HYBRID DATABASE SYSTEM FOR BIG DATA STORAGE AND MANAGEMENT

Blessing E. James and P.O.Asagba

Department of Computer Science, University of Port Harcourt, Choba, Rivers State, Nigeria

ABSTRACT

Relational database systems have been the standard storage system over the last forty years. Recently, advancements in technologies have led to an exponential increase in data volume, velocity and variety beyond what relational databases can handle. Developers are turning to NoSQL which is a non- relational database for data storage and management. Some core features of database system such as ACID have been compromised in NOSQL databases. This work proposed a hybrid database system for the storage and management of extremely voluminous data of diverse components known as big data, such that the two models are integrated in one system to eliminate the limitations of the individual systems. The system is implemented in MongoDB which is a NoSQL database and SQL. The results obtained, revealed that having these two databases in one system can enhance storage and management of big data bridging the gap between relational and NoSQL storage approach.

KEYWORDS

ACID, BASE, Big data, NoSQL, SQL, MongoDB

1. INTRODUCTION

Rapid industrialization, greater affordability of devices, sudden escalation in the usage of portable devices and recent advancement in technological knowledge such as the connection of every object to the internet known as Internet of Things (IoT), mesh or cloud computing, big user etc. have led to an inconceivably massive realm of data known as big data[18][19]. Generally, big data is regarded as enormous volume (terabyte and pentabyte) of data consisting of various data types (structured, semi-structured and unstructured) and of real time availability (velocity) so that it becomes impossible for such data to be stored or managed using means like traditional relational database systems. In simple words, big data is data whose volume and nature (semi-structured and unstructured data type) is greater than what conventional database systems could handle. This enormous increase in data quantity has opened up great opportunities for significant scientific achievements, improved business strategies, health care methods etc**. [**14]

Relational database systems have served as the actual storage systems for several years. However, within the last four years, there have been great resolutions in the computing world that have lessened the prevalent of relational databases which has led to an up thrust in the consideration of new storage model called NoSQL **[**17**].** This is because relational databases were never designed to store or manage such volume of data, with high velocity and variety, unstructured data or rapid growth [15]. Enterprises are therefore turning to a new emerging storage approach called NoSQL for solutions to inherent challenges in big data. Most NoSL scale horizontally with increase in data volume and are also sufficiently flexile in order to hold partially structured and dispersed data collection [11]. With NoSQL, data in the form of voice, video, email, and documents can be properly stored and managed. However, as appealing as this approach is, it is not without limitations. The innovators of NoSQL will-fully or perhaps unintentionally left out some

DOI: 10.5121/ijcsea.2017.7402

desirable database ingredient such as transactions, and security in order to achieve what relational databases could not offer .

In this work, we developed a hybrid database system using MongoDB and MySQL databases which are popular variants of NoSQL database and relational database systems respectively. Prior to data storage, data is categorized into structured data and unstructured data depending on the nature of data. Unstructured data are stored and managed in the MongoDB database while storage and management of strictly structured data is carried out using MySQL database. Our hybrid system is such that, the databases making up the system can function separately for instance, our system could be used as a separate and complete MongoDB database. Instead of giving up the functions of relational database systems for NoSQL database we have developed an approach that offers the benefits of the both systems in a single database system.

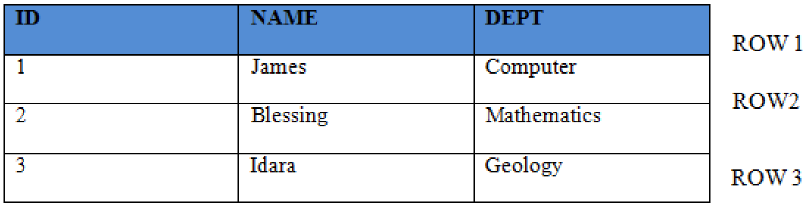
2. Relational Database Storage Approach

The paradigm governing relational databases are built on the principles of relations in mathematics. Many of the prevalent and free databases are in this class. Being able to store data in tables of rows and columns while retaining and imposing the relationship between the sored data is one of the basic traits of relational databases [1]. The basic rules of relational databases [1] are:

- Data and information must be stored in tables of rows and columns.
- To access the content of a column, the name of the table, column and the primary key must be specified.
- Cases of absent and inappropriate entry must be handled systematically different from expected entries and not dependent on the type of data being entered.
- The Database Management System should support an active online catalogue.
- There must be at least one language supported by the Database Management System which could be used either separately or within programs.
- The Database Management System must be able to update views
- Basic operations such as insertion, updating and deletion must be supported by the Database Management System.
- Modifications performed on the logical structure such as adding or removing columns from tables must not affect the user views.
- Changes at the physical level such as storage must not affect the entire application.
- Restrictions pertaining to Integrity should be isolated from the application.
- In distributed environment, the impact of database distribution must be perceived by user.

In relational model, data is represented in tables or relations. A table as shown in table 1 is primarily a collection of related data entries and it is made up of several columns and rows referred to as fields and records respectively. Almost all databases built on relational model guarantee Atomicity Consistency Isolation and Durability transactions.

Table 1: Tabular data representation in relational database



ID	NAME	DEPT
1	James	Computer
2	Blessing	Mathematics
3	Idara	Geology

ROW 1
ROW2
ROW 3

3. NOSQL DATABASE STORAGE APPROACH

NoSQL databases are Object Oriented databases designed to address processing issues created by expanding data volumes and diversity, particularly in big data applications. NoSQL databases are considered to be very necessary in situation where the volume of data is far beyond what could be handled by relational database and also the information constituent is such that must not be stored in a relational database. NoSQL databases are built on distributed model to guarantee Basically Available, Soft-state, Eventual consistency (BASE) properties [21]. Figure 1 shows a model of document base NoSQL database. In this model instead of storing data in rows and columns in a table, data which may fill several columns in a table could be stored in documents which are grouped into collections. There are four basic categories of NoSQL database [7].

