Design and Implementation of an Improved Query Processing Technique for Big Data Management

Friday E. Onuodu[1], Stanley C. Nwafor[2]
[1]*Department of Computer Science, University of Port Harcourt, Nigeria*
[2]*Department of Computer Science, Ignatius Ajuru University of Education, Rivers State, Nigeria*

***Abstract*—The processing of huge amount of data is usually a tedious task for big multi enterprises and corporate organizations, hence the need for an improved query processing technique to handle big data. Big Data management tools and techniques are rising in demand due to the use of Big Data in businesses. Organizations can find new opportunities and gain new insights to run their business efficiently. These tools help in providing meaningful information for making better business decisions. The companies can improve their strategies by keeping in mind the customer focus. Big data analytics efficiently helps operations to become more effective, thereby, improving the profits of the company. This study investigates an Improved Query Processing Technique for Big Data Management. The Proposed System was implemented with PHP Programming Language due to its unique Graphical User Interface as Frontend, and MySQL Relational Database Management System (RDBMS) as backend. Furthermore, we adopted Structured System Analysis and Design Methodology (SSADM) in this approach. Our results show an increase in the efficiency of the business and analysis of data becomes easier and quicker. This, in turn, leads to fast and better decision making, saving time and energy. The insights provided by the big data analytics tools could help in knowing the needs of customers better. This work will be beneficial to companies that handle big data and to the customers of such companies.**

***Keywords*—Big Data, Query Processing, record, Database management System (DBMS), NoSQL.**

I. INTRODUCTION

Query processing denotes the compilation and execution of a query specification usually expressed in a declarative database query language such as the structured query language (SQL). Query processing consists of a compile-time phase and a runtime phase. At compile-time, the query compiler translates the query specification into an executable program. This translation process (often called query compilation) is comprised of lexical, syntactical, and semantically analysis of the query specification as well as a query optimization and code generation phase.

The code generated usually consists of physical operators for a database machine. These operators implement data access, joins, selections, projections, grouping, and aggregation. At runtime, the database engine interprets and executes the program implementing the query specification to produce the query result. The performance of a database management system (DBMS) is fundamentally dependent on the access methods and query processing techniques available to the system. Traditionally, relational DBMSs have relied on well-known access methods, such as the ubiquitous B+-tree, hashing with chaining, and, in some cases, linear hashing. Object-oriented and object-relational systems have also adopted these structures to a great extent.

During the past decade, new applications of database technology requirements for non-standard data types and novel update and querying capabilities have emerged that motivate a re-examination of a host of issues related to access methods and query processing techniques.

Big data is a term that describes the large volume of data. It can be structured or unstructured. But it's not the amount of data that's important. But what matters is, what organizations do with this data. Big data can be analyzed for insights which lead organizations to take better decisions and also help them in making strategic business moves.

The rapid increase in the volume of data produced lately has however given great opportunities for significant achievements such as improved business strategies, transformed financial services, healthcare methods etc. [1]

The use of non-relational database systems has raised sustainability over the past few years due to benefits such as scalability, and a compatibility with data [2].

Big Data is a general term used to refer to massive and complex datasets which are made of a variety of data types (structured, semi structured and unstructured data) from a multitude of sources. Data volumes are continuing to grow so are the possibilities of what can be done, while big data offers a ton of benefits, it comes with its own set of issues.

Adopting Big Data-based technologies not only mitigates the problems stated in course of this research, but also opens new perspectives that allow extracting value from Big Data. Big Data-based technologies are being applied with success in multiple scenarios such as e-commerce and marketing where count the clicks that the crowds do on the web allow identifying trends that improve campaigns, evaluate personal profiles of a user so that the content shown is the one he will most likely enjoy; government and public health, allowing the detection and tracking of disease outbreaks via social media or detect frauds; transportation, industry and surveillance,

with real-time improved estimated times of arrival and smart use of resources [3] It also focuses on addressing the design, evaluation and operation of the current problems with Big Data applications. With efficient means to manage big data, other processing functions such as analysis and computations could be performed for sake of leveraging the enormous opportunities of big data.

A. Aim and Objectives

The aim of this study is to develop an improved query processing technique for big data management. The specific objectives of this study are to:

i. design a Query Processing system to help organizations and multi enterprises to handle big data within such organizations.
ii. implement with PHP Programming Language as frontend interface and MySQL Relational Database Management System due to its flexibility (RDBMS) as backend.
iii. compare our results with other existing ones.

