Performance Analysis of Apriori Algorithm Using MongoDB

A Project

Submitted in partial fulfillment of the requirements for

the award of the Degree of

B.Sc. in Computer Science (Hons) Under CBCS

By

1. AVISHEK MITRA

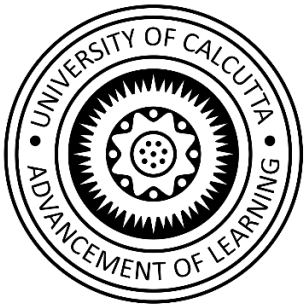
ROLL NO. – 183211-21-0032 **REGN. NO.** – 211-1111-0376-18

DEPARTMENT OF COMPUTER SCIENCE

MAHARAJA MANINDRA CHANDRA COLLEGE

UNIVERSITY OF CALCUTTA

Kolkata, West Bengal, India,2021



Certificate from the Supervisor

This is to certify that the work presented in the thesis entitled **"Performance Analysis of Apriori Algorithm using MongoDB"** in partial fulfillment of the requirement for the award of the degree of **B.Sc. in Computer Science Honours** of **University of Calcutta** is an authentic work carried out under my supervision and guidance.

To the best of my knowledge, the content of this thesis does not form a basis for the award of any previous Degree to anyone else.

It is understood that by this approval, the undersigned does not necessarily endorse any conclusion drawn or opinion expressed there, but approves the project for the purpose for which it is submitted.

Date: 07.08.2021

_____

PROF. SASWATI CHAKRABORTY

Dept. of Computer Science

Maharaja Manindra Chandra College

_____

Prof. Monali Poddar

Head of the Department

Dept. of Computer Science

Maharaja Manindra Chandra College

ACKNOWLEDGEMENT

We would like to express our gratitude to our mentor **Prof. Saswati Chakraborty** for her guidance and support in completing our Project and also our Head of the Department **Prof. Monali Poddar** as well as our college **'Maharaja Manindra Chandra College'**, who gave us the golden opportunity to do this wonderful project on the topic "Performance Analysis of Apriori Algorithm using MongoDB", which also helped us in doing a lot of research work and as a result we came to know about a lot of new things.

Every member of the group had also worked hard and helped in completing the project in due time.

The project helped us in increasing our knowledge and skills immensely

Thanks to all the group members too-

Name of Student – **Avishek Mitra**

C.U. Roll No – 183211-21-0032

CONTENTS

ABSTRACT

In present days there are lots of problems with Excel files and RDBMS(MySQL) to handle a large amount of data, problems like importing the data, Data accessing time, etc. To minimize this problem NoSQL Database MongoDB has come. Also, MongoDB is very popular nowadays so we choose this in our project. It also helps New MongoDB learners it is very easy to operate because it is schemaless, join free, document-oriented database and can handle a large amount of structured, unstructured data.

we have also implemented the Apriori algorithm using excel, RDBMS(MySQL) and we faced that execution get slow. So we use MongoDB as database. At the last, we compared their execution time and found that MongoDB is 79% faster than excel files and 73% faster than RDBMS(MySQL).

By this project we proof that in Bigdata environment MongoDB will be a great choice and these days there is a lot of demand on the market for NoSQL applications. So we think this application will be very helpful for people who are interested in MongoDB and used structured and unstructured data.

1.Introduction :

1.1 Domain Description

Nowadays we need to work with a huge amount of data in some companies the average amount of records are 3-4 billion per day. In this situation work with SQL server backend that collects and stores an extremely large amount of records. It can crash because it is very time-consuming to get table records.

To overcome this problem NoSQL database comes into the field. Dozens of NoSQL data stores are available MongoDB is one of them. It is an open-source document database. It usually does not have a schema. It has more few features like Document-oriented means instead of having data in relational type format, it stores the data in the document. This makes MongoDB very flexible and adaptable to real business world situations and requirements. Ad hoc queries MongoDB supports search by field, range queries, and regular expression searches. Indexing- index can be created to improve the performance of searches within MongoDB. Replication- MongoDB can provide high ability with replica sets. Load balancing- MongoDB can run over multiple servers, balancing the load and/or duplicating data to keep the system up and running in case of hardware failure.

