Journal of Engineering and Applied Sciences 12 (Special Issue 7): 8173-8179, 2017
ISSN: 1816-949X

A Study on Output Performance of NoSQL Data Processing over RDBMS in Big Data

S. Srividyaa and R. Varalakshmi
Department of Computer Applications (MCA), Vels University,
Pallikaranai, 600100 Chennai, Tamil Nadu, India

Abstract: Due to increasing of users and usage in social network with smart phones, data processing will be the difficult task and it becomes an issue. Typically, such data is unstructured and which receives from multiple sources in different format. Consequently, the abstraction of data for rendering is difficult that lead to the development of a computing system that can store data in unstructured format and support distributed parallel computing. In existing approaches to handle big data, a new method was using NoSQL. This study is about a study of output performance review of NoSQL and Relational Database Management System (RDBMS). By reviewing each approach, the mechanics of NoSQL systems can be clearly distinguished from the existing RDBMS. Basically such methods depend on multiple factors which include the query language, architecture, data model and consumer API. In this study, we are also defining the application that matches the system and subsequently able to accurately correlates to a specific NoSQL system, PostgreSQL and MongoDB have been selected to represent RDBMS and NoSQL, respectively for approximate analysis.

Key words: NoSQL data processing technique, structured data, unstructured data, big data processing, MongoDB, PostgreSQL

INTRODUCTION

In existing methods big data received attention for its volume of data. However, due to development of various applications not only text data but also various multimedia data that does not fit the existing report formats are being generated in real-time. Relational Database Management System (RDBMS) that has been being utilized has issues in structuring unstructured data and has performance cost problems in processing massive data. Having memory mapping function, NoSQL performs read write fast which makes NoSQL suitable for processing big data. In addition, unlike RDBMS that processed mainly of structured data, NoSQL can handle unstructured data with more ease. Thus, many companies building big data system that includes unstructured and sensor data, tend to take advantage of NoSQL's features. This study evaluates the performance changes between RDBMS and NoSQL and provides optimal design to improve process when using NoSQL.

Modeling relational databases exists to avoid duplication and to establish a tree data structure for a set of tables that are connected. The modeling of relationships is operated from Foreign keys. The explicit declaration of relationships and types allow the database to consolidate a strong integrity of the data that it preserves. Data integrity is enhanced through the transactions which consent to perform operations for obtaining a healthy state. The NoSQL model fulfills the scalability problems. The horizontal scalability grants to replicate data from a database on different shards. A shard is a partition for storing data. The shards can be located on the same server or on different servers. MongoDB is a NoSQL database management system, oriented document that does not require a predefined physical schema which means that data can be enriched on the fly without reconfiguration or base change. MongoDB is a JSON object repository that replaces the concept of atomic transactions access to documents, joins, Foreign keys, unique keys auto-incremented by the concept of referencing or embedded document. MongoDB uses a storage document almost without constraints which offers a multitude of concepts within a single structure.

The NoSQL is used to satisfy user needs and accommodate the evolution of data which continue to rise in social networks and the new complex applications. However, as the major part of the data today are still stored in relational databases, resulted in the birth of new approaches to relational database migration to NoSQL. technology. Several processes have been proposed to establish the conversion of the RDBs to the NoSQL.

Corresponding Author: S. Srividyaa, Department of Computer Applications (MCA), Vels University, Pallikaranai, 600100 Chennai, Tamil Nadu, India

MATERIALS AND METHODS

NoSQL Models and features: The requirement of NoSQL database originates from the needs of online shopping, services offered via. web and online social media communication. These applications require the provision of reports, warning and data integrity for the improvement in the services. Such data is typically collected from different foundations and need to be handled in real time.

The NoSQL database is built based on multiple thoughts and methods to understand data demonstration and related query language. The advance of modern network technologies has made digital TV systems available throughout the world. To provide secure media delivery in digital TV systems, alarge number of messages are exchanged for key updates in the conventional key distributed schemes of Conditional Access Systems (CAS) (Varalakshmi and Uthariaraj, 2015).