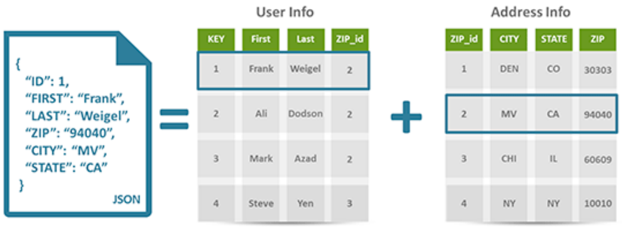
- Key-Value Store: Data is stored using two connected but distinct items- a distinctive identifier called key and a corresponding value which could be a data or a pointer to the location of the data. It is very suitable for key based systems. E.g. Dynamo, Riak
- Column family store: Data is organized in rows and columns as in relational database systems. E.g. Cassandra
- Graph family store: for processes which could be represented in the form of relationship with interconnected elements such as social network. E.g. Neo4j
- Document Store: Data is stored in documents .Appropriate for storage of documents in diverse format. E.g. MongoDB

3.1 STRENGTH OF NOSQL DATABASE AND LIMITATIONS OF RELATIONAL DATABASE

NoSQL databases were developed to eliminate the limitations or drawbacks encountered in the use of relational databases especially in big data storage enronments. As such, most of the drawbacks in the relational storage system form the strength or advantages of NoSQL database systems. The weaknesses of relational databases and the strength of NoSQL databases depend mostly on the features discussed in the following sections.

3.1.1 SCALABILITY

In relational storage systems, expansion is achieved by replacing the existing storage or server with a bigger (more expensive) server which implies increasing the horse power of the existing hardware . This is known as the vertical scaling or scaling up . It is obvious that as the volumn of data increases, it may get to a stage that the biggest affordable server may not be able to meet the storage requirement as shown in figure 2 , this may in turn reduce the system's performance. And also the system is plagued with a single point of failure. NoSQL databases are built on distributed architecture such that partitioning (sharding) of a database across several servers is possible. As such expansion is achieved through the addition of inexpensive servers connected to the database cluster shown in figure 3. This is known as horizontal scaling or scaling out. Horizontal scaling increases system's performance at minimal cost by promoting rapid data expansion and eliminating the single point of failure existing in relational databases.



```
{
  "ID": 1,
  "FIRST": "Frank",
  "LAST": "Weigel",
  "ZIP": "94040",
  "CITY": "MV",
  "STATE": "CA"
}
```

User Info

KEY	First	Last	ZIP_id
1	Frank	Weigel	2
2	Ali	Dodson	2
3	Mark	Azad	2
4	Steve	Yen	3

Address Info

ZIP_id	CITY	STATE	ZIP
1	DEN	CO	30303
2	MV	CA	94040
3	CHI	IL	60609
4	NY	NY	10010

Figure 1: Document base data representation (Source: [23])

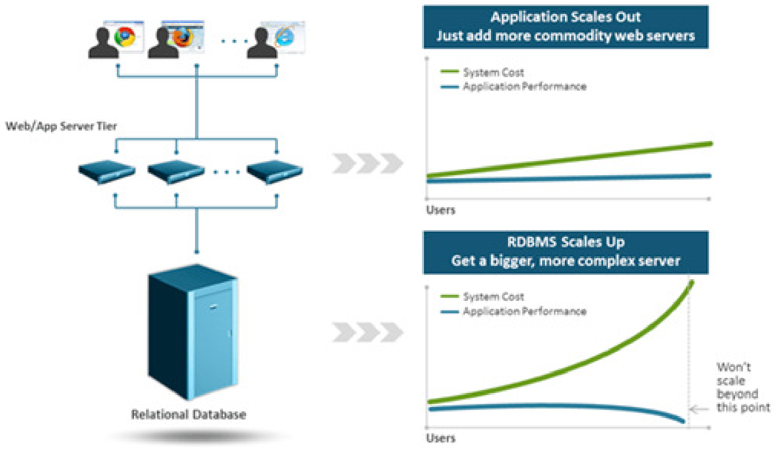


Figure 2.Vertical scaling in relational databases. (Source [23])

3.1.2 Flexibility

Relational databases are schema-agnostic; data cannot be stored without defining the schema of such data. As such in big data environment where there is need for storage of unstructured data, it is impossible to know the schema or structure of data beforehand. NoSQL on the other hand have dynamic schema such that the schema must not be pre-defined. As such, NoSQL database could be for storage of both structured and unstructured data.

Weaknesses Of Nosql Database

Solving some of the problems in relational databases introduced certain weaknesses in NoSQL database. Some of the weaknesses of NoSQL database are:

- Complex transactions: MongoDB does not support multi-document transactions. With the availability of the NoSQL databases, support for ACID transactions across documents was typically given up. Exclusion of ACID transaction is a trade-off used by NoSQL to provide solution to issues pertaining to scalability.
- Stability: Some NoSQL databases are still in their pre-production phases and are therefore not stable or matured enough for some sensitive task.
- Global Support: Enterprises demand global support and services from database vendors when a core component of the system fails. NoSQL lacks such services to enterprise customers.