Our software is project helpful too for people who are used to structure database queries and learning MongoDB. The project Performance analysis of the Apriori Algorithm using MongoDB provides lots of features. In the current system, we have few on our hands but in the future, we will enhance it with more features. The application has a good appearance and is easy to surf. It has very simple source code which saves time. This project contains advanced modules like MongoDB which makes the system very powerful.

1.1.1 What is Big Data?

Big Data is a collection of data that is huge in volume, yet growing exponentially with time. It's data with so large size and complexity that none of the traditional data management tools can store it or process it efficiently. Big data is also various types of data like video, photos, audio, webpages, and multimedia content but with a huge amount of size.

Big Data means new opportunities for the organization to create business value – and extract it. The MongoDB NoSQL database can underpin many Big Data Systems, not only as a real-time, operational data store but in offline capacities as well. With MongoDB, organizations are serving more data, more users, more insight with greater ease and creating more value worldwide.

.

1.1.2 How NoSQL databases are related to Big Data :

1. HBase is a No SQL database used for Hadoop, a popular NoSQL database is also used by Facebook for its messaging infrastructure.
2. HBase is also used by Twitter for creating data, storing and monitoring data.
3. MongoDB is another NoSQL Database used by CERN, it is a European Nuclear Research Organization for collecting data.
4. LinkedIn also used NoSQL Database for various data processing and data monitoring related work.

1.1.3 Over view of NoSQL databse:

NoSQL Database is a non-relational Data Management System. That does not require a fixed schema. It avoids join and is easy to scale. The major purpose of using a NoSQL database is for distributed data stores with humongous data storage needs. NoSQL is used for Big data and real-time web apps. For example, companies like Twitter, Facebook, and Google collect terabytes of user data every single day.

NoSQL database stands for "Not Only SQL" or "Not SQL." Though a better term would be "NOREL", NoSQL caught on. Carl Strozzi introduced the NoSQL concept in 1998.

In the case of Big data, NoSQL is better than RDBMS(Relational Database Management System), So That's the reason NoSQL now leads the way for the popular internet companies such as LinkedIn, Google, Amazon, and Facebook - to overcome the drawbacks of the 40 -year- old RDBMS.

Some popular NoSQL databases examples are MongoDB, CouchDB, CouchBase, Cassandra, HBase, Redis, Riak, Neo4 ,etc.

1.1.4 What is MongoDB?

MongoDB, the most popular NoSQL database, is an open-source document-oriented database. MongoDB was created by Eliot and Dwight in 2007 when they faced scalability issues while working with a relational database. MongoDB isn't based on the table-like relational database structure but provides an altogether different mechanism for the storage and retrieval of data. This format of storage is called BSON (similar to JSON format). Relational Database Management System (RDBMS) is not the correct choice when it comes to handling big data by the virtue of their design since they are not horizontally scalable**.** MongoDB is such a NoSQL database that scales by adding more and more servers and increases productivity with its flexible document model.

1.1.5 Advantages of MongoDB over RDBMS :

- **Schema less** − MongoDB is a document database in which one collection holds different documents. Number of fields, content and size of the document can differ from one document to another.
- Structure of a single object is clear.
- No complex joins.
- Deep query-ability. MongoDB supports dynamic queries on documents using a document-based query language that's nearly as powerful as SQL.
- Tuning.
- Ease of scale-out − MongoDB is easy to scale.
- Conversion/mapping of application objects to database objects not needed.

1.1.6 What is Apriori algorithm ?

The Apriori algorithm uses frequent item sets to generate association rules, and it is designed to work on the databases that contain transactions. With the help of these association rule, it determines how strongly or how weakly two objects are connected. This algorithm uses a **breadth-first search** and **Hash Tree** to calculate the itemset associations efficiently. It is the iterative process for finding the frequent item sets from the large dataset.

