International Journal of Computer Science
Scholarly Peer Reviewed Research Journal - PRESS - OPEN ACCESS
ISSN: 2348-6600
SKRGC Since 2012

http://www.ijcsjournal.com Volume 5, Issue 1, No 2, 2017 ISSN: 2348-6600
Reference ID: IJCS-164 PAGE NO: 995-1003

Sri Vasavi College, Erode Self-Finance Wing 3rd February 2017
National Conference on Computer and Communication NCCC'17
http://www.srivasavi.ac.in/ nccc2017@gmail.com

Bigdata Analytics: Comparative Study of Tools

H. Vignesh Ramamoorthy[#1], G. Murugesan[*2]

[#1]Assistant Professor, Department of Computer Science

[*2]Assistant Professor, Department of Information Technology

Sree Saraswathi Thyagaraja College, Pollachi – 642107, Tamil Nadu, India

[#1]hvigneshram@gmail.com, [*2]murugesan_g29@yahoo.com

ABSTRACT

The term “Big Data” was first introduced to the computing world by Roger Magoulas from O’Reilly media in 2005 in order to define a great amount of data that traditional data management techniques cannot manage and process due to the complexity and size of this data.

A study on the Evolution of Big Data as a Research and Scientific Topic shows that the term “Big Data” was present in research starting with 1970s but has been comprised in publications in 2008. Nowadays the Big Data concept is treated from different points of view covering its implications in many fields.

In the information era, enormous amounts of data have become available on hand to decision makers. Big data refers to datasets that are not only big, but also high in variety and velocity, which makes them difficult to handle using traditional tools and techniques. Due to the rapid growth of such data, solutions need to be studied and provided in order to handle and extract value and knowledge from these datasets. Furthermore, decision makers need to be able to gain valuable insights from such varied and rapidly changing data, ranging from daily transactions to customer interactions and social network data. Such value can be provided using big data analytics, which is the application of advanced analytics techniques on big data. This paper aims to analyze some of the different analytics methods and tools which can be applied to big data, as well as the opportunities provided by the application of big data analytics in various decision domains.

Keywords—Big data, Big data analytics, Business analytics, Bid data tools.

I. INTRODUCTION

The term “Big Data” was first introduced to the computing world by Roger Magoulas from O’Reilly media in 2005 in order to define agreat amount of data that traditional data management techniques cannot manage and process due to the complexity and size of this data. A study on the Evolution of Big Data as a Research and Scientific

Topic shows that the term "Big Data" was present in research starting with 1970s but has been comprised in publications in 2008. Nowadays the Big Data concept is treated from different points of view covering its implications in many fields. According to MiKE 2.0, the open source standard for Information Management, Big Data is defined by its size, comprising a large, complex and independent collection of data sets, each with the potential to interact. In addition, an important aspect of Big Data is the fact that it cannot be handled with standard data management techniques due to the inconsistency and unpredictability of the possible combinations.

Big Data has four aspects as shown in fig 1:

Volume: refers to the quantity of data gathered by a company. This data must be used further to obtain important knowledge;

Velocity: refers to the time in which Big Data can be processed. Some activities are very important and need immediate responses, that are why fast processing maximizes efficiency;

Variety: Refers to the type of data that Big Data can comprise. This data can be structured as well as unstructured;

Veracity: refers to the degree in which a leader trusts the used information in order to take decision. So getting the right correlations in Big Data is very important for the business future.

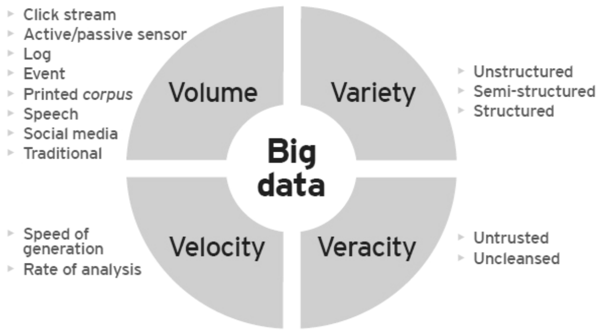


Fig 1. V's of Big Data

The amount of data stored in various sectors can vary in the data stored and created, i.e., images, audio, text information etc., from one industry to another. From the practical perspective, the graphical interface used in the big data analytics tools leads to be more efficient, faster and better decisions which are massively preferred by analysts, business users and researchers [1].

