 ISBN: 978-81-927230-0-9

# MongoDB vs Hadoop

Mrs.Rohini Gaikwad,
Research Scholar
Vidya Pratishthan's Institute of Information Technology
Savitribai Phule Pune University
Baramati, Pune, Maharashtra-413133.

Dr. A.C.Goje
Research Guide
Vidya Pratishthan's Institute of Information Technology
Savitribai Phule Pune University
Baramati, Pune, Maharashtra-413133.

***Abstract - Big Data is a data whose scale, variety, and complexity require new structural design, techniques, algorithms, and analytics to manage it and pull out value and unseen knowledge from it. To process the large amounts of data in an economical and proficient way, parallelism is used.To handle the unstructured and semistructured data NOSQL databse has been introduced. Another platform Hadoop is introduced for big data analytics purposes. It is designed to scale up from a single server to thousands of machines, with a very high degree of fault tolerance.***

***In this paper, Author reviewed the contributions of researchers in the area of Mongodb and Hadoop. Both are having their significance in their own way. Author focusing on the capabilities of these technologies ,their use cases in real world. And concluded with the future of integration of these technology.***

***Keyword: NOSQL, Hadoop, Big data,Use cases***

## I. INTRODUCTION

In recent years there has been bang of data. Big Data refers to large sets of data that cannot be analyzed with traditional tools. It stands for data related to large-scale processing architectures.

Gartner defines Big Data as high volume, velocity and variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision making.

According to IBM, 80% of data captured today is unstructured, from sensors used to gather climate information, posts to social media sites, digital pictures and videos, purchase transaction records, and cell phone GPS signals, to name a few. All of this unstructured data is Big Data. The various operations are used for the data processing .Data is generated from the many sources in the form of structured as well as unstructured form . The two main problems regarding big data are the storage capacity and the processing of the data.

For handling such a huge data the need of new technologies such as NOSQL, Hadoop much more is arises. But Still there is debate on the performance related issues of these various NOSQL databases.

## II. MONGODB

MongoDB is an open-source document database that provides high performance, high availability, and automatic scaling.

MongoDB was developed by 10gen and was initially released in 2009. It was developed using C++. It is a high performance and efficient database. It provides features like consistency fault tolerance, persistence. MongoDB provides additional features like aggregation, ad hoc queries, indexing, auto sharding etc. In MongoDB the documents are mainly stored in BSON (Binary JSON) format. BSON documents contain an ordered list of elements consisting of field name, type and value. BSON is efficient both in storage space and scan speed when compared to JSON MongoDB uses GridFS as a specification for storing large files. MongoDB is well suited for applications like content management systems, archiving, real time analytics etc. MongoDB is currently
being used by MTV networks, Foursquare, The Guardian etc. It is also being used in projects like CERN‟s LHC, UIDAI Aadhaar which is India's unique identification project. The disadvantages are that it can be unreliable and indexing takes up lot of RAM [6][7].

## III. HADOOP

Hadoop [4], the open-source implementation of Google's MapReduce [12], has become a commonly used tool for Big Data analytics.

Hadoop was an open-source project from the start; created by Doug Cutting(known for his work on Apache Lucene, a popular search indexing platform), Hadoop originally stemmed from a project called Nutch, an open-source web crawler created in 2002. Over the next few years, Nutch followed very closely at the heels of different Google Projects; in 2003, when Google released their Distributed File System (GFS), Nutch released their own, which was called NDFS. In 2004, Google introduced the concept of MapReduce, with Nutch announcing adoption of the MapReduce architecture shortly after in 2005. It wasn't until 2007 that Hadoop was officially released. Using concepts carried over from Nutch, Hadoop became a platform for parallel processing mass amounts of data across clusters of commodity hardware.