1.1.7 Benefits of Apriori Algorithm :

- This is the most simple and easy-to-understand algorithm among association rule learning algorithms.
- The resulting rules are intuitive and easy to communicate to an end-user. Avoid rescanning the database.

1.2 Motivation :

Main motivation of our project "Performance analysis of Apriori Algorithm using MongoDB" is reduce the time, because in case of Big data NoSQL database- MongoDB is faster and better than RDBMS. Normally an Apriori algorithm is implementing by using Excel/CSV file as database, but In our project we are using MongoDB instead of Excel or MySQL to get more faster results as compare to others.

1.3 Scope of the project :

- Through our project we done maximum software related work like searching of useful data within minimum time. Though it is reducing the time so it can increase sales and customer satisfaction in future.
- Our project is efficient for market basket analysis and helps to enhance market sale by assisting customers during the purchase of the item. Another popular application is Google Autocomplete in which the search engine suggests the other associated words according to your specified word, and also it is used in the Amazon recommendation system.
- In this project we use MongoDB in Bigdata environment, and it will save our time in future works.

2.Background

In the last few years, the researchers have shown the impact of MongoDB on the real life events and needs in their research activities. Moreover, a notable amount of works have been proposed in the related fields to monitor the effect of the same.
We have reviewed some of the articles and works as per our Project interest:

1. We reviewed articles for better understandings of works about Big data, Big data is a field that treats ways to analyze, systematically extract information from, or otherwise deal with data sets that are too large or complex to be dealt with by traditional data-processing application software. [1]
2. We also followed some articles and thesis about comparison between MongoDB and MySQL to understand the real life effects. [2]
3. We gone through with few real life works and articles about Mongo db for the better clarification about the project.[3]MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas. [4]
4. We also reviewed some works about the practical difference between MongoDB and relation database for acquiring depth of the project and to understand the immense effect and power of the systems. [5]
5. We followed some articles about Apriori algorithm [6] to gain our knowledge and to understand the improved efficiency of Apriori algorithm using transaction reduction [7]
6. To improve the efficiency of level-wise generation of frequent itemsets, an important property is used called Apriori property which helps by reducing the search space. [8]

After review this things , we understand that Performance analysis of Apriori Algorithm using Excel file, RDBMS(MySQL) and NoSQL(MongoDB) will not be not be occurred.

3.Methodology

3.1 PROBLEM FORMULATION :

Big Data is a collection of data that is huge in volume, yet growing exponentially with time. It's data with so large size and complexity that none of the traditional data management tools can store it or process it efficiently.

we all are still using Excel files to transform and/or analyze some important data. There is nothing wrong with using Excel. Excel files are a great tool to collect and transform small amounts of data. But when users want to work with Big Data it showing some limitations. In Excel File there is no error control, reusability is little, scalability of Excel file is too much problematic and Excel file takes too much time when it works with Big Data.

In this situation when we work with SQL for Big Data SQL server a backend that collects and stores an extremely large amount of records. It can be crushed and burned fairly quickly, obviously because it is impossible to query a table that's having many records inserted and SQL takes also a huge amount of time when it works with Big Data.

3.2 Algorithm Description :

Apriori algorithm, a classic algorithm, is useful in mining frequent itemsets and relevant association rules. Usually, you operate this algorithm on a database containing a large number of transactions.

One such example is the items customers buy at a supermarket.

It helps the customers buy their items with ease, and enhances the sales performance of the departmental store.

This algorithm has utility in the field of healthcare as it can help in detecting adverse drug reactions (ADR) by producing association rules to indicate the combination of medications and patient characteristics that could lead to ADRs.

4.Implementation

To implement our project we divided our implementation into three parts and we used some software-Python, MySQL, MongoDB. we use the same data in every case.

- Implementing Apriori algorithm using Excel file.
- Implementing Apriori algorithm using MySQL.
- Implementing Apriori algorithm using MongoDB.