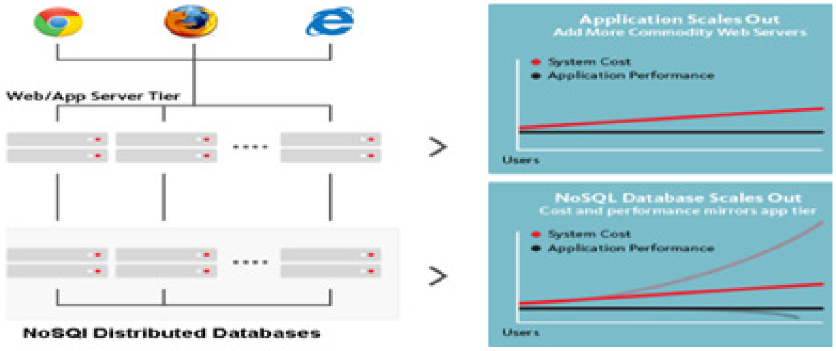

Web/App Server Tier
NoSQl Distributed Databases
Application Scales Out
Add More Commodity Web Servers
System Cost
Application Performance
Users
NoSQL Database Scales Out
Cost and performance mirrors app tier
System Cost
Application Performance
Users

Figure 3. Horizontal scaling in NoSQL database. (Source [24])

4. Big Data

Big data refers primarily to group of data which have become extremely voluminous (petabytes and terabyte) consisting of various data types (structured, semi-structured and unstructured) and of real time availability (velocity) such that it is not efficient to be stored or processed with traditional tools or means such as conventional database systems. These traditional tools have been used over the years to hold, process and analyse voluminous set of information in companies, industries etc. Big data does not only refer to collection of voluminous data, but extremely large volume of structured and unstructured data subject to very high rate of change, derived from divers avenues which may include email, social media, phone calls etc. Big data is used to describe a collection of sophisticated data whose volume increases continuously and rapidly at such rate that the use of conventional management tools for storage and analysis becomes inefficient and inappropriate. The complexity in this enormous volume of dispersed data is attributable to the fact that data collected from different sources may also have different format and it may be necessary to integrate them into a unit for analysis.

4.1 Properties Of Big Data

Although the word big indicates the size of a thing, in big data, big is not limited to data volume but also incorporates other attributes such as Velocity, and Variety. These three attributes describe the primary properties of big data known as the 3 V's of big data [22];

- Volume: This is sometimes taken to be the ultimate attribute of big data. It depicts very large and ever growing amount of data ranging from terabyte (1012 byte) to yotta-byte which is trillions of gigabyte.
- Velocity: Velocity refers to real time availability of data for processing. Big data is also characterized by instantaneous arrival of enormous data for processing. It entails the rate at which data is circulated within the system e.g. the velocity upon which data is derived out of internal and external operations and sources such as interactions with machines, humans, social media etc
- Variety: This represents the diverse format of data in a data set. Big data is made up of data derived from various sources such as emails, machines, social networks, business transactions, mobile devices etc. Data from different sources assume different forms such as spread sheets, photos, videos etc. Variety as a property of big data describes divers forms of data derived from diverse sources.

- Veracity: Refers to the truthfulness of the data. It deals with the relevance of the data (processed or analysed) to the task at hand. It reveals the need to avoid accumulation of dirty data.
- Volatility: Deals with the reasonable life span of stored data in the world of real time data processing. It investigates the validity of stored data to the current analysis.
- Validity: Decisions are as valid as the data used in the analyses

4.2 BIG DATA CHALLENGES

Big data challenges could also be referred to as the steps involved in the processing of this extremely large volume of data of divers types and high velocity for use. To leverage the numerous benefits of big data, the data must go through the following processes [5], in other words, the following challenges must be overcome;

- Ingestion: The procedure through which data is obtained and imported for use or storage. Data can be ingested once it is supplied by the source or grouped into batches and ingested within specified interval. The process usually starts with ranking of data sources, authenticating each file before channelling data to accurate destination.
- Storage: Storage is very complex; it includes search and retrieval of data and may also include complex security and privacy issues. Storage framework for big data should be able to hold large volumes of structured, semi-structured and unstructured data efficiently.
- Analytic: Big data is almost useless without efficient analytic tools and procedures through which useful information is extracted from what seems to be junk of data. Analytics in the field of big data also include computations on big data. It covers frameworks and tools such as Map Reduce and Hadoop that are used to derive meaning from big data. The result of data analytics can be used to improved services rendered to customers, marketing strategies, and general decision making.
- Visualization: This covers data presentation for easy identification of patterns or grasping of new concepts. Big data visualization tools and techniques allow data representation in the form of graphs, charts, maps and even videos making it easy for information communication.

5. RELATED WORKS

[13] [2] presented a report on the classification, properties and comparison of NoSQL databases. They explored the strength and weaknesses of the different types of NoSQL databases. [3] Investigated the elasticity of NoSQL databases based on the scalability and speed of read and update operations. It was shown that the speed at which read operation is performed in Hbase is high whereas insert operation is fast in Cassandra and Riak is slow at both read and write operations. [17] Carried out a detailed comparison between MongoDB and Microsoft SQL databases. Microsoft SQL is a relational database system as such the storage capacity of the system can only be increased by introducing a server with bigger capacity and this usually incurs extra cost. NoSQL database on the other hand is non-relational system whose capacity could easily scale horizontally to accommodate more data. The system was implemented in Java programming language using Eclipse Integrated Development Environment. It was observed that although MongDB and Microsoft SQL perform write operations faster than read operations, read and write operations in MongoDB is almost ten times faster than read and write in Microsoft SQL database. Although MongoDB has a higher read/write ability, there exists situations were speed is not the ultimate or the only requirement for databases. MongoDB is not appropriate for heavy transactional tasks. An evaluation of the performances of MongoDB, PostgreSQL and Cassandra by [4] revealed that Cassandra is more appropriate for large distributed sensor systems. [6]

Opined that more engineering task is demanded from programmers who depend on relational databases for data storage and management.