Properties of big data: The term big data refers to the data which have unlimited quantity of datasets and is very complex to collect and store. Furthermore, the data must be efficiently examined and analysed using the traditional applications and approaches. The data comprises of 4 V's: Volume, Velocity, Variety and Variability. The typical size of data is considerably large, i.e. in the magnitude of petabyte's or exabyte's and is rapidly increasing every year (Volume). It is essential to provide a distributed fault tolerant data architecture that satisfies the following requirements: to store the high velocity streaming data set on database without losing any data from the streaming domain. As more people now have access to internet, the rate of data available also increases (Velocity). Chung *et al*. (2014) have proposed a scalable solution to handle large-scale data. The method is based on the MapReduce framework (He *et al*., 2008) that is attractive for treating massive unstructured data simultaneously in HBase (George, 2011). The multiple type of data from different sources need to be processed, however, the integrity and security of data need to be maintained. Figure 1 shows the data from different sources.

NoSQL architecture: As previously stated, in the NoSQL architecture there are four types of models used for data management. It includes document based databases, key value based database, column based databases and graph based databases. In the key value based database, data is stored using hash tables which have pointers and a unique key for specific type of data. In contrast, the document based databases employs key values approach and the data is stored in documents structure format. In column based databases the data is stored in cells,

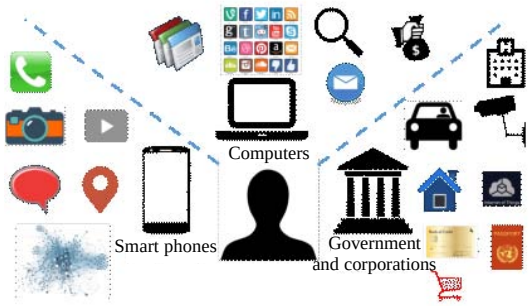

Fig. 1: Big data sources

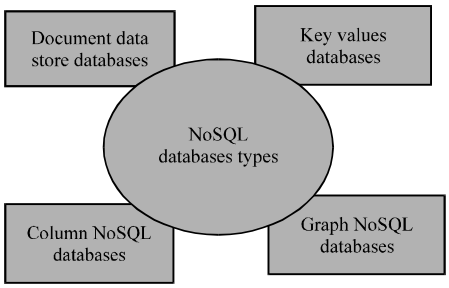

Fig. 2: Database types

grouped in columns instead of rows. A framework entitled Jack Hare with the query compiler SQL, JDBC driver and a systematic method (Lee *et al*., 2011). Jack Hare is responsible for storing data that are originally in RDBs without being aware of the NoSQL database. The data is distributed over the node and is arranged in a non-overlapped form called sharing. Such method exists in RDBMS systems but just recently has gained importance by the advancement in the NoSQL systems. Figure 2 shows four different storage architectures for NoSQL databases.

Conventionally, the database in RDBMS is constructed based on the relational data model. The RDBMS agrees to the description of data structures, storage and retrieval processes and reliability. In such databases, data is stored in tables which consist of rows and columns. To alleviate data retrieval issue, data in table is not repeated. For that reason, the RDBMS is dependent on primary keys where each row allocated with a unique row number. Ping *et al*. (2011) use NoSQL to replace relational databases which are applied in traditional information management systems. All studies carried out focuses on the improvement of maintenance the level of NoSQL which is more difficult to manage compared to relational databases (Ping *et al*., 2011). Lawrence, R presents techniques that allow NoSQL to query SQL and interact with systems using the JDBC driver via. a standardized generic architecture. The architecture allows