  - We use the same quantity of data in every case to get the time difference between them.

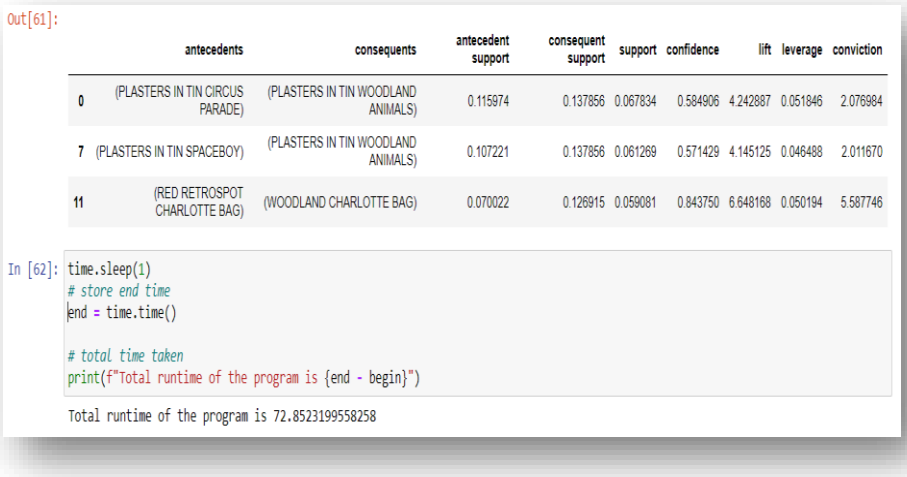
Implementing Apriori algorithm using Excel file

1. Organized all the data in an Excel file.
2. At first add some libraries like Mlxtend, Pandas for the day-to-day task.
3. Then add the Excel file to the python by using [pd.read_excel()]. We have to give the path of the excel file. (Figure 1)

```
In [6]: import pandas as pd
        from mlxtend.frequent_patterns import apriori
        from mlxtend.frequent_patterns import association_rules

In [7]: import time

        # store starting time
        begin = time.time()

In [8]: df = pd.read_excel('C:\\Users\\91798\\Downloads\data.xlsx')

In [9]:
        df.head()
```

Figure 1

4. Write down the source code of the Apriori algorithm.
5. At the end calculate the time taken by this program. (Figure 2)



Out[61]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(PLASTERS IN TIN CIRCUS PARADE)	(PLASTERS IN TIN WOODLAND ANIMALS)	0.115974	0.137856	0.067834	0.584906	4.242887	0.051846	2.076984
7	(PLASTERS IN TIN SPACEBOY)	(PLASTERS IN TIN WOODLAND ANIMALS)	0.107221	0.137856	0.061269	0.571429	4.145125	0.046488	2.011670
11	(RED RETROSPOT CHARLOTTE BAG)	(WOODLAND CHARLOTTE BAG)	0.070022	0.126915	0.059081	0.843750	6.648168	0.050194	5.587746

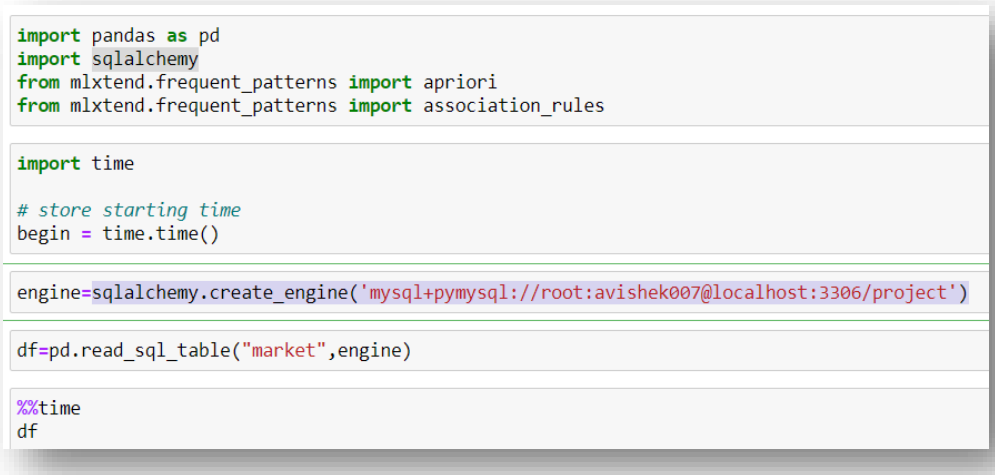
In [62]:
```
time.sleep(1)
# store end time
end = time.time()

# total time taken
print(f"Total runtime of the program is {end - begin}")
```

```
Total runtime of the program is 72.8523199558258
```