6. PROPOSED SYSTEM

The aim of the proposed system is to design a Hybrid Database System for the storage and management of big data. Our hybrid system is made up of MySQL database and MongoDB which are the most popular relational and NoSQL (non-relational) database servers. Data is grouped into structured and unstructured data category, structured data is channelled into the MongoDB database, while the choice of database for the unstructured data depends on the mode in which the application runs in; this could be MongoDB for hybrid mode and MySQL for SQL mode. The databases integrated in the proposed system can also function in isolation.

[8] Presented an overview of the existing NoSQL databases using data model, query model, replication and consistency model.

6.1 DESIGN OF PROPOSED SYSTEM

System design shows the components that make up a system. The proposed system consists of the following basic components; MySQL database, MongoDB database. These components are further discussed in detail and the architectural design of the proposed system given in figure 3 shows the connections amongst these components.

- SQL Component: contains the storage engine which handles data storage in MySQL database. The storage engine is made up of a transaction log file and data file groups which could be hierarchically broken down into data files table, indexes, extent and page which is the smallest unit of storage in relational databases. The transaction log file component of the storage engine is used to achieve and maintain data integrity and recovery in the database. It records the start and end of each operation and also every modification performed on data in the database.

- MongoDB Component: MongoDB uses replication to ensure redundancy and consistency. Influx of data from different destinations and in different format are broken down and equally dispersed to a collection of non-static extensible terminals called shard. Data describing other data within the cluster are saved in configuration servers. Every of these servers contain replica of all metadata for the purpose of redundancy. When client request is made, it forms one of the routing processes which are used to check the configuration servers to know the position of the request.

6.2 ALGORITHM

Internally, our hybrid system uses both the B-tree algorithm and the proportional fill algorithm.
The relational database (SQL) runs majorly on a proportional filled algorithm. The proportional fill algorithm is used by the SQL storage engine to write data to the database files depending on the size of free space in each data file rather than writing in each file until it is full then moving to the second one sequentially. As such, the SQL server storage engine will write more frequently to the files with more free space. MongoDB on the other hand is built on a B-tree algorithm. However, a step-by-step procedure for data processing in our hybrid system is given as;

```
Step 1: Load data
Step 2: Define class of data
Step 3: Initialize DB hybrid Interface
Step 4: Test Data: if
        Data is structured then
                Store in SQL database
        Data is unstructured then
                Store in MongoDB database
```

Step 5: Update database hybrid interface
Step 6: View, Delete, Update, Exit

7. Experimental Setup And Results

The proposed system is implemented in C# in visual studio Integrated Development Environment. MongoDB and MySQL were used for data storage. Some of the classes used in the program are; Person, Symbol, PatientInfo, FileInfo, Panel, Person.

For our application to function properly, the MongoDB server would first be started from command prompt by navigating to the installation directory and executing the command mongod.exe as shown in figure 4. By default, MongoDB server will start at port 27017. To start the client shell, the command mongo.exe is executed on a separate command prompt window. Figure 5 shows the client shell connecting to Test which is the default system database.

To execute our hybrid application, the input data which is made up of structured and unstructured data is saved in the system's local disk. We invoke our application by double clicking on the application icon. An overview of our application shown in figure.6 shows the modes that our proposed system can function in.

When big data is loaded in our hybrid system, the database used for storage is determined by the mode in which the application runs in. Unstructured data is channelled to the MongoDB database (except when the application is in MongoDB mode in which case MongoDB stores both structured and unstructured data) while MySQL database is used to store and manage structured data.

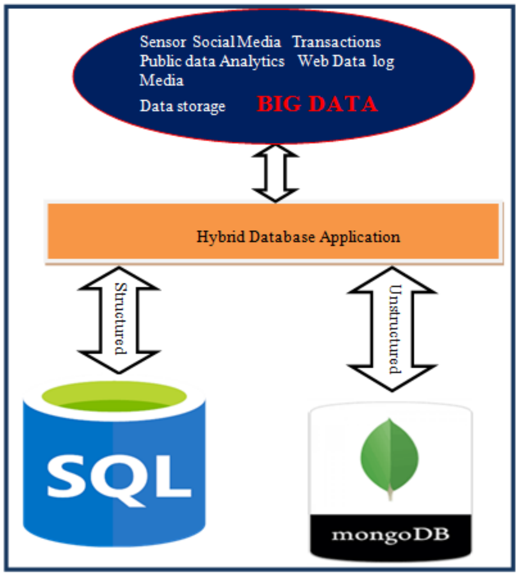
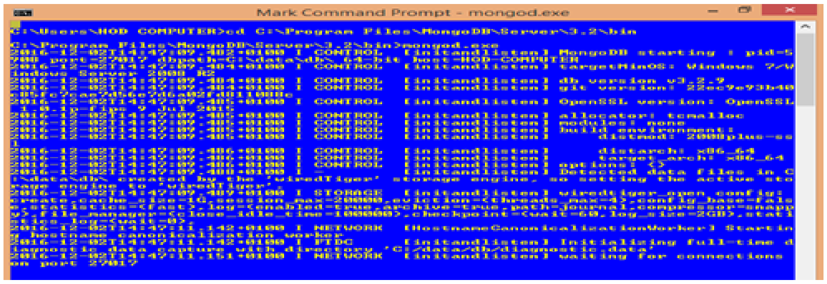
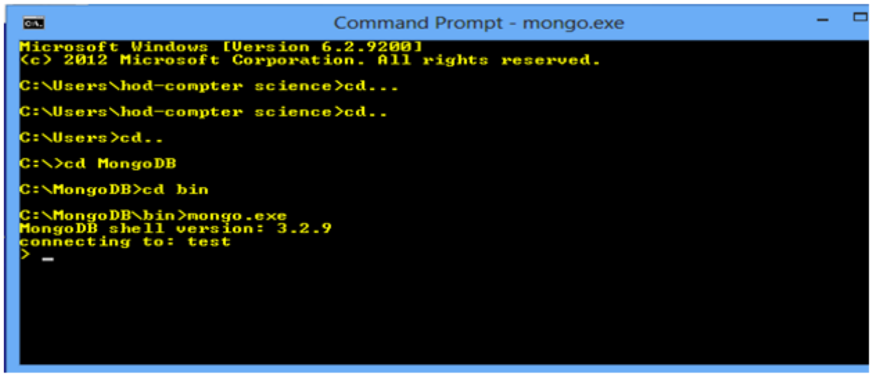