B. Database Management Systems

A Database Management System is an intermediary between database applications and database. The DBMS creates and manages the database. DBMS can be categorized based on its data model. The database systems are based on some data models. Data models describe the logical structure of data items and their associated operations. Create, update, read and delete operations are four basic ways to interact with a database. They are often known as CRUD operations. The Structured Query Language (SQL) is a standard way to execute CRUD operations on database. The DBMS acts as a gatekeeper. All the information going in or out of database must pass through the DBMS. It is a critical mechanism for maintaining quality of data and database. Users and database applications are not allowed directly to interact with database. Database is a collection of data items that provides an organizational structure for information storage and management. The Database also provides a mechanism for querying, creating, modifying and deleting data. A list can also be used to store data but in a list, redundancy is a major issue. A database can store relationships and data that are more complicated than a simple list with lesser or no redundancy.

Database management system consists of database users, database applications and Database Management Systems (DBMS). Database users need not to be always human. It is possible, for example, for other software programs to be users of the database. Users interact with database application and application further depends on the DBMS to extract and store data in the database.

C. Nosql Database Management System

Although traditional relational database management systems (RDBMS) have existed over decades and are constantly being improved by database vendors, RDBMS struggle to handle extremely large volumes of data produced in recent times. Not only structured Query Language (NoSQL) as first used by Carlo Strozzi in 1998, for his open source stated that relational database do not offer an SQL interface. NoSQL data stores were developed basically to address the challenges of traditional databases. NoSQL data stores in recent years. Especially in the cloud based companies and service providers.

Now NoSQL data stores have wide acceptance in variety of industries ranging from manufacturing, oil and gas, energy, banking and health care. NoSQL databases are very important components of big data for storing, managing and retrieving large volumes of data; they follow relatively weaker consistency model “BASE’ stands for "Basically Available, Soft state, Eventually consistent". BASE brings a softer consistency model. Basically Available means the data stores assure system availability in terms of CAP theorem. Soft State states that the system state may change over period of time even if no input is given. Finally Eventual Consistent indicates that the system eventually become consistent with time if system is not fed with any input during that time [4].

There are four types of NoSQL databases: Key-Value Data stores, Document-Oriented, Wide Column (Column Family) store and Graph Store [5]

Key-Value datastores: stores data in form of matched pairs with only two columns permitted, a distinctive identifier called key and a corresponding value. The values can be simple or complex data types such as sets of data.

Document-Oriented Stores: stores related information in the form of a document. The values are stored in documents as lists or nested documents. Few examples are MongoDB, SimpleDB, and CouchDB.

Wide Column (Column Family) store: These data stores have a format very similar to that of RDBMS but are much more flexible than the RDBMS.

Graph Stores: supports data that has undefined number of network connections. This type of data supports map data, bus transportation and relationships found in social media.

D. Big Data

Big data refers basically to group of datasets which are completely voluminous (pentabyte and terabyte) consisting of various data types (structured, semi-structured, unstructured and of real time availability i.e velocity) such that it is not efficient to be stored or processed with conventional database systems. [6] Pointed out that big data refers only to collection of voluminous data, but extremely large volume of structured

and unstructured data subject to very high rate of change, derived from various components such as e-mail, social media, sounds, images.

These three attributes are traditionally referred to as the 3 Vs of big data [7]. Big data can be described with the following features:

i. *Volume*: The quantity of generated and stored data. The size of the data determines the value and potential insight and whether it can be considered big data or not.

ii. *Velocity*: In this context, the speed at which the data is generated and processed to meet the demands and challenges that lie in the path of growth and development. Big data is often available in real-time. It entails the rate at which data is circulated within the system.

iii. *Variety:* The type and nature of the data set. This helps people who analyze it to effectively use the resulting insight. Big data draws from text, images, audio, video; plus it completes missing pieces through data fusion.

No matter the source of data, it could be categorized into one class of data [8] Data can be grouped into the followings groups:

i. *Unstructured Data*: data sets such as text, and multimedia contents such as mails, word processing documents, videos, photos audio files and other forms of business documents. They are called unstructured because they do not assume predetermined form. Unstructured data is everywhere and most individuals and organization conduct their lives around unstructured data.

ii. *Structured Data: concerns* all data which can be stored in database SQL in table with rows and columns. They have relational keys and can be easily mapped into predesigned fields. They are organized in predetermined format in tables, spreadsheets etc. and also most suitable for relational database management systems.

iii. *Semi (Multi) Structured Data:* these consists data that does not reside in relational database but have some organizational properties that make them easy to analyze. They are neither completely structured nor completely unstructured.

