www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 4 Issue 8 Aug 2015, Page No. 13635-13639

An Approach to Implement Map Reduce with NoSQL Databases

Ashutosh Singh Chauhan[1], Anjali Kedawat[2], Pooja Parnami[3]

[1]Department of computer Science,
Amity University, Jaipur, Rajasthan, India,
ashu.smec@gmail.com

[21]Department of computer Science,
Amity University, Jaipur, Rajasthan, India,
akedawat@amity.edu

[31]Department of computer Science,
Amity University, Jaipur, Rajasthan, India,
pparnami@jpr.amity.edu

Abstract: MapReduce is a well-known programming model and an implementation method for executing, processing and generating massive data sets. MapReduce algorithm consists of a map function that processes a key/value pair to produce a set of intermediate key/value pairs, and a reduce function which combine all these values related with the same intermediate key. MapReduce executes in parallel itself without implementing any parallel programming model and it is the most efficient way to process unstructured data.
In this research MapReduce algorithm is implemented on a cluster based machine using Hadoop distributed file system (HDFS) in order to perform a Pattern matching algorithm for different volumes of datasets. The quantitative performance analysis of MapReduce algorithm is done for the different volumes of data on the basis of execution time and number of patterns searched.

So far relational databases are used for storing the data for the applications but now there is need to store huge amount of data to store and manage which cannot stored by relational databases. NoSQL technology over comes this problem. This research paper provides a brief introduction to NoSQL database working and comparative study between MongodB and CouchDB, Which are mostly used for big data application. The operations are performed to explore the results as distinguish between both NoSql databases. This paper shows the performance of Mongodb and CouchDB. Results proves that CouchDB is more powerful than Mongodb to load and process on big data and processing very fast as compare to Mongodb. This paper describes the functionality of Mongodb and CouchDB over the large dataset.

Keywords: NoSql Databases, Mongodb, CouchDB, Big Data.

1. INTRODUCTION

NoSql Technology: it stands for Not only SQL. NoSql technology architecture provides the facility to use various type of databases architectures like document databases, graph databases, key value databases which are used in different ways as document store database, graph visualization and key value pair [1]. This technology solve the problem of storing the massive datasets and unstructured data which is not possible in Rdbms.
This technology provides the scalability and availability. NoSQL does not have any type of deterministic architecture and permits schema migration without downtime. It solves the problem of Rdbms.

Figure 1: Popular NoSQL Databases

1.1 Document Database: This databases store the data in disc as document format. This solves the limitation of storing the data into table format in Rdbms. It also stores unstructured data that are not lettered in format of table. It stores into document and can store large amount of data. In this it is very

easy to store the massive data and easy to handle documents they does not need more joins to connect the documents because data are stored in form of arrays [2].
Document database can be performed on web platform and give the best results even the incomplete data is given. It uses the dynamic schema to handle polymorphism.

1.1.1 Mongodb: Mongodb database is very popular document database which is known for high performance, high availability and scalability. This database can embed easily and have capacity to read and write data fast [3].
This database uses the indexes from the document and arrays to perform the faster. It gives the high availability for higher performance and very easy to scale and easy to manage the operations.
It stores the data into documents and collections not in table format as rows and columns. Collections represents the relationship in a meaningful way that is very easier to understand. Documents are stored in disc so no problem to store large amount of data. It can also store unstructured form of data.
It can perform several operations on documents and collection which is not possible in Rdbms [3].
There are some following features of MongoDB

- Mongo DB supports Map reduce and Aggregation Tools.
- Java scripts can be embed into Mongo DB.
- It is a schema less Document based database.
- It uses secondary indexes and geospatial indexes to perform faster.
- Easy to manage the Mongo DB in cases of failures.
- Mongo DB designed to provide high available and high performance.
- It can store large amount of data into disc.

Figure 2: MongoDB Features

1.1.2 CouchDB: CouchDB is a database that completely uses the web platform. It stores data with JSON format documents and very easy to access documents and query on indexes with a web browser, via HTTP. Java script is used to indexing, combine and transform documents. CouchDB works well with modern web platform. CouchDB can be used to distribute data, efficiently using CouchDB's incremental replication. It supports master-master setups with conflict detection automatically [4].

CouchDB provide several features, such as on-the-fly document transformation and real-time update notifications, which makes easier to handle data. It gives administration console for query analyzer. It is highly available, scalable and partition tolerant. It is also consistent. CouchDB has a fault-tolerant storage database that gives the safety about the data.
Couchdb uses the several components to provide the best performance on the dataset.

a) Document Storage

CouchDB stores the documents on the host server. Each document is uniquely named in the database, and Rest API can be used to read and update the document on server in couchDB.
Documents are as primary unit of data in CouchDB and having any number of fields and attachments. Documents also include metadata description which is maintained through the database system. Document fields are uniquely defined and contain values of different types (text, Boolean, lists, number etc.), and there is no limit set for text size or element count [4].