Sensor SocialMedia Transactions
Public data Analytics Web Data log
Media
Data storage
BIG DATA
Hybrid Database Application
Structured
Unstructured
SQL
mongoDB

Figure 3. Architectural design of the proposed system



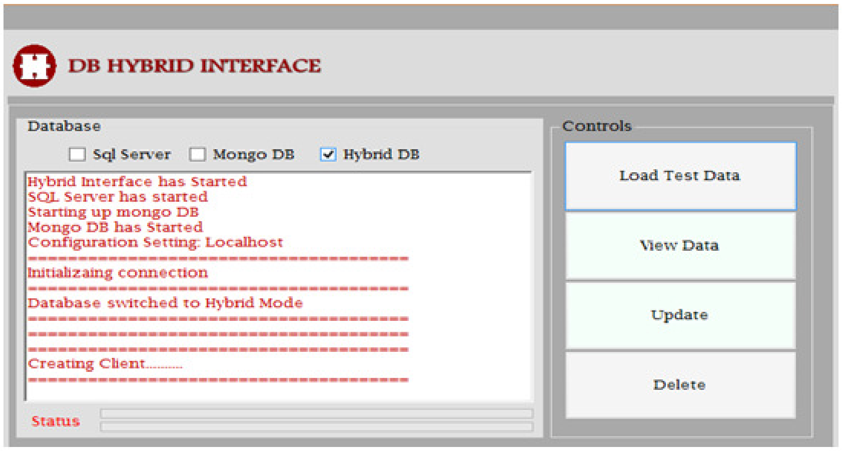
```
C:\Users\HOD COMPUTER>cd C:\Program Files\MongoDB\Server\3.2\bin

C:\Program Files\MongoDB\Server\3.2\bin>mongod.exe
2016-12-02T14:47:09.402+0100 I CONTROL  [initandlisten] MongoDB starting : pid=5
900 port=27017 dbpath=C:\data\db\ 64-bit host=HOD-COMPUTER
2016-12-02T14:47:09.404+0100 I CONTROL  [initandlisten] targetMinOS: Windows 7/W
indows Server 2008 R2
2016-12-02T14:47:09.404+0100 I CONTROL  [initandlisten] db version v3.2.9
2016-12-02T14:47:09.404+0100 I CONTROL  [initandlisten] git version: 22ec9e93b40
c85fc7cae7d56e7d6a02fd811088c
2016-12-02T14:47:09.405+0100 I CONTROL  [initandlisten] OpenSSL version: OpenSSL
 1.0.1p-fips 9 Jul 2015
2016-12-02T14:47:09.405+0100 I CONTROL  [initandlisten] allocator: tcmalloc
2016-12-02T14:47:09.405+0100 I CONTROL  [initandlisten] modules: none
2016-12-02T14:47:09.405+0100 I CONTROL  [initandlisten] build environment:
2016-12-02T14:47:09.405+0100 I CONTROL  [initandlisten]     distmod: 2008plus-ss
l
2016-12-02T14:47:09.406+0100 I CONTROL  [initandlisten]     distarch: x86_64
2016-12-02T14:47:09.406+0100 I CONTROL  [initandlisten]     target_arch: x86_64
2016-12-02T14:47:09.406+0100 I CONTROL  [initandlisten] options: {}
2016-12-02T14:47:09.408+0100 I -        [initandlisten] Detected data files in C
:\data\db\ created by the 'wiredTiger' storage engine, so setting the active sto
rage engine to 'wiredTiger'.
2016-12-02T14:47:09.409+0100 I STORAGE  [initandlisten] wiredtiger_open config:
create,cache_size=1G,session_max=20000,eviction=(threads_max=4),config_base=fals
e,statistics=(fast),log=(enabled=true,archive=true,path=journal,compressor=snapp
y),file_manager=(close_idle_time=100000),checkpoint=(wait=60,log_size=2GB),stati
stics_log=(wait=0),
2016-12-02T14:47:11.142+0100 I NETWORK  [HostnameCanonicalizationWorker] Startin
g hostname canonicalization worker
2016-12-02T14:47:11.142+0100 I FTDC     [initandlisten] Initializing full-time d
iagnostic data capture with directory 'C:/data/db/diagnostic.data'
2016-12-02T14:47:11.151+0100 I NETWORK  [initandlisten] waiting for connections
on port 27017
```

Figure 4. Starting MongoDB



```
Microsoft Windows [Version 6.2.9200]
(c) 2012 Microsoft Corporation. All rights reserved.

C:\Users\hod-compter science>cd...

C:\Users\hod-compter science>cd..

C:\Users>cd..

C:\>cd MongoDB

C:\MongoDB>cd bin

C:\MongoDB\bin>mongo.exe
MongoDB shell version: 3.2.9
connecting to: test
> _
```

