of Apache Giraph [43]

ii. Disadvantages of Apache Giraph

- Apache Giraph is still in a very immature phase of development [41-42].
- It lacks in providing a complete set of offered algorithms [42].

2.2. Big Data Storage Tools

2.2.1. HBase

Apache HBase [45-46] is an open source non-relational database that aims to host very large sized tables consisting of millions to billions of rows and columns. HBase allows grouping various attributes to make column families as described in Figure 9. In this way, attributes of a column family are put together in the table [47]. Apache HBase is a distributed version of the database that facilitates the same capabilities to Hadoop's HDFS as the Big Table of Google provides to the Google File System [48].

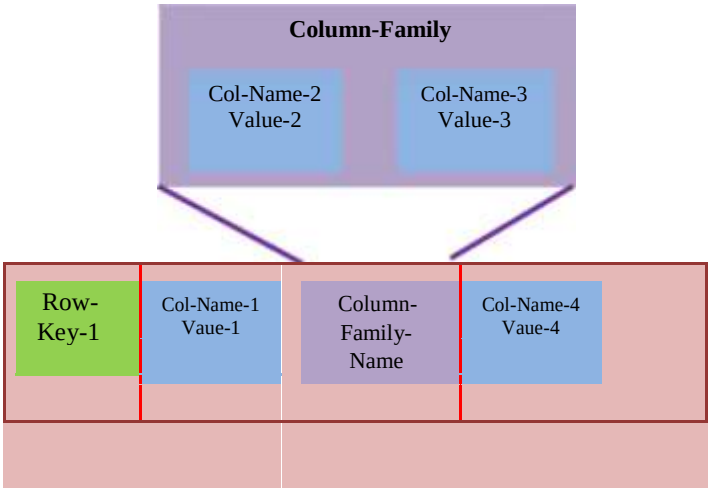


Figure 9. Column Family Containing as Attributes Columns 2 and 3 [46]

i. Advantages of HBase

Apache HBase provides following capabilities:

- Scalability: It scales horizontally, as it is a wide-column key-based data stores. Therefore, it is robust also [45].
- HBase performs consistent reads/writes on the underlying data in the database but it is optimized for performing read operations [49].

ii. Disadvantages of HBase

There are some technical limitations with almost all NoSQL solutions and so is the case with HBase:

- Compactions affect the consistent low latency in HBase [49].
- Single Point Failure: In HBase rows are partitioned into regions [49] and each region is allocated to a Region-Server which becomes a single point of failure. Also, HBase takes long recovery times for node failures. On the other hand, the Region Server failover takes approximately 10-15 minutes which is quite high [52].

2.2.2. Apache Hive

Apache Hive, built upon Apache Hadoop, is a data warehouse tool that provides effective management of very large data which is stored in HDFS. It also provides effective query execution facility using a query language resembling to SQL. This query language is known as HiveQL. Since the language is SQL-like, hence the SQL users can easily fire their query on the database. Also, it is helpful for those programmers who know the MapReduce paradigm of computing [53].

i. Advantages of Apache Hive

Apache Hive facilitates following capabilities:

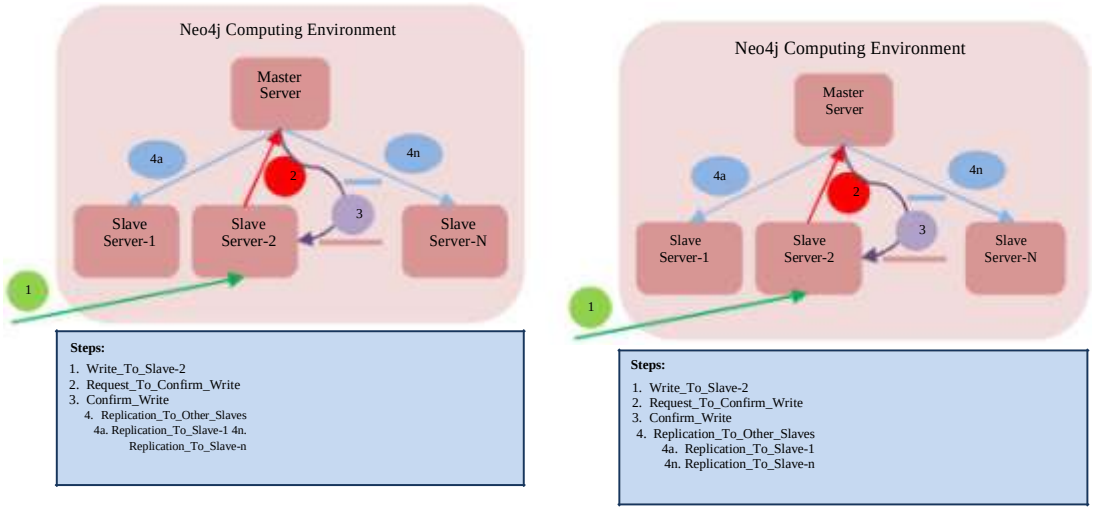
- Easy data ETL services: Hive provides data extract, data transform and data load operation in an easy way. Hive performs reads/writes which are independent of file formats. It uses SerDe (Serializers/Deserializers) framework libraries to support formats such as text, sequential files, control delimited or a user defined file format [54-55].
- Hive has provision for tables at external level to facilitate data processing without storing it actually on HDFS. Data partitioning in Hive, is performed at table level that improves query execution performance [54].