Big data challenges refer to the steps taken in the processing of this voluminous data types. To leverage the numerous benefits of big data must overcome the following challenges:

- Privacy and Security: this is the most important issue with big data which is sensitive and includes conceptual, technical as well as legal significance. The personal information of a person or clients should be held with outmost respect.
- Storage: the storage of big data is very complex; it includes searching and retrieval of data. Storage framework for big data should be able to hold extremely large volumes of various data types efficiently. Building up indexes right in the beginning while collecting and storing data is a good practice.
- 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. The type of analysis to be done on huge amount of data irrespective of the data type requires a large number of advance skills.
- Data Access and Sharing: if data to be used to make accurate decisions in time it becomes necessary that it should be available in accurate, complete and timely manner [9] This makes the data management and storage process a bit complex sharing data about clients of an organization and operations threatens the culture of secrecy and competitiveness.

II. RELATED WORKS

Khan et al [10] surveyed and classified big data based on the characteristics of data, rate at which it increases, quantity of data, method of data processing and related security issues. In their article, a data life cycle was proposed using the models and vocabulary as regards to big data frameworks. The cycle consist of phases which include data gathering, depurating, scrutinizing, storing and publication which jointly transform raw fact into published information which is an integral part in experimental data organization. They came to a conclusion that the separation of beneficial information from enormous inflow of fact is one of the pressing issues of big data. They however were unable to point out specifically other issues surrounding big data and how it can be resolved.

Baker et al [11] proposed the use of megastore; an extensible, high availability data store suitable for use in modern interactive environments. Unlike NoSQL data stores, megastore supports hierarchical data model where tables are linked onto big table. It uses big table to store replica in a data centre. Megastores possess a fascinating model which is prominent because its implementation of paxox is kept at shallow latency level. Although megastore has been proven to be highly scalable, its major setback is that at the development stage of the database data must be grouped into entity classes such that a particular transaction cannot gain access to data outside the class to which the entity belongs. Therefore it does not support the concept of Availability, Consistency, Isolation and Durability beyond replication group

Moniruzzaman and Hossian [12] presented a report on the classification, properties and comparison of NoSQl databases. They explored the strength and weaknesses of the various NoSQL data stores. NoSQL database is a non-relational distributed database system built for the storage of big data and also for massive concurrent processing of data over a collection of commodity servers. A detailed comparison between databases in the domain of NoSQL revealed that databases in the document class use volatile memory systems, conditional entry updates and compression protocols. The internal structure of these models is a shared nothing architecture which makes it horizontally scalable. It uses master-slave replication to share data across servers.

Kumar et al [13] proposed the use of MongoDB which is a varient of NoSQL database for effective handling of big data. They discussed reasons for fall of other databases as the reason for raise in the use of MongoDB.

The system was implemented on Ubuntu 14.04 LTS using MongoDB to show effective way to handling big data problems. The system is very scalable because MongoDB runs on a distributed architecture.It can store both structured and unstructured data. However, it is not very efficient in heavily transactional environments and also does not support joints.

Abbes and Gargouri [14] proposed a MongoDB database and modular ontologies-based approach for big data integration. They illustrated an approach for ontology based big data integration taking into account their characteristics. Their approach is based on a NOSQL database namely MongoDB and modular ontology. At the implementation stage, after wrapping each data source to a corresponding MongoDB database, they generated an ontology corresponding to each data source by means of transformation rules from MongoDB to the ontology representation language OWL. Then, the resulting ontology was into a global one.

III. MATERIALS AND METHODS

Taking into consideration all other works related to this, we chose the work of Kumar et al (2015). This is driven the strength and weaknesses of this approach. They proposed a NoSQL approach to big data storage which has proved a key enabler to efficiently analyze large amounts of data and create additional value. The system is implemented using MongoDB which is one of the most used NoSQL database management systems. MongoDB is a document-oriented database with features such as scalability, sharing and ability to store big data. This research looks at big data management using MongoDB which will help in eliminating the weaknesses and challenges of big data.

A. Existing System

The data stores for some applications need to provide good horizontal scalability. Relational databases cannot solve these problems as they are not horizontally scalable. Similar issues are involved with cloud services, social networks, mobile usage and social media. Due to cloud services data is growing fastly and the data accumulated by social networks is more connected and semi-structured.

There is need for a database system that can store and process, store and manage big data effectively and can satisfy the demand for high performance while reading and writing. Relational databases have many problems to cope with these trends. Technologies such as NoSQL were developed to meet the reliability and scalability needs. A growing number of developers and users have begun to use NoSQL databases. With this usage, there arose need to transform the existing applications from relational databases to NoSQL databases so that they can run on such platform.