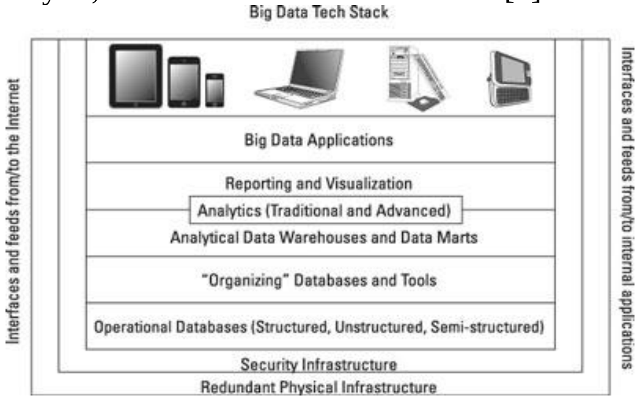

Big Data Tech Stack
Big Data Applications
Reporting and Visualization
Analytics (Traditional and Advanced)
Analytical Data Warehouses and Data Marts
"Organizing" Databases and Tools
Operational Databases (Structured, Unstructured, Semi-structured)
Security Infrastructure
Redundant Physical Infrastructure
Interfaces and feeds from/to the Internet
Interfaces and feeds from/to internal applications

Fig 2. Big Data Architecture

Here's a closer look at what's in the image and the relationship between the components:

- **Interfaces and feeds:** On either side of the diagram are indications of interfaces and feeds into and out of both internally managed data and data feeds from external sources. To understand how big data works in the real world, start by understanding this necessity.

- **Redundant physical infrastructure:** The supporting physical infrastructure is fundamental to the operation and scalability of a big data architecture. Without the availability of robust physical infrastructures, big data would probably not have emerged as such an important trend.
- **Security infrastructure:** The more important big data analysis becomes to companies, the more important it will be to secure that data. For example, if you are a healthcare company, you will probably want to use big data applications to determine changes in demographics or shifts in patient needs.
- **Operational data sources:** When you think about big data, understand that you have to incorporate all the data sources that will give you a complete picture of your business and see how the data impacts the way you operate your business.

II.DIFFERENT BIG DATA TOOLS

A. Hadoop

The name Hadoop has become synonymous with big data. It's an open-source software framework for distributed storage of very large datasets on computer clusters. All that means you can scale your data up and down without having to worry about hardware failures. Hadoop provides massive amounts of storage for any kind of data, enormous processing power and the ability to handle virtually limitless concurrent tasks or jobs.

Hadoop is not for the data beginner. To truly harness its power, you really need to know Java. It might be a commitment, but Hadoop is certainly worth the effort – since tons of other companies and technologies run off of it or integrate with it. Hadoop platform involves a cluster of storage/computing nodes (or machines) out of which one node is assigned as master and other as the slave nodes. The HDFS [18] maintains each file in the chunk of same size blocks (except the last block). Also, various replicas of these blocks are maintained on various nodes in the cluster for the sake of reliability and fault tolerance. The Map-Reduce [19-20] computing technique divides the whole task of processing into smaller blocks and assign to various slave machines where the required data is available and executes computing right at that node. In this way it saves significant time and cost involved in transferring data from data server to the computing machine.