Figure 2

Implementing Apriori algorithm using MySQL

1. At first, we have to make a local host, a database, and a table for storing data.
2. Then Import the data into MySQL's table.
3. Add some libraries like Mlxtend, sqlalchemy, etc at the beginning.
4. Now connect MySQL with Python by using[sqlalchemy.create_engine()] and added localhost, root, database name, and table name. (Figure 3)



```
import pandas as pd
import sqlalchemy
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
```

```
import time

# store starting time
begin = time.time()
```

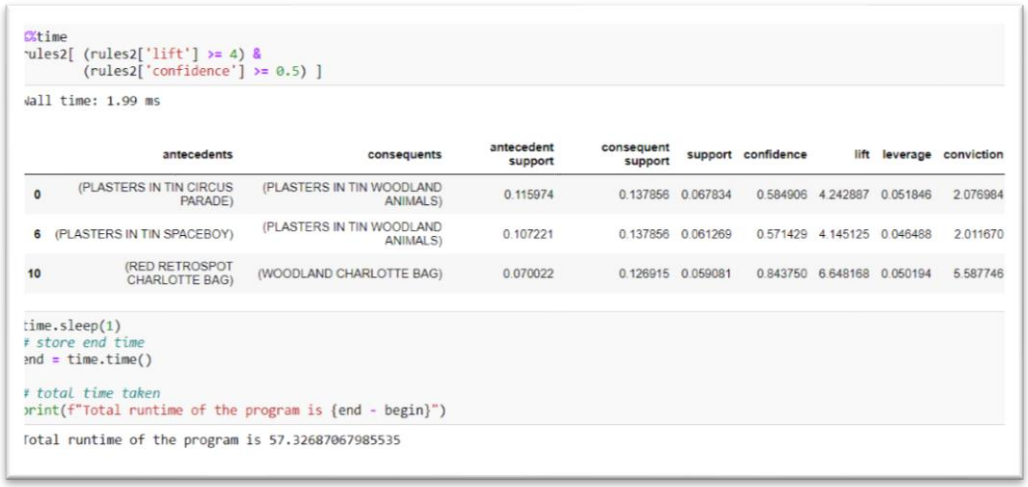
```
engine=sqlalchemy.create_engine('mysql+pymysql://root:avishek007@localhost:3306/project')
```

```
df=pd.read_sql_table("market",engine)
```

```
%%time
df
```

Figure 3

5. Write down the source code of the Apriori algorithm.
6. At the end calculate the time taken by this program. (Figure 4)



```
%%time
rules2[ (rules2['lift'] >= 4) &
        (rules2['confidence'] >= 0.5) ]
```

```
Wall time: 1.99 ms
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(PLASTERS IN TIN CIRCUS PARADE)	(PLASTERS IN TIN WOODLAND ANIMALS)	0.115974	0.137856	0.067834	0.584906	4.242887	0.051846	2.076984
6	(PLASTERS IN TIN SPACEBOY)	(PLASTERS IN TIN WOODLAND ANIMALS)	0.107221	0.137856	0.061269	0.571429	4.145125	0.046488	2.011670
10	(RED RETROSPOT CHARLOTTE BAG)	(WOODLAND CHARLOTTE BAG)	0.070022	0.126915	0.059081	0.843750	6.648168	0.050194	5.587746