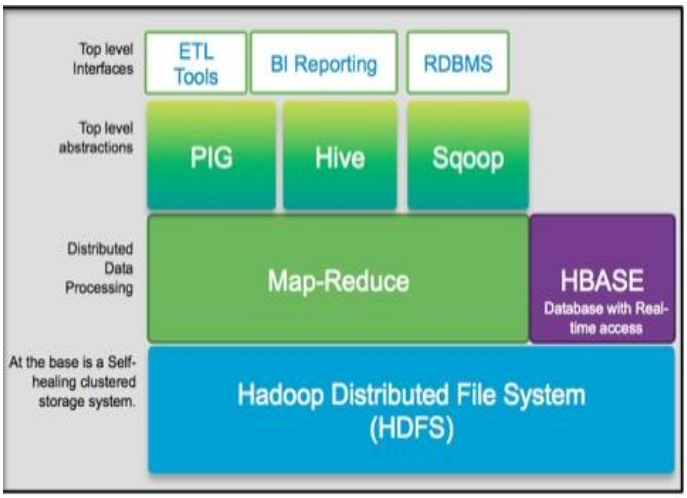


Fig1 : **Architecture of Hadoop** [9]

## IV. USE CASES

**a) Mongodb:**

Operational Intelligence:

As an introduction to the use of MongoDB for operational intelligence and real time analytics use, the document "***Storing Log Data***" describes several ways and approaches to modeling and storing machine generated data with MongoDB. Then, "***Pre-Aggregated Reports***" describes methods and strategies for processing data to generate aggregated reports from raw event-data. Finally "***Hierarchical Aggregation***" presents a method for using MongoDB to process and store hierarchical reports (i.e. per-minute, per-hour, and per-day) from raw event data.

Product Data Management:

MongoDB's flexible schema makes it particularly well suited to storing information for product data management and e-commerce websites and solutions. The "*Product Catalog*" document describes methods and practices for modeling and managing a product catalog using MongoDB, while the "*Inventory Management*" document introduces a pattern for handling interactions between inventory and users' shopping carts. Finally the "*Category Hierarchy*" document describes methods for interacting with category hierarchies in MongoDB.

Content Management Systems:

The content management use cases introduce fundamental MongoDB practices and approaches, using familiar problems and simple examples. The "***Metadata and Asset Management***" document introduces a model that you may use when designing a web site content management system, while "***Storing Comments***" introduces the method for modeling user comments on content, like blog posts, and media, in MongoDB.

b)Hadoop:

ETL: for Unstructured data streaming in real time,Hadoop is a good choice to structure the dataand then store it.Hadoop provides way to preprocess data prior.

Machine Education: Apache Mahout is built on top of Hadoop and essentially works along with it to facilitate targeted trade in e-commerce.

Log Processing: Log Files are usually very large and there are typically lots of them. These creates huge amount of data. Hadoop is the best answer to this problem; splitting the log into smaller workable chunks and assigning them to workers results in very fast processing.

## V. COMPARISON OF MONGODB AND HADOOP

- Mongodb[2] is a nosql –document store database, hadoop is a data processing and analysis framework.
- Mongodb focuses on efficient retrival of data,hadoop focuses on data processing.
- Mongodb has its own map reduce framework,hadoop has hbase.
- Mongodb[3] can certainly be considered a big data solution, it's worth noting that it's really a general-purpose platform, designed to replace or enhance existing rdbms systems, giving it a healthy variety of use cases. While hadoop has a specific purpose, and is not meant as a replacement for transactional rdbms systems, but rather as a supplement to them, as a replacement of archiving systems, or a handful of other use cases.
- Mongodb[2] preallocates space for storage,improving performance but wasting space. Hadoop optimizes space usage, but ends up with lower write performance .
- Mongodb[5] is used with systems less than approximately 5TB of data.Hadoop has been used for systems larger than 100TB of data.
- Using Hadoop [11]for MapReduce jobs is several times faster than using the built-in MongoDB MapReduce capability.

## VI. Mangodb+Hadoop [14]

MongoDB is used as the "operational" real-time data store and Hadoop is used for offline batch data processing and analysis.