Document-oriented databases are one of the categories of NoSQL databases that are appropriate for web-applications which involves storage of semi-structured data and dynamic queries. MongoDB document databases are able to face these new challenges as they allow horizontal scalability, support high-availability and have the flexibility to handle semi-structured and unstructured data. MongoDB has typical applications in content management systems, mobiles, gaming and business transactions.

B. Disadvantages of the Existing System

After some online researches and interviews, we were able to spot out some major setbacks of the existing system as discussed.

i. Complex transactions: relational databases do not support multi-document transactions. With the introduction of the NoSQL databases, support for ACID transactions across documents was typically thrown away.
ii. Inability to store large volumes of data: there is a limit to the amount of data a relational database can store. This has become a key limitation considering the rapid change in data types.
iii. Performance and Speed: One other major setback of the existing system is its ability to handle large unstructured data. On the contrary, MongoDB is magically faster because it allows users to query in a different manner that is more sensitive to workload.
iv. High availability in an unreliable environment: Setting replica Set (set of servers that act as Master-Slaves) is easy and fast. Moreover, recovery from a node (or a data centre) failure is instant, safe and automatic.
v. High Speed: The proposed system takes advantage of the high speed of MongoDB when querying data. It provides faster data access irrespective of data volume.
vi. Horizontal Scalability: the storage capacity of the MongoDB in the proposed system could be optimized when required.

C. *Proposed System*

This application is based on 3-tier architecture of PHP as shown in figure 3.2. The 3-tier includes the three hierarchy of the flow of programming logic from user interface to database and again database to user interface with the desired information requested by the clients. In between there involves the logic layer for effectively and correctly manipulating the request. The 3-tier includes:

i. *Application User:*The visual part is implemented using all kinds of dot net framework components, which does not make databasecalls. The main function of this tier is to display information to the user upon user's request generated by user's inputs such as firing button events. For example, inventory list will display when user click "display" button if he or she wants to know the list of stock remaining in the organization.

ii. *Inventory Management System*: The middle tier, business transactions (inventory system) called by the client to make data document queries. It provides core function of the system as well as connectivity to the data tier, which simplify tasks that were done by the client tier.

iii. Database: Database store is also the class that gets the data from the business tier and sends it to the database or gets the data from the database and sends it to business tier. This is the actual MongoDB access layer or object layer also called the business object. The database back-n end stores information that can be retrieved by using the MongoDB data Connectivity. MongoDB data connectivity is used to manage the communication between the middle tier and the backend database by issuing complex document queries.

The proposed system design could be seen as the application of systems theory to product development. In order to overcome the limitations and challenges in the existing system, it is ideal that we propose a database system for the efficient storage and management of big data for an inventory management system usingmongoDB. The proposed system will enable the storage of large volume of data as the user need and database increases as well as provide high speed and availability to data. This is a result of the inability existing system to save data of various types We employ the strength of MongoDB which is an open source document-oriented NoSQL database store and its support for dynamic schemas.