```
time.sleep(1)
# store end time
end = time.time()

# total time taken
print(f"Total runtime of the program is {end - begin}")
```

```
Total runtime of the program is 57.32687067985535
```

Figure 4

Implementing Apriori algorithm using MongoDB

1. At first, we have to make a localhost, database, and a collection for storing data.

2.Then Import the data into MongoDB's collection. (
Figure 5)

- Open the Import Wizard. Then, choose CSV as the import format.
- Click on the folder icon and locate the CSV file to be imported.
- Then Run this task and your data is successfully imported.

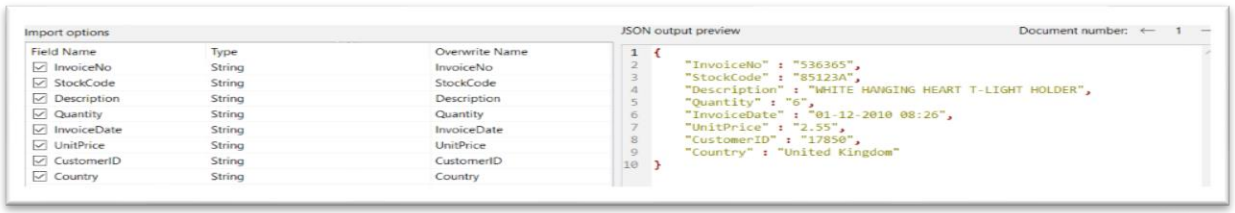


Figure 5

3. Add some libraries like Mlxtend, Pandas for day-to-day tasks.

4. Next connect MongoDB with python. (Figure 6)

- We have to connect with MongoDB to Python by adding the path("mongodb://localhost:27017/avishek").
- Then add the database name, collection name where the data is store in MongoDB.
- Store the data into an object.

```
In [13]: import pandas as pd
         from mlxtend.frequent_patterns import apriori
         from mlxtend.frequent_patterns import association_rules

In [14]: import time

         # store starting time
         begin = time.time()

In [15]: from pymongo import MongoClient

         connection_string="mongodb://localhost:27017/avishek"

In [16]: client = MongoClient(connection_string)

In [17]: db = client.get_database("avishek")

In [18]: print(db.list_collection_names())
         ['avidata', 'newdata']

In [19]: collection =db.get_collection("avidata")

In [20]: document = collection.find_one()

In [21]:
         cursor = collection.find()

In [22]: df=pd.DataFrame(cursor)

In [23]: %%time
         df.head()
```

Figure 6

5. Write down the source code of the Apriori algorithm.

6. At the end calculate the time taken by this program. (Figure 7

Out[41]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
1	(PLASTERS IN TIN CIRCUS PARADE)	(PLASTERS IN TIN WOODLAND ANIMALS)	0.115974	0.137856	0.067834	0.584906	4.242887	0.051846	2.076984
7	(PLASTERS IN TIN SPACEBOY)	(PLASTERS IN TIN WOODLAND ANIMALS)	0.107221	0.137856	0.061269	0.571429	4.145125	0.046488	2.011670
11	(RED RETROSPOT CHARLOTTE BAG)	(WOODLAND CHARLOTTE BAG)	0.070022	0.126915	0.059081	0.843750	6.648168	0.050194	5.587746

```
In [42]: time.sleep(1)
         # store end time
         end = time.time()