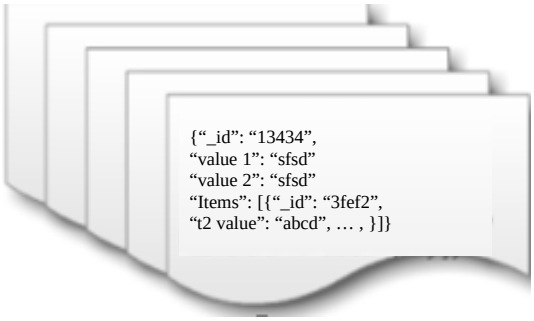

Fig. 3: NoSQL documenet oriented NOSQL schema

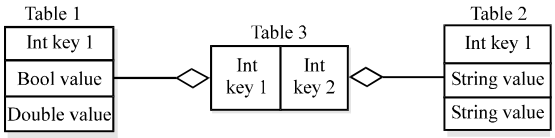

Fig. 4: Relational database schema

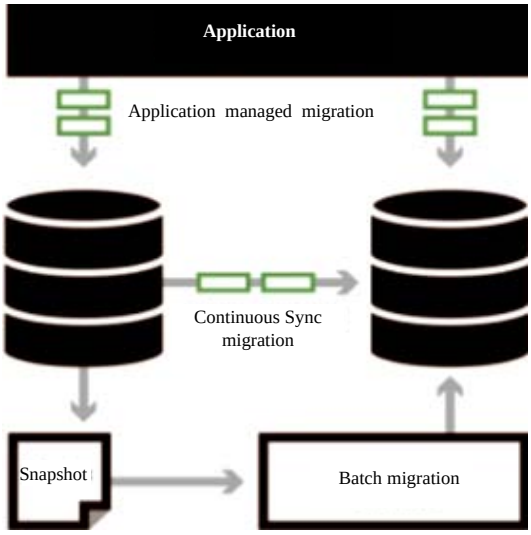

Fig. 5: Overview of a RDB migration procedure towards the NoSQL (MongoDB)

users to combine data from relational model and NoSQL Model in a single query, the combination is made from the SQL query translation to non-standard APIs NoSQL which automatically performs the unsupported operation by NoSQL System (Lawrence, 2014). The NoSQL provides a flexible schema for the data stored in row including the NULL value problem that persistently occur in the RDBMS. Using the NoSQL data models, only a single application is needed rather than multiple applications. A collection of distributed data which are organized into named columns is called data frames (Seshatheri and Bhuvaneswari, 2016). Optimization rules and data sources acts as a catalyst to speed up the evaluation of data frame API (Seshatheri and Bhuvaneswari, 2016).

Figure 3 shows the NoSQL document oriented database schema. Document oriented database is a designed for storing, retrieving and managing document oriented information and it relies on internal structure in the document in order to extract metadata that the database engine uses for further optimization. Figure 4 shows the relational database schema.

MapReduce in NoSQL: In RDBMS, the MapReduce functionality requires transferring data from one database to another database. The NoSQL system can alleviate such issue. The MapReduce operation is embedded in the NoSQL and data migration from one system to other may be avoided. MapReduce is a method which can process large amount of data with concurrent datasets. The data sets can be prepared for data mining by horizontally aggregating the data present in relational tables (Jasti and Vasumathi, 2013). The data typically lies within the pattern characterized by three steps: mapper, shard and reducer. The mapper part includes the map process which simultaneously process multiple parts of data and produces a couple, i.e. (key-value). Later, the shard ensures that the mapper process is completed. Upon completion of the map process, it sorts the (key-value) pair. Later, the reducer produces result by combining the values and increases the count after getting the separate (key-value) pairs (Kilicoglu *et al*., 2012).

Query language for NoSQL: In RDBMS, the queries mechanism is very complex. Mior (2014) offers solutions to optimize query performance using a guide mapping for the transformation, the approach is made from a data model to a given physical schema. The schema design is essential to improve the performance of the NoSQL database. The solution involves a query language and methods that traces the execution of requests.

Figure 5 shows the difference between data query operation in relational database and NoSQL database. A typical mechanism to query in RDBMS is by using SQL CLI while MongoDB CLI is the method to query in NoSQL databases.