D. *Algorithm*

I START

II. LAUNCH MONGODB SOFTWARE

III. INPUT VALID USERNAME

IV. INPUT VALID PASSWORD

V. IF (USERNAME AND PASSWORD = DATABASE ENTRY)

VI. RETURN TO START PAGE

VII. LOAD DATA

VIII. TEST IF PRODUCT NUMBER EXISTS

IX. DISPLAY ERROR MESSAGE

X. RETURN TO PRODUCTS PAGE IF EXISTS

XI. SEARCH FOR RECORDS THAT MEET A

SEARCH PATTERN

XII. RETURN RECORDS THAT MEET THE

PATTERN

XIII. CONDUCT BIG DATA SEARCH

XIV. OUTPUT RESULTS OF SEARCH

XV. ADD NEW RECORDS TO DATABASE

XVI. PRODUCE SUMMARY REPORTS

XVII. RETURN TO START PAGE

XVIII. STOP

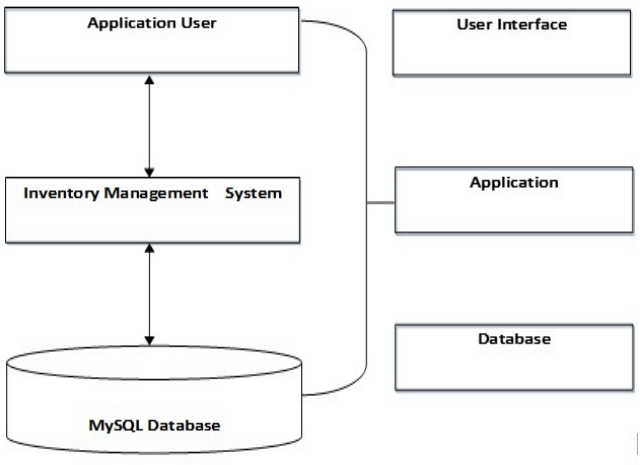

Fig.1. Architecture of the existing System (**Source:** Kumar et al [12]**).**

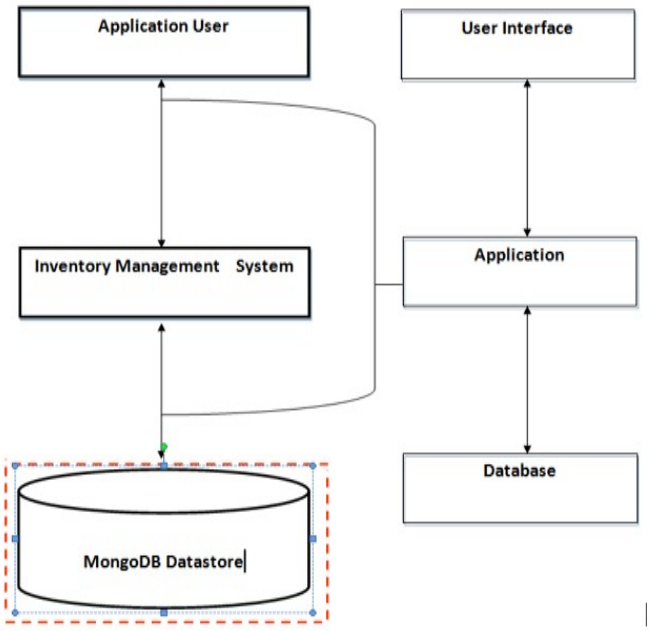

Fig.2 An Enhanced Keyword-based NLQ to SQL Conversion System (Proposed System)

IV. RESULTS AND DISCUSSION

User interfaces are the access points where users interact with designs. Graphical user interfaces (GUIs) are designs' control panels and faces. User interface design or UI design generally refers to the visual layout of the elements that a user might interact with in a website, or technological product. This could be the control buttons of a radio, or the visual layout of a webpage. User interface designs must not only be attractive to potential users, but must also be functional and created with users in mind. User interface design can dramatically affect the usability and user experience of an application.

User interface designs should be optimized so that the user can operate an application as quickly and easily as possible. The user interface is shown in Figure 3.

Fig.3. User interface layout design of the Start Up Page

Fig.4. Product Addition Page

Fig.5. Big Data Search Page

Figure 5 Shows the Big Data Search Page where the search is authorized and conducted. The page requests the administrator to enter the Product name as ewell as the Product ID and then click on the 'PERFORM BIG DATA SEARCH' buton to perform the search for a partcular item within the Mongo Database.

Figure 4. shows the Product addition Page to the MongoDBFor easy management of the big data, a friendly user interface was developed for database administration. All data entered into the system can be accessed and manipulated via MongoDB Compass for further analysis.