2.2.3. Neo4j

Neo4j is a graph database that is available as open source as well as commercial licensed version. It stores data modeled as a graph which is a collection of nodes (with an Id) and relationships among them represented as edges in the graph. These nodes or edges store some properties represented as key/value pairs. Neo4j is an embedded, fully transactional, a disk-based Java persistence engine.

i. Advantages of Neo4j

- Massive scalability: Neo4j can easily handle large graphs containing nodes / relationships / properties of order of billions using even a single machine. Its computation can run in parallel on multiple processors via read threads [54].
- Single Point Failure: Neo4j has a Master -Slave model for replication as depicted in Figure 10(a) and 10(b) where all write operations are handled by the master and changes performed are reflected to the read only slaves. At master level, there can be a single point failure [57].
- Slow Online Write Transaction Speed: While committing in Neo4j, data is made permanent on disk that requires disk writes at each commit hence write speed is limited by the single server hardware's I/O capacity.



(a) When Master is Written (b) When a Slave is Written

Figure 10. Replication Model of Neo4j [64]

3 Results and Discussion

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 as shown in Figure 11.

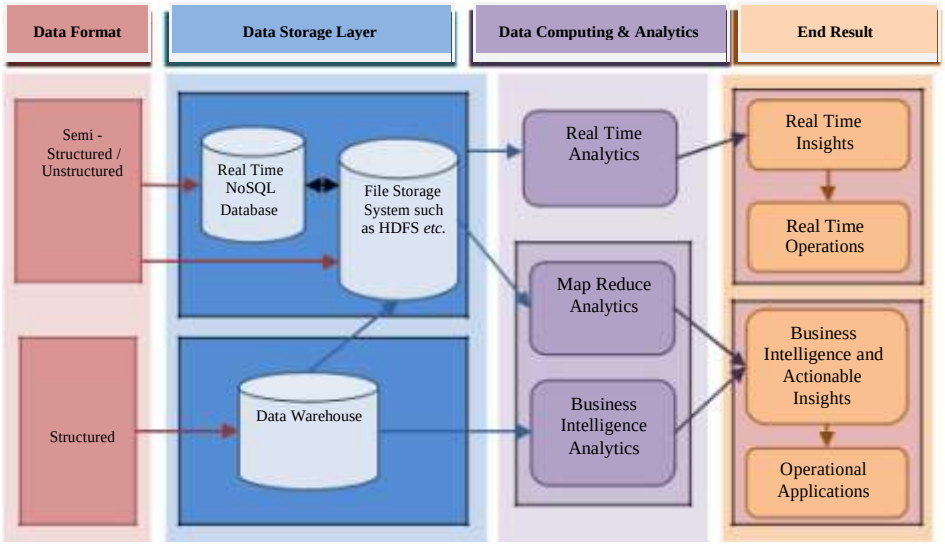


Figure 11. Layered Architecture for Big Data Handling

Through the detail analysis of various computing and storage tools, we have found several attributes that may give us a way to compare these tools. The various advantages/disadvantages of these tools let us know the suitability of various tools in various kinds of application domains.

3.1. Comparison of Computing Tools

Below is comparison Table 1, Table 2 consisting of various computing tools and the key features or facilities they support.

Table 1. Computing Tools Comparison Table

Computing Tools	Scala-bility	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 host quits 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 2. Computing Tools Comparison Table

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	Yes / very less	No	No	Graph Database

Based on the comparison Table 1, Table 2 we identified following set of categories on which we would like to evaluate the above tools and computing paradigm in subsequent sub-sections:

3.1.1. 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.

3.2. Comparison of Storage Paradigms/Tools

Below is comparison Table 3, Table 4 and Table 5, consisting of various storage tools and the key features or facilities they support.

Table 3. Storage Tools Comparison Table

Storage Tools	Open Source	Distributed	Scalable	Data Storage Format	ETL Required?
HBase	Yes	Yes	Yes	Structured *i.e.* Tabular but not exactly row-oriented Relational Table	Yes
Apache Hive	Yes	Yes	Yes - Good	Structured/ Unstructured	Yes - Hence a bit higher latency in minutes
Neo4j	Yes	Yes	Yes	Non-relational *i.e.* graph database (schema less)	No
Apache Cassandra	Yes	Yes	Yes -vast	Structured / Semi-structured / unstructured (schema less)	No

Table 5. Storage Tools Comparison Table

Storage Tools	ACID Transaction Support	Real Time Query / OLTP	Stream Query Support	Range of SQL supported queries	Single Point Failure
HBase	Yes - Rollback support	No	No - partially	No Support of SQL - Can support when integrated with Hive	Yes - At Region Server level
Apache Hive	Yes	No	No	Limited - through HiveQL that has been extended through writing custom functions	Yes - At master node of underlying hadoop framework
Neo4j	Yes	Yes – in form of graph traversal and insertion and deletion of nodes	No	Queries in the form of Graph Traversals	Yes - At Master level responsible for write replicas
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

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

3.2.1. 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 is also scaled up by simply adding new nodes if required and can model 232 billion 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.