Its update model is lockless and optimistic. Client application edits the documents like loading the documents, make some changes on document and saving the documents. If another client is editing the same document and save the changes, then client faces the problem of confliction. To solve the update conflict, the latest document version will be opened, the changes are applied after saving.

Document updates like add, edit and delete operations are all or nothing, either succeeding completely or failing completely. The database never has partially saved or partially updated document [5].

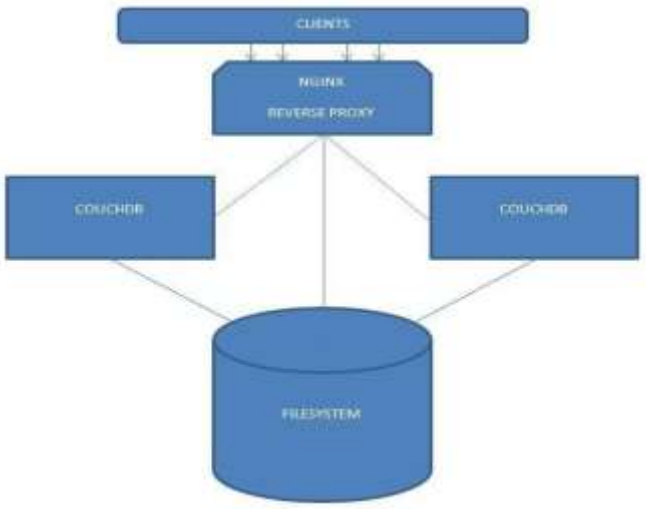

Figure 3: CouchDB Architecture

b) ACID Properties

CouchDB supports the Atomicity, consistency, isolation and durability. These properties maintain the database perfect and prevent from failure. This database does not overwrite saved data or associated structures, ensures that the database file is in a consistent state. There is no different replica of any document. If the couchDB server crashes or shut down then no document will be in saved state.

c) Operations

Document updates like add, edit and delete operations are serialized, except for binary blobs that are written concurrently. Database readers never have to wait for writers to save their documents or other readers and are no need locked out. Any number of readers can read the documents without any wait or being locked out or interrupted through any update or change, even on the same document.

Data are stored in the documents as json format so there is no need to follow the table format as row and column format. Documents can be saved at anywhere in disk.

When documents are saved, all data and their indexes are updated to disk and the transactional commit leaves the database in a proper consistent state.

1.2 JSON

JavaScript Object Notation extended form of the JavaScript scripting language. It is very easy to perform read and write operation on document. It is very lightweight text based interchangeable format. JSON format is not dependent and can be embed in any other language easily. It is used as document file for Mongodb database.

```
{
c_id:"1",
c_name:"Ashu",
value:700,
status:"1"
}
{
c_id:"2",
c_name:"Ankit",
price:800,
status:"1"
}.
```

Above document file is json format and having information of customer. These are separated in row format by using curly braces. It is easy to read and understand.

1.3 MapReduce

MapReduce is a framework for parallel processing to the analysis on big data on several servers. It was invented by the Google for the backend part of search engine to make capable large number of commodity servers to process efficiently to analysis of large numbers of webpage files collected from all over the world. Firstly this MapReduce is implemented by Apache into a project to process parallel, it was published as open source over the world and it enabled so many organizations, such as businesses and universities, to handle big data analysis.

MapReduce follows the principle of parallel and distributed processing developed by Google and it is used to perform operations on big data to analytics. It was adopted by many organizations for analyzing massive datasets on Hadoop. MapReduce [6] is a simple programming model for processing large data in parallel manner. It has master-slave based architecture. MapReduce divides the tasks into sub tasks then perform operations in parallel way then aggregate the final output to produce the best results.

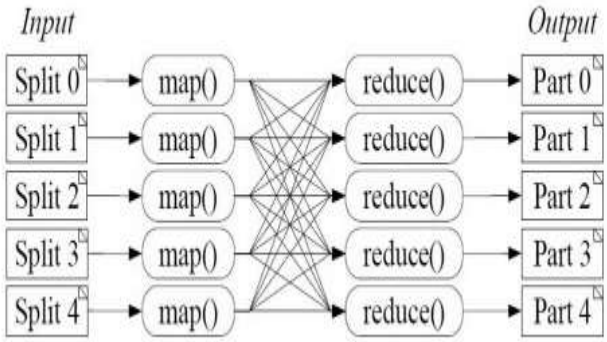

Figure 4: MapReduce Architecture

In most computation on big data, it is calculated that two main parts are commonly used which are shown in above figure 3.

MapReduce has two main functions Map and Reduce. Map function is used to map the data and assign the key to individual module and reduce produce the result after filteration.

2. LITERATURE SURVEY

In 2012, University of Toronto researchers studying NoSQL systems concluded that Cassendra is most powerful in terms of scalability throughout the experiments. It achieves the maximum throughput for the number of nodes. "It comes at the price of high write and read latencies [7] Mongodb database stores the large amount of data in JSON format and performs the operations that will give results quickly. But they did not compare MongoDB with Apache CouchDB.