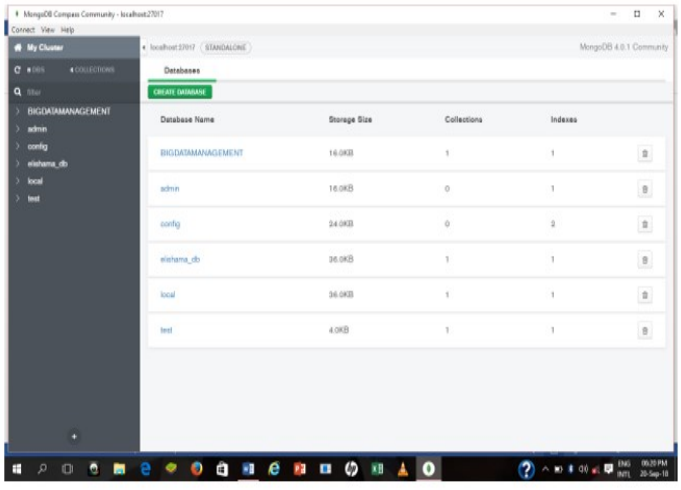

Fig.6. MongoDB

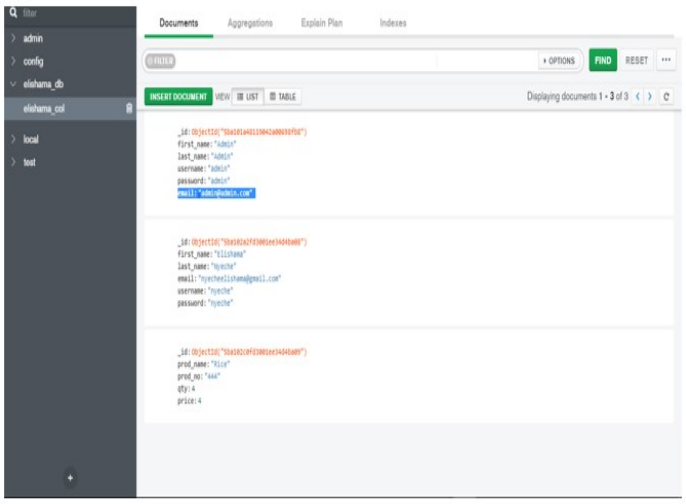

Fig.7. MongoDB

V. CONCLUSIONS

Finally, we implemented an Inventory Management System driven by Big Data and MongoDB Administration tools. This approach provides efficient data storage, which will be appreciated when the data grows bigger beyond what a traditional database management system, such as MySQL, can handle. From the results obtained and our evaluation of the system discussed in the previous chapter, it is clear that big data, specifically MongoDB, is an efficient choice tool for database management.

The research work was limited to only the use and application of MongoDB which is a variant of the NoSQL databases for storage and management of data. There should be an application of other types of the NoSQL database such as the column family, graph family and the key/value family in implementing big data storage and management.

REFERENCES

[1] Khan, N., Yaqoob, I., Hashem, I., Inayat, Z., Ali.W., Alam, M., Shiraz, M., Gani, A., (2014). Big data: Survey, Technologies, Opportunities, and challenges. The scientific World journal. Doi:10.1155/2014/712826

[2] Shi, Y., Meng, X., Zhao, J., (2010) Benchmarking cloud-based data management systems. In: proceedings of the second International workshop on cloud data management, New York, NY, USA: ACM.

[3] Borkar, V.R., Carey, M.J., Li, C. (2012). Big Data platforms: What's next? 19(1),44-49.

[4] Eric,Brewer (2000). Principles of Distributed Computing "towards robust distributed systems proceedings of the nineteenth annual ACM symposium , doi:10.11 45/343477.343502

[5] Abramova V., Bernardino, J., Furtado, P.(2014). Which NoSQL database? A performance overview.Open journal of Databases (OJDB), 1(2) ISSN 21993459.

[6] Sathi, A. (2012): Big data analytics: disruptive technologies for changing the game. Boise: Mcpress ISBN 10:1583473807.

[7] Hurt, J., (2012), Http://velvetchainsaw.com/2012/07/20/ three-vs-of-big-data-as-Applied-conference. Retrieved October, 4, 2014.

[8] Magoulas, R., Lorica B. (2009) Big data: technologies and techniques for large scale data, page 32 ISSN 1935-9446.

[9] Khan, Z., Anjum, A., Kiani. S.,(2013), conference proceeding of the IEEE 6th international conference of Utility and cloud computing. Doi.10.1109/UCC.2013.77

[10] Katal, A., Mohammed,W., Goudar, H.(2013). Big data: issues, challenges, tools and good practices. 2013 sixth international conference on comtemporary computing (IC3) DOI:10.1109/IC3.2013.6612229..

[11] Baker, J., Bond. C Cobert. J., Furman, J., Khorlin. ., Leon. J., Li. Y(2011)Megastore: providing scalable, highly available storage for interactive services. Proceedings of the 5th Biennial Conference on innovative Data Systems research. California, USA.

[12] Moniruzzaman, A.B., Hossain, S.A.(2013). NoSQL database: New era of databases for big data analytics, classification, characteristics and comparison. International Journal of database theory and application (IJDTA). 6(4)

[13] Kumar, R., Shilpi, C., Somya, B. (2015) Effective way to handling big data problems using NoSQLdatabase(MongoDB). Journal of Advanced Database Management &Systems (JADMS). 2(2):42-48.

[14] Abbes, A., Gargouri, F. (2016) Big data integration: A MongoDB database and modular ontologies based approach. Proceedings of the 20th International Conference on Knowledge Based and Intelligent Information and Engineering Systems, KES2016, New York, UK.