         # total time taken
         print(f"Total runtime of the program is {end - begin}")
```

```
Total runtime of the program is 15.242836952209473
```

Figure 7

5.Result and Discussion

After implementation, we calculated the time taken for every execution.

- The Apriori algorithm program, which connects with the **Excel file** takes time for total execution is 72.85 sec.

Performance graph of Apriori algorithm Using Excel file

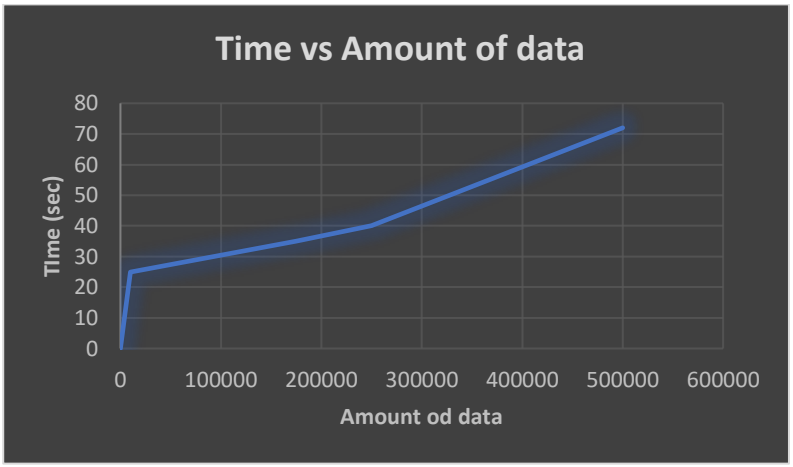


Figure 8

- The Apriori algorithm program, which connects with the **MySQL** takes time for total execution is 57.32 sec.

Performance graph of Apriori algorithm Using MySQL

Figure 9

- The Apriori algorithm program, which connects with **MongoDB** takes time for total execution is 15.24 sec.

Performance graph of Apriori algorithm Using MongoDB

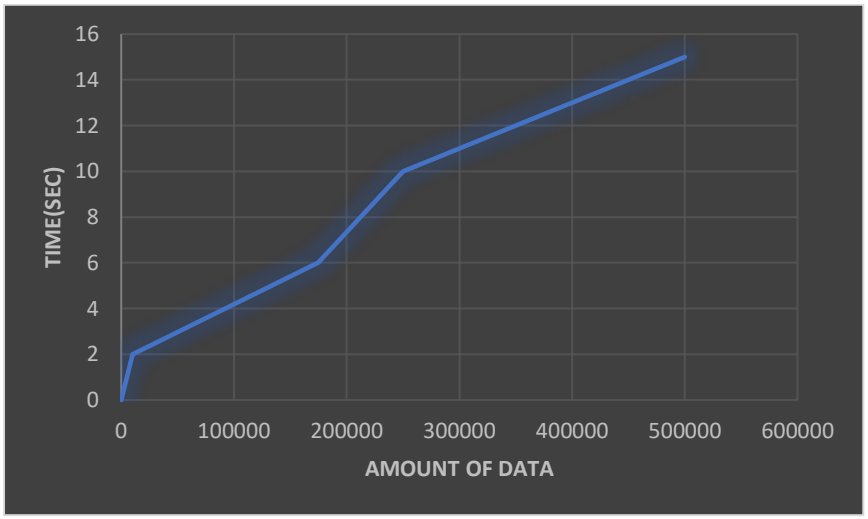


Figure 10

Performance Analysis

- Now we calculate the percentage of how much time MongoDB is faster than excel file.

X = percentage

$$X = \left\langle \frac{\text{Execution time of Excel program - Execution time of Mongodb program}}{\text{Execution time of Excel program}} \right\rangle * 100\ \%$$

$$X = \left\langle \frac{72.85 - 15.24}{72.85} \right\rangle * 100\ \%$$

X= 79 %

After calculation, we found that MongoDB is 79% faster than excel files in the Apriori algorithm

- Now we calculate the percentage of how much time MongoDB is faster than MySQL.

X = percentage

$$X = \left(\frac{\text{Execution time of MySql program - Execution time of MongoDB Program}}{\text{Execution time of MySql program}} \right) * 100$$

$$x = \left(\frac{57.32 - 15.24}{57.32} \right) * 100$$

X= 73%

After calculation, we found that MongoDB is 73% faster than MySQL in the Apriori algorithm.

Discussion

After getting the result we found that MongoDB is more efficient when we work Bigdata. Below the image(Figure 11), we can see that when the number of data is increasing MySQL and Excel files getting slow as respect to MongoDB.