Batch Aggregation

In several scenarios the built-in aggregation functionality provided by MongoDB is sufficient for analyzing your data. However in certain cases, significantly more complex data aggregation may be necessary. This is where Hadoop can provide a powerful framework for complex analytics.

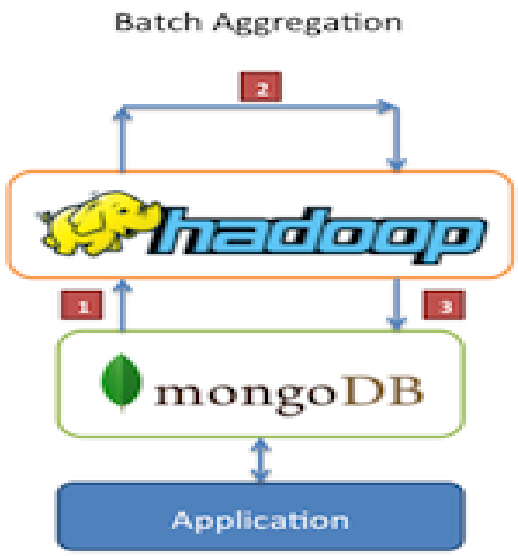


Fig2: Batch Aggregation[14]

Data Warehouse

In a typical production scenario, your application's data may live in multiple datastores, each with their own query language and functionality. To reduce complexity in these scenarios, Hadoop can be used as a data warehouse and act as a centralized repository for data from the various sources.

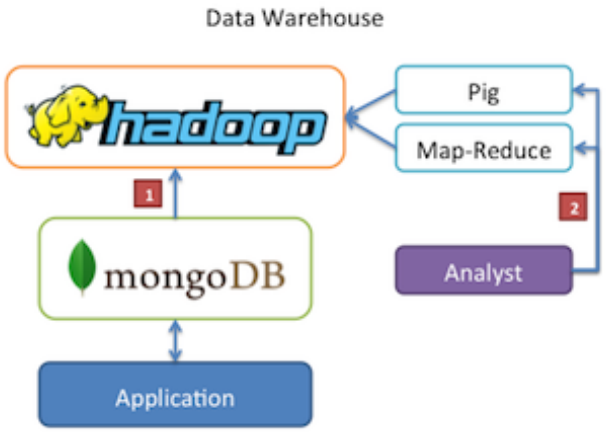


Fig 3:Data Warehouse[14]

ETL Data:

MongoDB may be the operational datastore for application but there may also be other datastores that are holding organization's data. In this scenario it is useful to be able to move data from one datastore to another, either from application's data to another database or vice versa. Moving the data is much more complex than simply piping it from one mechanism to another, which is where Hadoop can be used.

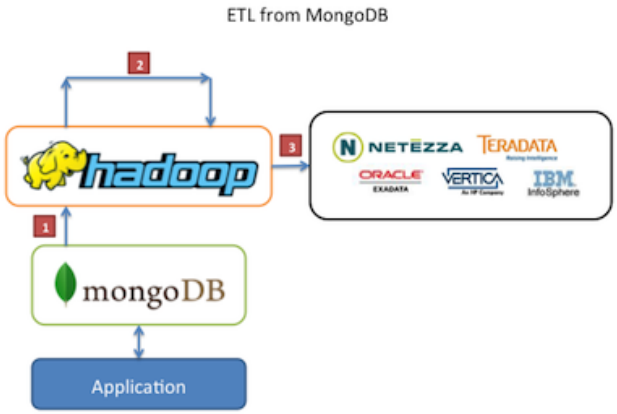


Fig4: ETL from Mongodb[14]