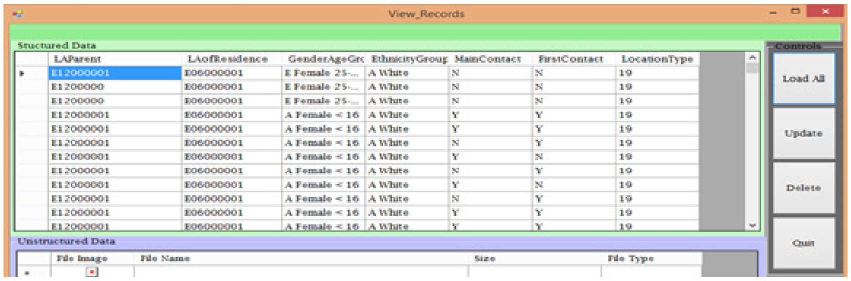
Figure 5. MongoDB connecting to default system database



☐ Sql Server ☐ Mongo DB ☑ Hybrid DB

```
Hybrid Interface has Started
SQL Server has started
Starting up mongo DB
Mongo DB has Started
Configuration Setting: Localhost
=========================================
Initializaing connection
=========================================
Database switched to Hybrid Mode
=========================================
=========================================
=========================================
Creating Client..........
=========================================
```

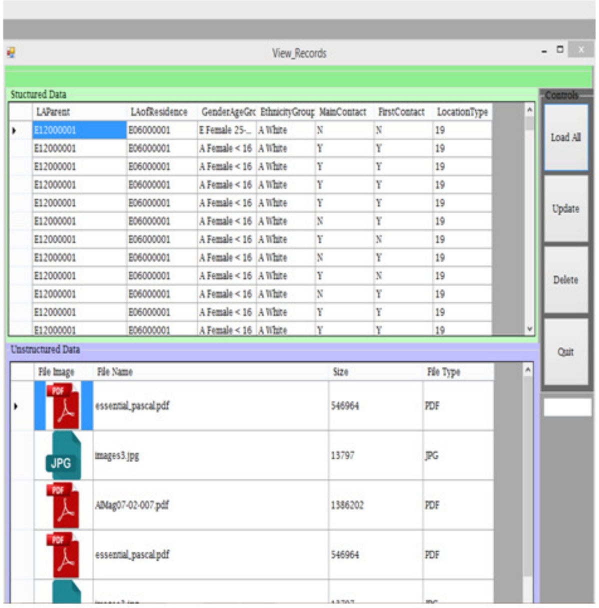
Figure 6. Overview of DB_Hybrid Interface



	LAParent	LAofResidence	GenderAgeGrc	EthnicityGroup	MainContact	FirstContact	LocationType
▸	E12000001	E06000001	E Female 25-...	A White	N	N	19
	E1200000	E06000001	E Female 25-...	A White	N	N	19
	E1200000	E06000001	E Female 25-...	A White	N	N	19
	E12000001	E06000001	A Female < 16	A White	Y	Y	19
	E12000001	E06000001	A Female < 16	A White	Y	Y	19
	E12000001	E06000001	A Female < 16	A White	N	Y	19
	E12000001	E06000001	A Female < 16	A White	Y	N	19
	E12000001	E06000001	A Female < 16	A White	N	Y	19
	E12000001	E06000001	A Female < 16	A White	Y	N	19
	E12000001	E06000001	A Female < 16	A White	N	Y	19
	E12000001	E06000001	A Female < 16	A White	Y	Y	19
	E12000001	E06000001	A Female < 16	A White	Y	Y	19

	File Image	File Name	Size	File Type
*				

Figure 7. View data operation in SQL mode.



	LAParent	LAofResidence	GenderAgeGrc	EthnicityGroup	MainContact	FirstContact	LocationType
▸	E12000001	E06000001	E Female 25-...	A White	N	N	19
	E12000001	E06000001	A Female < 16	A White	Y	Y	19
	E12000001	E06000001	A Female < 16	A White	Y	Y	19
	E12000001	E06000001	A Female < 16	A White	Y	Y	19
	E12000001	E06000001	A Female < 16	A White	Y	Y	19
	E12000001	E06000001	A Female < 16	A White	N	Y	19
	E12000001	E06000001	A Female < 16	A White	Y	N	19
	E12000001	E06000001	A Female < 16	A White	N	Y	19
	E12000001	E06000001	A Female < 16	A White	Y	N	19
	E12000001	E06000001	A Female < 16	A White	N	Y	19
	E12000001	E06000001	A Female < 16	A White	Y	Y	19
	E12000001	E06000001	A Female < 16	A White	Y	Y	19

	File Image	File Name	Size	File Type
▸	PDF	essential_pascal.pdf	546964	PDF
	JPG	images3.jpg	13797	JPG
	PDF	AIMag07-02-007.pdf	1386202	PDF
	PDF	essential_pascal.pdf	546964	PDF

Fig. 8: View data operation in hybrid mode.