4. Conclusion

This survey aims to find some available computing and storage paradigms and 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.

References

[1] S. Agarwal, Divya and G. N. Pandey, ―SVM based context awareness using body area sensor network for pervasive healthcare monitoring‖, IITM, ACM, New York, **(2010)**, pp. 271-278.
[2] M. R. Wigan and R. Clarke, ―Big Data's Big Unintended Consequences‖, IEEE Computer Society, DOI:http://dx.doi.org/10.1109/MC.2013.195, vol. 46, no. 6, **(2013)**, pp. 46-53.
[3] M. Kendrick, ―Big Data, Big Challenges, Big Opportunities: 2012 IOUG Big Data Strategies Survey‖, http://www.ioug.org/p/cm/ld/fid=91, (Retrieved on September 2, 2015), **(2012)**.
[4] N. Wallis, ―Big Data in Canada: Challenging Complacency for Competitive Advantage‖, in: T. White (Eds.), Hadoop: The Definitive Guide, third ed., O'Reilly Media, Yahoo Press, **(2012)**.
[5] J. Constine, ―How Big Is Facebooks Data? 2.5 Billion Pieces Of Content And 500+Terabytes Ingested Every Day‖, http://techcrunch.com/2012/08/22/how-big-is-facebooks-data-2-5-billion-pieces-of-content-and-500-terabytes-ingested-every-day, (Retrieved on September 2, 2015), **(2012)**.
[6] D. Tam, ―Facebook processes more than 500 TB of data daily‖, http://news.cnet.com/ 8301-1023_3-57498531-93/facebook-processes-more-than-500-tb-of-data-daily, (Retrieved on September 3, 2015), August **(2012)**.
[7] IBM, ―What is big data?‖ http://www-01.ibm.com/software/data/bigdata/what-is-big-data.html, (Retrieved on September 3, 2015), **(2013)**.
[8] D. Tomar and S. Agarwal, ―Predictive Model for diabetic patients using Hybrid Twin Support Vector Machine‖, Proceedings of 5th International Conferences on advances in communication Network and Computing, **(2014)**.
[9] B. R. Prasad and S. Agarwal, ―Modeling risk prediction of diabetes—A preventive measure‖, Proceedings of 9th IEEE International Conference on Industrial and Information Systems (ICIIS' 14), **(2014)**, pp. 1-6.
[10] D. Tomar, B. R. Prasad and S. Agarwal, ―An efficient Parkinson disease diagnosis system based on Least Squares Twin Support Vector Machine and Particle Swarm Optimization‖, Proceedings of 9th IEEE International Conference on Industrial and Information Systems, **(2014)**, pp. 1-6.
[11] D. Tomar, and S. Agarwal, ―A survey on Data Mining approaches for Healthcare‖, International Journal of Bio-Science and Bio-Technology, vol. 5, no. 5, **(2013)**, pp. 241-266.
[12] S. Agarwal, Divya and Siddhant, ―Prediction of Software Defects using Twin Support Vector Machine‖, Proceedings of 2nd IEEE International conference on Information Systems & computer Networks (ISCON), **(2014)**, pp. 128-132.
[13] J. Venner, ―Pro Hadoop‖, a press, **(2009)**.
[14] T. White,‖Hadoop: The Definitive Guide‖, third ed., O'Reilly Media, Yahoo Press, **(2012)**.
[15] S. Ketu, B. R. Prasad and S. Agarwal, ―Effect of Corpus Size Selection on Performance of Map-Reduce Based Distributed K-Means for Big Textual Data Clustering‖, In Proceedings of the Sixth International Conference on Computer and Communication Technology 2015, pp. 256-260. ACM, **(2015)**.

[16] W. Tantisiriroj, S. Patil and G Gibson, ―Data-intensive File Systems for Internet Services‖, A Rose by Any Other Name (CMU-PDL-08-114). Research Centers and Institutes at Research Showcase, http://repository.cmu.edu/pdl/9. Technical report, Carnegie Mellon University, **(2008)**.

[17] M. K. McKusick and S. Quinlan, ―GFS: Evolution on Fast-forward‖, ACM Queue, New York, vol. 7, no. 7, **(2009)**.

[18] K. Shvachko, H. Kuang, S. Radia and R Chansler, ―The Hadoop Distributed File System‖, Proceedings of IEEE Conference, 978-1-4244-7153-9/10, **(2010)**.

[19] J. Dean and S. Ghemawat, ―Mapreduce: Simplified data processing on large clusters‖, commun. ACM, vol. 51, no. 1, **(2008)**, pp. 107–113.

[20] J. Dean and S. Ghemawat, ―Mapreduce: A flexible data processing tool‖, commun. ACM, vol. 53, no. 1, **(2010)**, pp. 72–77.