3. PROBLEM STATEMENT

Traditional Relational databases are not able to store the data with relationships or the data that consists unstructured records. RDBMS also lacks the capacity to store the Billions of records of Terabytes in size. Most popular RDBMS MySQL can store upto 25 GB of data while most advance PostGres Store upto 35 GB of Data. Relational databases takes infinite time to process the data with several relations as they process them using several joins which are very expensive in terms of CPU consumption.

Here NoSQL database comes as a solution but the difficulty is that there are more than 126 NoSQL databases [9]. Now to select the right database is a problem. This research compares one of the most popular NoSQL databases and reveals that which NoSQL database is better in which scenario while storing and processing data as JSON.

4. PROPOSED APPROACH

Mongodb and CouchDB both are NoSQL databases which are used when data is huge. Here JSON file is used to store large amount of data. On which Mongodb operations are performed such as MapReduce, Insertion, Deletion and Updation.

In case of CouchDB there is database created by using the CQL (CouchDB Query Language). This database contains the same data as JSON file has.

There are for collections created in Mongodb. First collection contains 50k records, second collection contains 100k records, third collection contains 500k reocrds and fourth collection contains the 1000k records in it.

CouchDB follows the same scenario that uses four tables and having same amount of records as Mongodb has.

Any platform can be used to perform the operations on MongoDB and CouchDB. These databases provide the high speed and high throughput as compare to relational databases.

3.1 Setting up The Environment

In this research, all the tests are performed under following specifications:

1) **Host System:** Intel i5 core processor with 6 GB RAM and 1000 GB Hard disk.
2) **Operating System:** Microsoft Windows 8.1
3) **Mongo DB:**
4) **CouchDB:**

a) **Execution Time:** Execution time can be defined in terms of time consumed by an algorithm in order to solve a problem using processor p.

3.2 MapReduce using MongoDB & CouchDB

Consider the following document which is storing the information of customer. The document stores c_id, name, price and status of the customer.

```
{
c_id:"2",
c_name:"Ravi",
price:652,
status:"1"
}
{
c_id:"5",
c_name:"Hari",
price:522,
status:"1"
}
```

Now, we will execute a mapReduce query on document to select all the customer's information who has active status, group them on the basis of c_id and then calculate the sum of values of data by each user using the following code:

```
db.data.mapReduce
(
 function()
 {
 emit (this.c_id, this.value);
 },
 function(key, values) {return Array.sum(values)},
 {
 query: { status:"1"},
 out:"sum"
 }
)
```

This operation will return the output that will help us to analyze the performance of MongoDB and CouchDB document store databases.

5. CONCLUSION

For the right analysis we need to analyze the performance in terms of scalability also. Achieving the highest throughput for the maximum number of nodes in all experiments that comes at the price of high write and read latencies is also a major factor for the analysis.

As the number of records in document database increases, the difference between the execution time taken by different databases for the computation of different database operations is what we are looking for.

For the data retrieval operation, data updation, data creation operation and data deletion the performance of which NoSQL document database is better for the different numbers of records or as the number of records increases.

6. FUTURE SCOPE

The present and future of NoSQL technologies are bright, and full of opportunities and great challenges as it processes big data. The practical outcomes for this work will also help in the right selection of NoSQL document database.

In future we can compare these two document based databases for the different types of documents such as XML and CSV. We can also compare them with other NoSQL document databases as CouchDB and RavenDB and Cassandra.

7. REFERENCES

[1] Nathan Hurst, "Visual Guide to NoSQL Systems.", http://blog.nahurst.comlvisual-guide-to-NoSQL-systems/

[2] Mongodb,http://www.mongodb.org/display/DOCSlHome

[3] Couchdb, http://couchdb.apache.org/.

[4] Onur Savas, Yalin Sagduyu, Julia Deng, and Jason Li, "Tactical Big Data Analytics: Challenges, Use Cases and Solutions", Big Data Analytics Workshop in conjunction with ACM Sigmetrics 2013, June 21, 2013.

[5] Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C. and Byers, A.-H. (2011), "Big data: the next frontier for innovation, competition, and

productivity", McKinsey Global Institute, available at: www.mckinsey.com/Insights/MGI/Research/Technology_and_Innovation/Big_data_The_next_frontier_for_innovation

[6] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters", Commun. ACM, Pages:107-113, 2008.

[7] V Rabl, Tilmann; Sadoghi, Mohammad; Jacobsen, Hans-Arno; Villamor, Sergio Gomez-; Mulero -, Victor Muntes; Mankovskii, Serge (2012-08-27). "Solving Big Data Challenges for Enterprise Application Performance Management". VLDB. Retrieved 2013-07-25.

[8] Feuerlicht, G. and Pokorny, J. (2012), "Can relational DBMS scale-up to the cloud?", in Pooley, R.J.,Coady, J., Linger, H., Barry, C. and Lang, M. (Eds), Information Systems Development – Reflections, Challenges and New Directions,