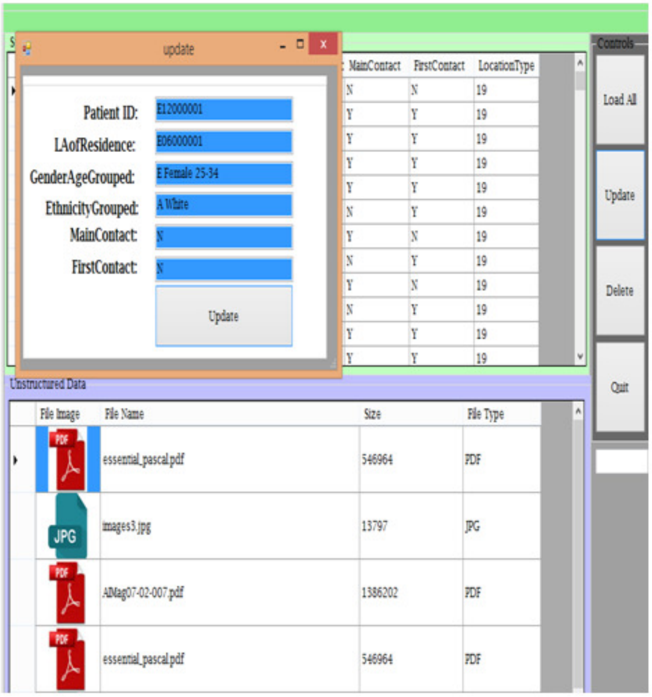
8. DISCUSSION OF RESULT

We loaded big data in our hybrid database system in SQL mode, MongoDB mode and also hybrid mode. The databases used for data storage and management vary depending on the mode that the system runs in. The output of the implemented hybrid database system for big data storage and management is discussed here;

In SQL mode, data is stored and managed in MySQL database. The system discards data in unstructured form since it cannot be stored in SQL database. A screen shot of the load data operation performed by our hybrid system in SQL mode is given in figure 7, a view of the database content shown in figure 8 shows that unstructured data is discarded by our hybrid application in SQL mode. In MongoDB mode, storage and management of both structured and unstructured data is performed using MongoDB database. This stores both structured and unstructured data in MongoDB. Also in hybrid mode, data in structured form is stored and managed using SQL database while MongoDB is used to store and manage unstructured data, this is shown in figure.9

From the output in figure 7, figure 8, and figure 9, it can be seen that our hybrid system integrates the functions or flavours of MySQL which is a popular relational database and MongoDB which is a non-relational database in one database system allowing the databases in the system to function in isolation and also in integration.

The proposed system supports managerial operations such as Update and Delete which can be used to manage data in the system. Figure 10 shows update operation in our hybrid database system.

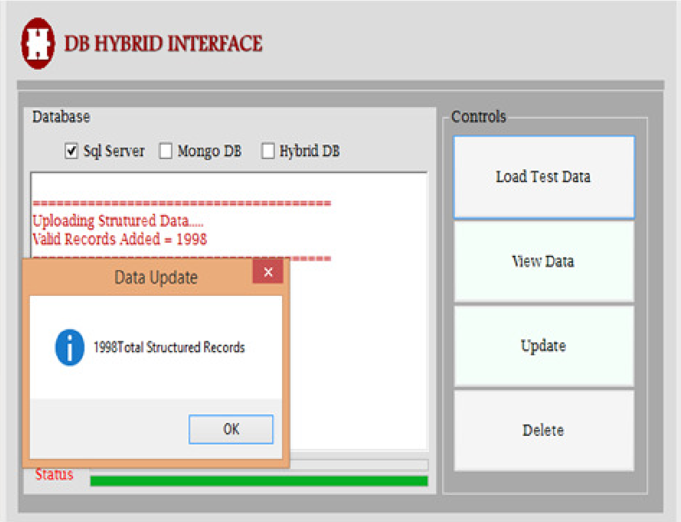


Patient ID: E12000001
LAofResidence: E06000001
GenderAgeGrouped: E Female 25-34
EthnicityGrouped: A White
MainContact: N
FirstContact: N

MainContact	FirstContact	LocationType
N	N	19
Y	Y	19
Y	Y	19
Y	Y	19
Y	Y	19
N	Y	19
Y	N	19
N	Y	19
Y	N	19
N	Y	19
Y	Y	19
Y	Y	19

File Image	File Name	Size	File Type
PDF	essential_pascal.pdf	546964	PDF
JPG	images3.jpg	13797	JPG
PDF	AIMag07-02-007.pdf	1386202	PDF
PDF	essential_pascal.pdf	546964	PDF

Figure 9. Update operation



☑ Sql Server ☐ Mongo DB ☐ Hybrid DB

```
========================================
Uploading Strutured Data.....
Valid Records Added = 1998
========================================
```

1998Total Structured Records

Figure 10. Load data operation in SQL mode.

9. CONCLUSION AND RECOMMENDATION

In conclusion, we proposed a method which combines SQL database which belongs to the relational group of database systems and MongoDB being a NoSQL database to store and manage big data. With the result obtained it is understandable that our system can be used for storage and management of big eliminating the weaknesses in both databases.

10. CONTRIBUTION TO KNOWLEDGE

This work presents the following contributions to knowledge;

1. Development of a hybrid database system for big data storage and management.
2. This approach improves on the use of MongoDB database for big data storage.
3. The study establishes the possibility of having the flexibility and scalability of NoSQL database and also the stability and transactional ingredients of a relational database in one database management system.

REFERENCES

[1] Codd, E. F. (1970). "A relational model of data for large shared data banks", Communications of the ACM Vol. 13, No. 6, pp 377–387.

[2] Hecht, R., Jablonski, S. (2011) " NoSQL evaluation: A use case oriented survey" Cloud and Service Computing (CSC) International Conference. pp336 – 341

[3] Konstantinou, I., Angelou, E., Boumpouka, C., Tsoumakos, D. and Koziris, N. (2011) "On the elasticity of NoSQL databases over cloud management platforms" In Proceedings of the 20th ACM international conference on Information and knowledge management (CIKM '11), New York, NY, USA

[4] Van der Veen, J.S., van der Waaij, B.,Meijer, R.J.(2012) "Sensor Data Storage Performance: SQL or NoSQL, Physical or Virtual" IEEE 5th International Conference on Cloud Computing (CLOUD) pp. 431 – 438.