Following are the advantages, disadvantages and latest version of Hadoop.

i. Advantages of Hadoop

- Open source: Being an open source, Hadoop is freely available [3].
- 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 [3].
- Scalable: To handle larger data, the
- Hadoop is capable to scale linearly by putting additional nodes in clusters [3].
- Fault Tolerant and Robust: It replicates data block on multiple nodes that facilitates the recovery from a single node or machine failure. Also, Hadoop's architecture deals with frequent malfunctions in hardware. If a node fails the task of that node is reassigned to some other node [4].
- High Throughput: Due to batch processing high throughput is achieved in Hadoop [4].
- Portability: Hadoop architecture can be effectively ported [5] while working with several commodities of operating systems and hardwares that may be heterogeneous [6].

ii. Disadvantages of Hadoop

Single Point Failure: Hadoop's (version up to 2.x) HDFS as well as MapReduce suffer from master level single points of failure [7].

- Low Efficiency/ Poor Performance than DBMS [7]: Hadoop shows lower efficiency due its inability to switch to the next stage before completing the previous stage tasks causing Hadoop unsuitable for pipeline parallelism, runtime scheduling that causes degraded efficiency per node. Unlike RDBMS, it has no specific optimization of execution plans that could minimize the transfer of data among various nodes.
- Inefficient Dealing with Small Files: As HDFS is meant for high throughput optimization [8], it does not suit to random reads on small files [9].
- Not Suitable for Real Time Access: MapReduce and HDFS employ batch processing architecture hence; it does not fit for real-time accesses [8].

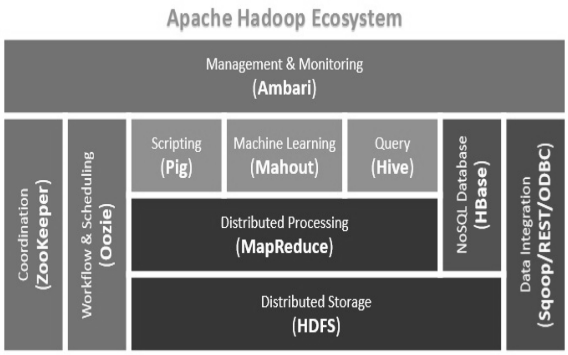


Fig 3. Hadoop Architecture

I. B. CASSANDRA

Apache Cassandra is a free and open-source distributed database management system designed to handle large amounts of data across many commodity servers, providing high availability with no single point of failure. Cassandra offers robust support for clusters spanning multiple datacenters,[1] with asynchronous masterless replication allowing low latency operations for all clients.

Cassandra also places a high value on performance. In 2012, University of Toronto researchers studying NoSQL systems concluded that "In terms of scalability, there is a clear winner throughout our experiments. Cassandra achieves the highest throughput for the maximum number of nodes in all experiments" although "this comes at the price of high write and read latencies.

A. Main features

- Decentralized

  Every node in the cluster has the same role. There is no single point of failure. Data is distributed across the cluster (so each node contains different data), but there is no master as every node can service any request.

- Supports replication and multi data center replication

  Replication strategies are configurable.[17]Cassandra is designed as a distributed system, for deployment of large numbers of nodes across multiple data centers.

- Scalability

  Read and write throughput both increase linearly as new machines are added, with no downtime or interruption to applications.

- Fault-tolerant

  Data is automatically replicated to multiple nodes for fault-tolerance. Replication across multiple data centers is supported. Failed nodes can be replaced with no downtime.

- Tunable consistency

  Writes and reads offer a tunable level of consistency, all the way from "writes never fail" to "block for all replicas to be readable", with the quorum level in the middle.[10]

- MapReduce support

  Cassandra has Hadoop integration, with MapReduce support. There is support also for Apache Pig and Apache Hive

C. CouchDB

Apache CouchDB is open source database software that focuses on ease of use and having an architecture that "completely embraces the Web".[11] It has a document-oriented NoSQL database architecture and is implemented in the concurrency-oriented language Erlang; it uses JSON to store data, JavaScript as its query language using MapReduce, and HTTP for an API.[11] CouchDB was first released in 2005 and later became an Apache Software Foundation project in 2008. Unlike a relational database, a CouchDB database does not store data and relationships in tables. Instead, each database is a collection of independent documents. Each document maintains its own data and self-contained schema. An application may access multiple databases, such as one stored on a user's mobile phone and another on a server. Document metadata contains revision information, making it possible to merge any differences that may have occurred while the databases were disconnected. CouchDB implements a form of multiversion concurrency control (MVCC) so it does not lock the database file during writes. Conflicts are left to the application to resolve. Resolving a conflict generally involves first merging data into one of the documents, then deleting the stale one [11].