Hashing in NoSQL: In NoSQL, the Hash trees are used, where each leaf node is tagged with the hash of its child node. The trees provide a protective method for the data transfer between servers to ensure reliability. Alomari *et al.* (2015) have proposed a data model based on a standardized API (Silva *et al*., 2013) for the SQL and NoSQL databases in the cloud appointed CD Port

framework (Alomari *et al*., 2015). The method applied hides potential change data storage model of the application layer (Harsh *et al*., 2012). For instance, if K represents the keys and n represents number of arrays, a stable hashing assures that in normal condition not more than Kin keys are changed to new arrays.

User of REST API in NoSQL: REST is a Hyper Text Transfer Protocol (HTTP) Application Program Interface (API). The API has four methods which are PUT, GET, POST and DELETE to perform multiple operations. Currently, several NoSQL tools are using the API to communicate with client using these API's. A Protobuf (Protocol buffer) is a technique used for effective serializing and transferring data. It is typically used for RPC (Remote Procedure Calls) as well as for storing data. On the other hand, Thrifts is a software framework and an Interface Definition Language (IDL) for cross-language services and API development (Chang *et al*., 2008).

RESULTS AND DISCUSSION

Migration of a RDBMS to a NoSQL DB (MongoDB): The migration of databases requires a connection between two types of DBMS to be able to affect a dialogue. The stage of migration is focused on a bridge that will act as a gateway that will connect the two DBMS which do not interact in the same way and differs in technology. A data model provides technological exchange between the relational model and the document-oriented model of the NoSQL. The model of data will be the intermediary between both DBMS which will aim at improving the contents/data, adding contextual information by tagging, categorization, classification of data and other sources of basic reference. Data migration approach originated from a RDB to NoSQL for adopting the technology of big data (Kilicoglu *et al*., 2012).

Data model: Our approach consists in structuring a generic method of semantic enrichment, representing the relational database in a form of flat data which will play the role of the core of the prototypes conceived for the migration. The model will be defined as follows:

$$\{C|C:=(Cn, carA, Rel, carB)\}$$

Where:

Cn = The name of the class
carA = Specifies the cardinality of the side of the starting class assigned with a reference to Cn
carB = Specifies the cardinality of the side of the end of class that makes a reference to the Rel
Rel = The class that interacts with our starting class
Cn = The class that enters into relation with Cn

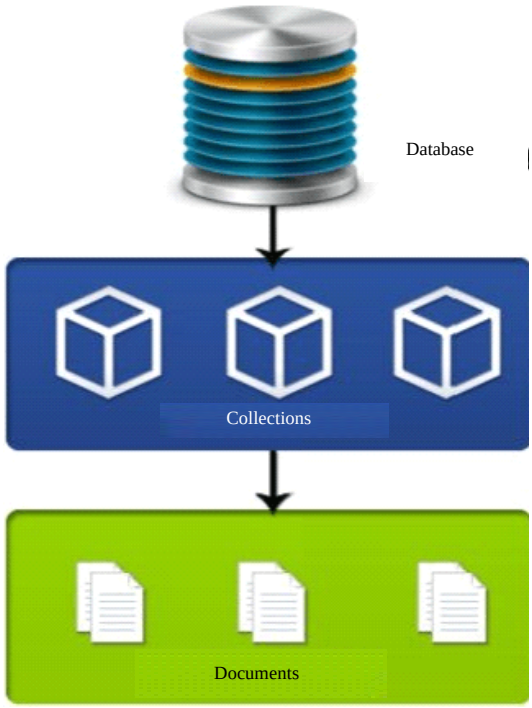

Fig. 6: The logical data structure of NOSQL-MongoDB

$$A = \text{attribute:} = \{a|a:=(an, tag)\}$$

Where:

An = The name of the attribute
Tag = Primary Key (PK)| Foreign Key (FK)| Unique Key (UK)

Figure 6 shows the logical structure of NoSQL mongoDB. A MongoDB server can contain multiple databases. A database is an independent container for data. Each database can contain multiple collections and a collection can contain multiple documents.

Data modelers frequently starts by looking at the logical constructs in a database. MongoDB's flexible document model makes it easy to create rich data models that reflect how data is used in application. The model schematizes the conception of the source model and the structure of the target physical schema. These parameters are obtained through responsible objects to query the database to obtain information about the physical structure of the database and container objects of the general information about the database (database Metadata, result set, result set Metadata).