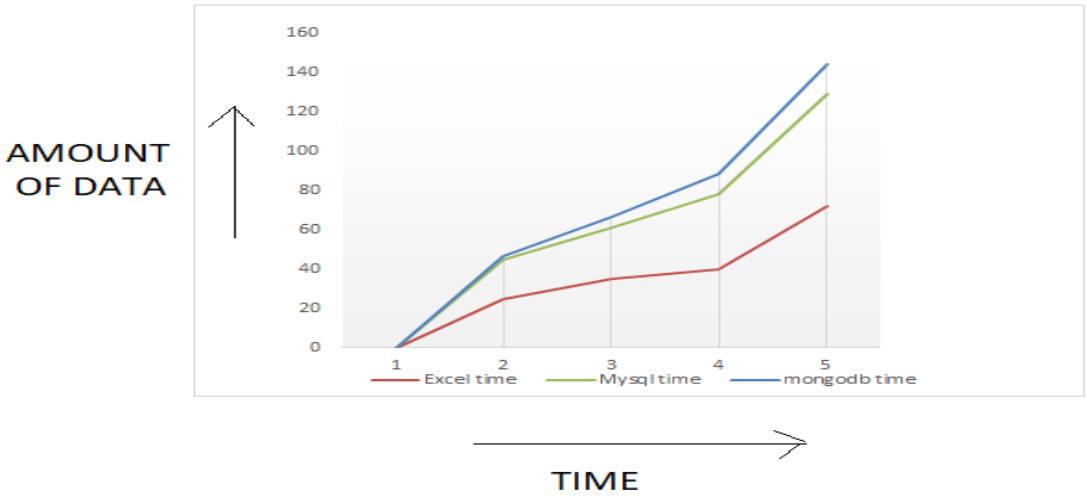


Figure 11

6.Conclusion

In our project “Performance analysis of Apriori algorithm using MongoDB” is proof that MongoDB is faster instead of using MySQL and Excel files in Big data environment.

When we work with huge amount of data, importing data into RDBMS(MySQL) is very time-consuming. But in MongoDB, we can import a huge amount of data in a very short time. In the end, we can say that MongoDB makes program execution time lesser and it will be more useful and saving our time.

7.References

1. S. Sagiroglu and D. Sinanc, "Big data: A review.," international conference on collaboration technologies and systems (CTS), pp. 42-47, may 2013.
2. C. Győrödi, R. Győrödi, G. Pecherle and A. Olah, “A comparative study: MongoDB vs. MySQL.,” 13th International Conference on Engineering of Modern Electric Systems (EMES), pp. 1-6, 2015.
3. Z. Wei-Ping, L. Ming-Xin and C. Huan, "Using MongoDB to implement textbook management system instead of MySQL," IEEE 3rd International Conference on Communication Software and Networks, pp. 303-305, May 2011.
4. K. Banker, D. Garrett, P. Bakkum and S. Verch, MongoDB in Action: Covers MongoDB version 3.0., Simon and Schuster, 2016.
5. G. Zhao, W. Huang, S. Liang and Y. Tang, "Modeling MongoDB with relational model.," Fourth International Conference on Emerging Intelligent Data and Web Technologies, pp. 115-121, September 2013.
6. M. Al-Maolegi and B. Arkok, An improved Apriori algorithm for association rules., arXiv preprint arXiv:1403.3948., 2014.
7. J. Singh, H. Ram and D. and Sodhi, "Improving efficiency of apriori algorithm using transaction reduction.," International Journal of Scientific and Research Publications,, no. 3(1), pp. 1-4, 2013.
8. J. Yabing, "Research of an improved apriori algorithm in data mining association rules.," International Journal of Computer and Communication Engineering, no. 2(1), p. 25, 2013.