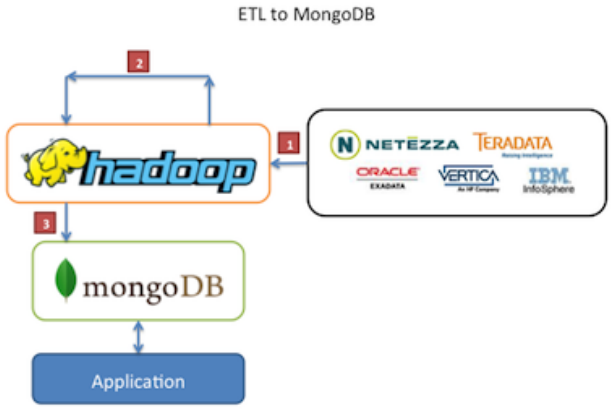


Fig 5: ETL to Mongodb[14]

## VII. FINDINGS

The summary of the features of both Mongodb and Hadoop is mentioned below in table 1:[8]

| Apache Hadoop | MongoDB |
|---|---|
| **Written In:** Java | **Written In:** C++ |
| **Open Source:** Yes | **Open Source:** Yes |
| **Type:** Framework | **Type:** Database |
| **OS:** Cross-Platform | **OS:** Cross-Platform |
| **State:** Suite of Products | **State:** Stand-Alone Product |
| **Best Application:** Large Scale Processing | **Best Application:** Real-Time Extraction and Processing |
| **MapReduce:** Available | **MapReduce:** Available |
| **Scalability:** Limited | **Scalable:** Yes |
| **NoSQL:** No | **NoSQL:** Yes |
| **High Availability:** Limited – Only One Failure Point Available | **High Availability:** Yes – Replication Enabled |
| **Storage:** File System Available | **Storage:** File System Available |
| **Server-Side Script Execution:** Yes | **Server-Side Script Execution:** Yes |
| **Data Structure:** Flexible | **Data Structure:** Only CSV and JSON can be imported |
| **Max rows in Query Result**: 9 999 999 999 | **Max rows in Query Result** :1000 |

## VIII. CONCLUSION

Managing big data with just a single tool or framework is nearly impossible as stated earlier. That is to say that technologies such as Hadoop and MongoDB must be utilized together. Clearly, one way or another both Hadoop and MongoDB are taking the place of conventional RDBMS. With all considerations and suggestions noted, it is also very important to know that neither MongoDB nor Hadoop are built to boast security. Both the technologies are meant to manage large data and both come with benefits and some of the limitations.

## IX. ACKNOWLEDGEMENT

I would like to say thanks to all researchers whose work directly & indirectly help me in my work.

## REFERENCES


1. NoSQL - http://nosql-database.org/.
2. ”Big Data- Mongodb Vd Hadoop: Big solutions for big problems” Deep Mistry,Open Source Software Integrators.
3. http://www.aptude.com/blog/entry/hadoop-vs-mongodb-which-platform-is-better-for-handling-big-data
4. Apache Foundation. Hadoop. http://hadoop.apache.org/.
5. http://developer.yahoo.com/blogs/hadoop/posts/2008/09/scaling_hadoop_to_4000_nodes_a/
6. “Type of NOSQL Databases and its Comparison with Relational Databases ” Ameya Nayak ,Anil Poriya , Dikshay Poojary Dept. of Computer Engineering Thakur College of Engineering and Technology .University of Mumbai . *International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868 Foundation of Computer Science FCS, New York, USA Volume 5– No.4, March 2013 – www.ijais.org*
7. www.mongodb.org
8. http://www.happiestminds.com/blogs/harnessing-the-big-data-hadoop-vs-mongodb/
9. ” Big Data – Hadoop from an Infrastructure Perspective” Arun Chakravarti,Cisco [2012]
10. http://en.wikipedia.org/wiki/Database_transaction
11. ” Performance Evaluation of a MongoDB and Hadoop Platform for Scientific Data Analysis” E. Dede, M. Govindaraju SUNY Binghamton Binghamton, NY 13902 {edede,mgovinda}@cs.binghamton.edu.
12. J. Dean and S. Ghemawat. MapReduce: Simplified data processing on large clusters. In OSDI, 2004.
13. https://docs.mongodb.com/ecosystem/use-cases/
14. https://docs.mongodb.com/ecosystem/use-cases/hadoop/