Conversion technique: The migration of a RDB to a document-oriented database MongoDB will be without predetermined scheme in the form of document BSON.

Storage is in the form of collections that have a shared index. The collections will be like relational tables for each table extracted from the model.

Figure 5 shows the architecture of the prototype, showing the drivers used and the migration steps of the NoSQL database. The first step is the realization of the data model following a series of processing and capture of metadata (Alami and Bahaj, 2014). According to the model of data, we perform a selection method which takes the parameter C. Cn which realize a selection from the RDB in an automatic way (Alami and Bahaj, 2015). The last step is the creation of the physical schema NoSQL with data mapping and the necessary constraint. The creation syntax of MongoDB is defined by DB. createCollection (name, options):

- Name = Name of the collection to create
- Options = Optional

Configuration options for creating a capped collection for preallocating space in a new collection or for creating a view.

Transformation rule: We are going to propose few rules allowing to develop the determined structure to exploit the constraints formulated on the schema. The cardinalities of an entity in an association expresses the number of times an instance of this entity may be involved in a case of association, the minimum and maximum. Cardinalities reflect management rules. These rules are specific to the organization studied which are decided by managers and decision makers. These rules express constraints on the model. A Foreign key constraint defines a document or combination of documents, used to establish and maintain a connection between the two data collections. The creation of a Foreign key constraint is made when creating or editing a collection. In a Foreign key reference, creating a link between two collections is done when attribute the documents containing the primary key values of a collection or in the documents of the other collection. This document becomes a Foreign key in the second collection.

Creating an index for a Foreign key is usually useful for the following reasons. Foreign keydocuments are used in referencing criteria when the data of theassociate collection are combined in queries, through correspondence or documents of the Foreign key constraint of a collection with the unique key or primary key document for the other collection. Unique indexes to specify the primary key constraint.

Ensure index, a syntax of creation which can organize documents in a field and guarantee the uniqueness of a constraint.

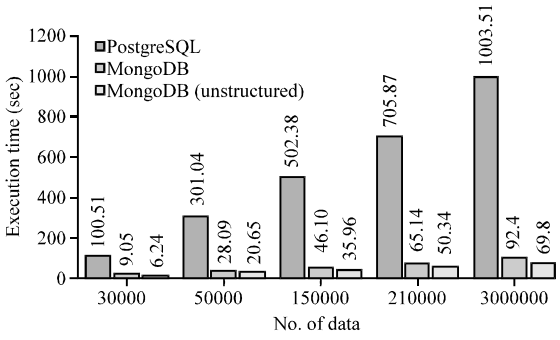

Fig. 7: Chart of insert operation speed

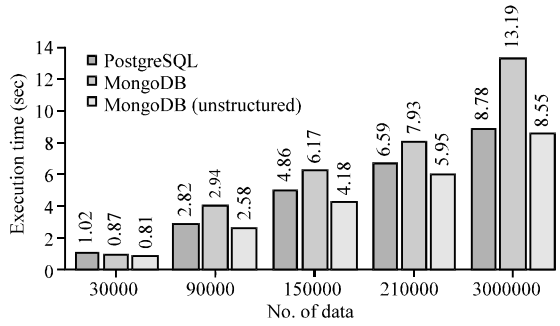

Fig. 8: Chart of select operation speed

We must distinguish several cases. Knowing that a linear relationship of the type (1.1)-(1.1) should not exist, we have the two following cases:

$$\text{Binary relation } (0.1)\text{-}(1.1)$$

We duplicates the key of the class 1 based on the entity to cardinality (0.1) in the class 2 based on the entity to cardinality (1.1). Example:

Class 1:
- PK1
- Attr1

Class 2:
- PK2
- Pk1(pk)
- Attr2

The PK1 which is the Primary Key of the class 1 becomes Foreign key in the class 2. Speed of select operation has been measured by time taken from selection of data to processing of inserted data to be available. shows chart of select operation speed. Figure 7 shows the comparison of the insert operation speed, followed by Fig. 8 which shows comparison of the select