B.

C. Main features

- ACID Semantics

CouchDB provides ACID semantics.[11] It does this by implementing a form of Multi-Version Concurrency Control, meaning that CouchDB can handle a high volume of concurrent readers and writers without conflict.

- Built for Offline

CouchDB can replicate to devices (like smartphones) that can go offline and handle data sync for you when the device is back online.

- Distributed Architecture with Replication

CouchDB was designed with bi-direction replication (or ynchronization) and off-line operation in mind. That means multiple replicas can have their own copies of the same data, modify it, and then sync those changes at a later time.

- Document Storage

CouchDB stores data as documents", as one or more field/value pairs expressed as JSON. Field values can be simple things like strings, numbers, or dates; but ordered lists and associative arrays can also be used. Every document in a CouchDB database has a unique id and there is no required document schema.

- Eventual Consistency

CouchDB guarantees eventual consistency to be able to provide both availability and partition tolerance.

- Map/Reduce Views and Indexes

The stored data is structured using views. In CouchDB, each view is constructed by a

JavaScript function that acts as the Map half of a map/reduce operation.

- HTTP API

All items have a unique URI that gets exposed via HTTP. It uses the HTTP methods POST, GET, PUT and DELETE for the four basic CRUD (Create, Read, Update, Delete) operations on all resources.

D. MongoDB

MongoDB is a free and open-source cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schemas. MongoDB is developed by MongoDB Inc. and is free and open-source, published under a combination of the GNU Affero General Public License and the Apache License.

D. Main features

- Ad hoc queries

MongoDB supports field, range queries, regular expression searches [12]. Queries can return specific fields of documents and also include user-defined JavaScript functions. Queries can also be configured to return a random sample of results of a given size.

- Indexing

Fields in a MongoDB document can be indexed with primary and secondary indices.

- Replication

MongoDB provides high availability with replica sets [12].A replica set consists of two or more copies of the data. Each replica set member may act in the role of primary or secondary replica at any time. All writes and reads are done on the primary replica by default. Secondary replicas maintain a copy of the data of the primary using built-in replication. When a primary replica fails, the replica set automatically conducts an election process to determine which secondary should become the primary. Secondaries can optionally serve read operations, but that data is only eventually consistent by default.

- Load balancing

MongoDB scales horizontally using sharding [12]. The user chooses a shard key, which determines how the data in a collection will be distributed. The data is split into ranges (based on the shard key) and distributed across multiple shards. (A shard is a master with one or more slaves.). Alternatively, the shard key can be hashed to map to a shard – enabling an even data distribution.

MongoDB can run over multiple servers, balancing the load or duplicating data to keep the system up and running in case of hardware failure.