[5] Singh, N., Garg,N.& Mittal,V.(2012). "Big data- Insight, Motivation and Challenges" Internal Journal of Scientific and Engineering Research, Vol. 4

[6] Schram, A., Anderson, K. M. (2012) "MySQL to NoSQL: data modelling challenges in supporting scalability" Proceedings of the 3rd annual conference on Systems, programming, and applications: software for humanity (SPLASH '12). ACM, New York, NY, USA, pp. 191-202.

[7] Indrawan-Santiago, M. (2012). "Database Research: Are We at a Crossroad? Reflection on NoSQL." Proceedings of the 15th International Conference on Network-Based Information Systems, pp 45-51.

[8] Tauro Clarence T.,Patil,Baswanth R., Prashant K.V.(2013). "A comparative analysis of different NoSQL databases on data model, query model, and replication model" Internal Conference on Emerging Research in Communication and Application ERCICA. Bangalor India

[9] Nayak, A. ,Poriya ,A.,Dikshay Poojary (2013)" Types of NoSQL databases and its comparison with relational databases". International Journal of Applied Information Systems Vol.5, No. 4 pp 16-19

[10] Grolinger, K.,Higashinow,T. Wari,Capretz,M.AM (2013)"Data Management in cloud environments: NoSQL and NewSQL data stores" Vo l2. No. 22

[11] Porkony, J. (2013). "NoSQL databases: A step to database scalability in web environment.", International Journal of Web Information Systems • Vol. 9, No.1 pp 69-82.

[12] Wu, L., Yuan,L., Huai, Y. (2013). "Survey of large scale data management system for big data applications" Journal of Computer Science and Technology. Vol. 30 pp 163-183.

[13] Moniruzzaman, A .B., Hossain, S.A.(2013). "NoSQL database: New era of databases for big data analytics-classification, characteristics and comparison" International Journal of Computer Science and Technology. Vol. 6 No. 4

[14] Nawsher Khan, Ibrar Yaqoob, Ibrahim Abaker Targio Hashem, Zakira Inayat,Waleed Kamaleldin Mahmoud Ali,Muhammad Alam,Muhammad Shiraz,Abdullah Gani (2014). “Big Data: Survey, Technologies, Opportunities, and Challenges” The Scientific World Journal Vol. 2014 , No. 712826

[15] Abramova, V., Bernardino,J., Furtado,P.(2014). “Experimental evaluation of: NoSQL databases” International Journal of Database Management System .Vol.6, No. 3

[16] Abramova, V., Bernardino,J., Furtado,P.(2014). “Which NoSQL database? A performance overview” Open Journal of Databases (OJDB). Vol. 1, Issue 2

[17] Wu, L., Yuan, L., You, J. (2014). “BASIC: An alternative to BASE for large scale data management systems”, https://webdocs.cs.ualberta.ca/~yuan/papers/ieee_bigdata.pdf

[18] Manoj,V.(2014) “Comparative study of NoSQL document, column store databases and evaluation of Cassandra” International Journal of Database Management Systems (IJDMS) Vol.6, No.4

[19] Rakesh, K., Shilpi, C., Somya, B(2015).”Effective way to handling big data problems using NoSQL database (MongoDB)” Journal of Advanced Database Management & Systems. Vol. 2, pp 42–48.

[20] Wu,C.M.,Huang,Y.F.,Lee,J.(2015). “Comparisons between MongoDB and MS-SQL databases on the TWC website” American Journal of Software Engineering and Applications. Vol. 4 No.2 pp 35-41.

[21] GC, Deepak (20016) "A Critical Comparison of NOSQL Databases in the Context of Acid and Base" Culminating Projects inInformation Assurance. Paper 8.

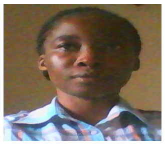
[22] Mukherjee, S.,Shaw R.(2016) “Big data concepts, applications, challenges and future scope” International Journal of Advanced Research in Computer and Communication Engineering Vol. 5, Issue 2

[23] http://refreshmymind.com/wpcontent/uploads/2016/02/nosql_documents_stores_mongodb couchdb.png

[24] Albertas, Krisaunas, Benefits of NoSQL. https://www.devbridge.com/articles/benefits-of-nosql/#

[25] Why NoSQL, http://img.blog.csdn.net/20151001143133915

AUTHORS



Blessing E.James is a Graduate Assistant in the Department of Computer Science ,Akwa Ibom State University, She obtained a Bachelor of Technology degree in Mathematics and Computer Science, 2010 from the Federal University of Technology Owerri, Imo State, Nigeria and a Master of Science degree (2007) from the University of Port Harcourt, Nigeria. Her interest is on Big data, Database Management Systems, Database Knowledge Management, Data Mining and Machine Learning.

Prince Oghenekaro ASAGBA had his B.Sc degree in Computer Science at the University of Nigeria, Nsukka, in 1991 with a Second Class (Hons) Upper Division.He had his M.Sc degree in Computer Science at the University of Benin in April, 1998, and a Ph.D degree in Computer Science at the University of Port Harcourt in March, 2009. Asagba is an avid researcher with over fifty researched published articles in reputable journals, both locally and internationally. Asagba possesses over 20 years of reasonable wealth of research / teaching experience at the University level. He is a Visiting Professor / Scholar to some Universities in Nigeria. His research interest include: Computing and Information Security, Network Analysis, Software Engineering, Database Management Systems, Modelling and Programming. He is a member of Nigeria Computer Society (NCS) and Computer Professional Registration Council of Nigeria (CPN).