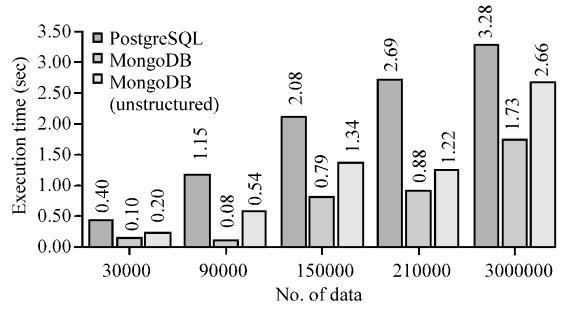

Fig. 9: Chart of update operation speed

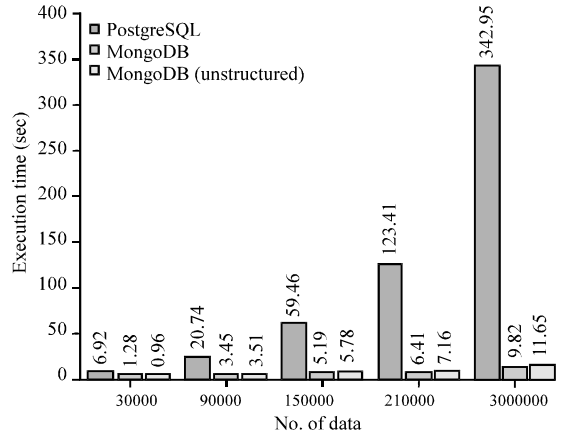

Fig. 10: The chart of delete operation speed

operation speed. Figure 9 shows speed of update operation and Fig.10 shows comparison of delete speed operation.

Some migration approach of the relational model towards the document oriented model. Proposes a migration approach to MangoDB, starting with a method for extracting of a data model from a RDB via. metadata. Then proceed with schema and data migration according to the data model, preserving the integrity constraints. Migration remedy the problems related to data redundancy, eliminate joins and harnesses the power of relational databases which consists to associate collections by combining data via. Foreign key (ensure index and unique indexes). There are many applications for NoSQL and each application has distinctive architecture and scheme. Each application has a unique data storage implementation method, NoSQL began as open source with small vendors. To date, the size of data in NoSQL has substantially increases and as such attracts new customers in the market. To better understand the NoSQL databases, four categories of NoSQL databases have been discussed.

CONCLUSION

Through this study, performance comparison of insert, select, update and delete operations on data has been conducted to compare RDBMS and MongoDB (NoSQL). Insert, select, update and delete operation speed of MongoDB was faster than that of RDBMS in general. For performance improvement of MongoDB, designing with unstructured data model seems to be better than designing with relational data model. Moreover, we could notice that the select operation of RDBMS could be improved by using index.

In conclusion, using MongoDB with unstructured data model will provide overall performance improvement. However, when the environment requires precise and structured data model, using RDBMS such as PostgreSQL will exhibit higher quality of performance. By applying the comparison results of this study to the DBMS design when implementing NoSQL, we can expect for better performance maintenance. We also expect that further studies on performance of distributed processing by RDBMS and NoSQL will contribute to advancement of database technology in big data environment.

ACKNOWLEDGEMENT

This research is done under the guidance of staff member of Department of Computer Applications, Vels University, Chennai. Thanks to our guide Varalakshmi for valued guidance to complete this journal.

REFERENCES

Alami, A.E. and M. Bahaj, 2015. Schema and data migration of a Relational Database RDB to the extensible markup language XML. World Acad. Sci. Eng. Technol. Intl. J. Comput. Electr. Autom. Control Inf. Eng., 9: 1763-1768.

Alami, E.A. and M. Bahaj, 2014. The migration of a Conceptual Object Model COM (Conceptual Data Model CDM, Unified Modeling Language UML class diagram...) to the Object Relational Database ORDB. MAGNT. Res. Rep., 2: 318-327.