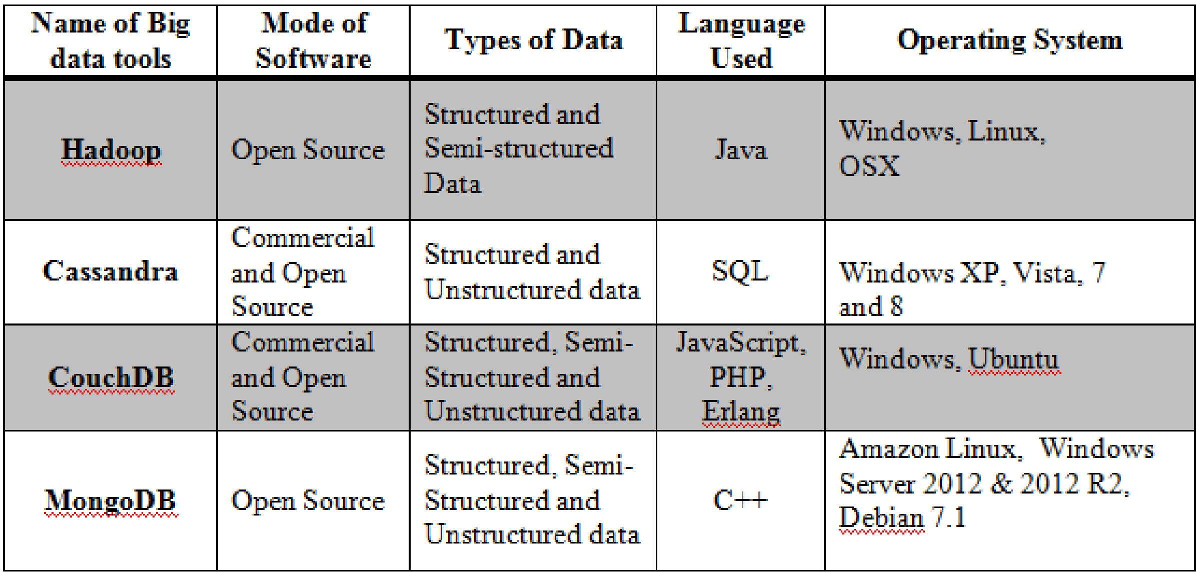
E. Architecture

- Programming language accessibility

MongoDB has official drivers for a variety of popular programming languages and development environments [12].There are also a large number of unofficial or community-supported drivers for other programming languages and frameworks [12].

III. SUMMARIZATION OF BIG DATA TOOLS

The following Table I demonstrate the comparative aspects of the diverse tools in big data based on compatible data sources and its operating system. The main objective of this comparison is not to criticize which is the best tool in big data, but to demonstrate its usage and to create alertness in various fields.



Name of Big data tools	Mode of Software	Types of Data	Language Used	Operating System
Hadoop	Open Source	Structured and Semi-structured Data	Java	Windows, Linux, OSX
Cassandra	Commercial and Open Source	Structured and Unstructured data	SQL	Windows XP, Vista, 7 and 8
CouchDB	Commercial and Open Source	Structured, Semi-Structured and Unstructured data	JavaScript, PHP, Erlang	Windows, Ubuntu
MongoDB	Open Source	Structured, Semi-Structured and Unstructured data	C++	Amazon Linux, Windows Server 2012 & 2012 R2, Debian 7.1

TABLE I: COMPARISON OF BIG DATA TOOLS

IV. CONCLUSION

In this paper, more than a few big data tools were elucidated along with their features of several tasks. Big data provide vastly effective supporting processes for collection of data sets which is too complex and large. This mandatory requirement gives the way for developing many tools in big data research. Whereas these data base tools are generated both in real time and also in very large scale which comes from sensors, web, networks, audio/video, etc. Thus the aim of this survey is to enhance the knowledge in big data tools and their applications applied in various companies. It also provides obliging services for readers, researches, business users and analysts to make enhanced and quicker decisions using data which will promote for development and innovation in the future.

REFERENCES

[1] http://www.slideshare.net/HarshMishra3/ harsh-big-data-seminar-report

[2] http://www.infoworld.com/d/business-intelligence/7-top-tools-tamingbig- data-191131.

[3] J. Venner, ―Pro Hadoop‖, a press, **(2009)**.

[4]T. White,‖ Hadoop: The Definitive Guide‖, third ed., O'Reilly Media, Yahoo Press, **(2012)**.

[5] 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

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

[7] 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)**.

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

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

[10] Hewitt, Eben (December 15, 2010). Cassandra: The Definitive Guide (1st ed.).

[11] Brown, MC (October 31, 2011), Getting Started with CouchDB (1st ed.), O'Reilly Media.

[12] Pirtle, Mitch (March 3, 2011), MongoDB for Web Development (1st ed.), Addison-Wesley Professional.