Alomari, E., A. Barnawi and S. Sakr, 2015. CDport: A portability framework for no SQL data stores. Arabian J. Sci. Eng., 40: 2531-2553.

Chang, F., J. Dean, S. Ghemawat, W.C. Hsieh and D.A. Wallach *et al.*, 2008. Bigtable: A distributed storage system for structured data. ACM. Trans. Comput. Syst., 26: 1-26.

Chung, W.C., H.P. Lin, S.C. Chen, M.F. Jiang and Y.C. Chung, 2014. JackHare: A framework for SQL to NoSQL translation using MapReduce. Autom. Software Eng., 21: 489-508.

George, L., 2011. HBase: The Definitive Guide. O'Reilly Media, Boston, Massachusetts, ISBN:978-1-449-39610-7, Pages: 501.

Harsh, P., F. Dudouet, R.G. Cascella, Y. Jegou and C. Morin, 2012. Using open standards for interoperability issues, solutions and challenges facing cloud computing. Proceedings of the 8th International Joint Conference and Workshop on Network and Service Management (CNSM) and Systems Virtualiztion Management (SVM), October 22-26, 2012, IEEE, Las Vegas, Nevada, ISBN:978-1-4673-3134-0, pp: 435-440.

He, B., W. Fang, Q. Luo, N.K. Govindaraju and T. Wang, 2008. Mars: A MapReduce framework on graphics processors. Proceedings of the 17th International Conference on Parallel Architectures and Compilation Techniques, October 25-29, 2008, ACM, Toronto, Ontario, ISBN:978-1-60558-282-5, pp: 260-269.

Jasti, S. and D. Vasumathi, 2013. Creating minimized data sets by using horizontal aggregations in SQL for data mining analysis. Intl. J. Adv. Trends Comput. Sci. Eng., 2: 32-37.

Kilicoglu, H., D. Shin, M. Fiszman, G. Rosemblat and T.C. Rindflesch, 2012. SemMedDB: A PubMed-scale repository of biomedical semantic predications. Bioinf., 28: 3158-3160.

Lawrence, R., 2014. Integration and virtualization of relational SQL and NoSQL systems including MySQL and MongoDB. Proceedings of the International Conference on Computational Science and Computational Intelligence (CSCI) Vol. 1, March 10-13, 2014, IEEE, Las Vegas, Nevada, ISBN:978-1-4799-3011-1, pp: 285-290.

Lee, R., T. Luo, Y. Huai, F. Wang and Y. He *et al*., 2011. Ysmart: Yet another SQL-to-map reduce translator. Proceedings of the 31st International Conference on Distributed Computing Systems (ICDCS), June 20-24, 2011, IEEE, Minneapolis, Minnesota, ISBN:978-1-61284-384-1, pp: 25-36.

Mior, M.J., 2014. Automated schema design for NoSQL databases. Proceedings of the Symposium on SIGMOD PhD, June 22-22, 2014, ACM, Snowbird, Utah, ISBN:978-1-4503-2924-8, pp: 41-45.

Ping, Z.W., L.I.M. Xin and C. Huan, 2011. Using MongoDB to implement textbook management system instead of MySQL. Proceedings of the IEEE 3rd International Conference on Communication Software and Networks (ICCSN), May 27-29, 2011, IEEE, Xi'an, China, ISBN:978-1-61284-485-5, pp: 303-305.

Seshatheri, E. and T. Bhuvaneswari, 2016. An efficient distributed data processing method for smooth environment. J. Eng. Appl. Sci., 11: 1855-1858.

Silva, L.A.B., C. Costa and J.L. Oliveira, 2013. A common API for delivering services over multi-vendor cloud resources. J. Syst. Software, 86: 2309-2317.

Varalakshmi, R. and V.R. Uthariaraj, 2015. Huffman based conditional access system for key distribution in digital TV multicast. Multimedia Tools Appl., 74: 2899-2912.