Perspectives

Perspectives on a Big Data Application: What Database Engineers and IT Students Need to Know

Emre Erturk
Senior Lecturer, School of Computing
Eastern Institute of Technology
Napier, New Zealand
eerturk@eit.ac.nz

Kamal Jyoti
Postgraduate Student, School of Computing
Eastern Institute of Technology
Napier, New Zealand
erkamaljyoti@gmail.com

***Abstract*— Cloud Computing and Big Data are important and related current trends in the world of information technology. They will have significant impact on the curricula of computer engineering and information systems at universities and higher education institutions. Learning about big data is useful for both working database professionals and students, in accordance with the increase in jobs requiring these skills. It is also important to address a broad gamut of database engineering skills, i.e. database design, installation, and operation. Therefore the authors have investigated MongoDB, a popular application, both from the perspective of industry retraining for database specialists and for teaching. This paper demonstrates some practical activities that can be done by students at the Eastern Institute of Technology New Zealand. In addition to testing and preparing new content for future students, this paper contributes to the very recent and emerging academic literature in this area. This paper concludes with general recommendations for IT educators, database engineers, and other IT professionals.**

Keywords-information technology; database design; big data; MongoDB

I. INTRODUCTION

This paper discusses the latest emerging database technology and concepts. The content is aimed at information technology and cloud computing students as well as working database designers and administrators surveying professional and scholarly literature, and considering retraining. In order to demonstrate certain aspects of big data, it is first necessary to briefly review the qualities of modern cloud based databases. This also involves revisiting some of the fundamentals of database technologies. Providing an instructional manual is not the purpose of this paper. The aim is to demonstrate the primary activities and concepts that current IT students and database engineers need to know in order to work professionally with big data applications. Then the paper relates the essential learning objectives and their relevance for industry practice and to the IT sector in general. The following activities can be done in the Database Management Systems and Database Administration courses. Students in these two courses are not required to do extensive programming or create new applications. Prior to the demonstrations in section IV, this paper covers the fundamentals of big data and MongoDB in sections II and III with a literature review. Finally, the paper summarizes and reflects on the authors' technical experimentation along with conclusions and recommendations in section V.

II. CLOUD COMPUTING AND THE IMPACT ON BIG DATA

Cloud Computing services operate on shared and remote resources on the internet rather than on an organization's own local servers or on the end users' own personal computers or devices [1-2]. As a result, cloud based services achieve greater availability, flexibility, and scalability. A wide range of platforms and applications are currently delivered under the banner of cloud computing.

Data management is an important issue in cloud computing as millions of people use cloud based hardware and software services that constantly store, update, and retrieve a great amount of data. Big Data is the term used to describe massive volumes of both structured and unstructured data [3]. It is difficult and inefficient to process this amount and type of data using traditional database applications [4]. One example of big data comes from the constantly expanding social media platforms and their users. Another example is from the increasingly complex health care industry offering new devices and web based applications for the customers. All of this data is hosted on remote servers on the cloud. Managing and running large and 'live' databases on the internet involves many technical aspects such as virtualization, concurrency control, operating systems, network administration, process scheduling, load balancing, transaction management, and database design.

III. UNDERSTANDING BIG DATA APPLICATIONS

In order to manage big data, many NoSQL (Not only SQL) databases have been introduced in recent years. These NoSQL databases handle data in ways different from the tables and

structured query language (SQL) statements of traditional relational databases. There are many NoSQL document oriented databases, for example, CouchDB and MongoDB. Document oriented databases store and retrieve data according to the meta-data definitions and tags found in their documents.

MongoDB was first developed by in 2009. It has versions that are compatible with different editions of Windows, Linux, Solaris, and Mac operating systems. MongoDB is an open source and free database application, under the GNU Affero General Public License. It is currently the most popular application in the category of document oriented databases [5].

MongoDB can store semi-structured and polymorphic data as well as structured data. Much of the current big data exchanged on the internet does not follow clear cut structures, rigid schemas, or restrictions in terms of data type and length. This also means handling emails, forums, complicated large objects, and multimedia files [6]. MongoDB allows users to flexibly define arrays within documents, and perform various operations on those array fields. Furthermore, MongoDB offers database querying functionality and supports high performance indexing. These are useful, for example, for text searches and manipulation of geospatial data. Instead of SQL Join statements, MongoDB users may reference a document from another or embed a document inside another document. It also features a powerful data aggregation and data analysis framework, for performing calculations on large data sets.

MongoDB aims to provide high performance, availability, and scalability, for cloud based information systems and big data environments. In order to ensure availability, sophisticated database systems create replicas of database files in different locations so that, if one location is down, then the user can access the information from another location. As shown in Figure 1, a new machine becomes the primary server while the 'heartbeat' diagnostic program continues to check if the various replica servers are online and working properly. Additionally, it is possible to automate the failover (switching from a server that is down to an available one) and data recovery processes.

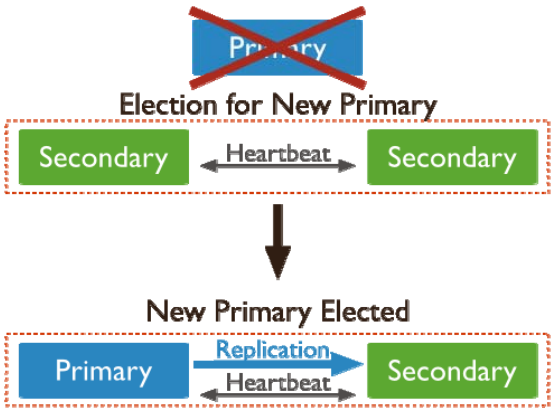

Fig. 1. Database Replication and Failover.

Scalability is the ability of databases to grow tremendously in size while maintaining usability or performance. There are very large databases that constantly accumulate data on the internet. In MongoDB, more servers can be allocated to a database to balance the workload and increase performance; this is called horizontal scalability. Sharding is the process in which a single database is divided into multiple components. These database components are stored on different servers. These servers may also be virtual machines or cloud based.

IV. LEARNING ACTIVITIES USING MONGODB

One of the first skills that the database engineer needs is to be able to quickly install the MongoDB server software on a given computer. A brief example in this paper is installing it on the Microsoft Windows operating system. After navigating to https://www.mongodb.org/downloads, the user needs to select the appropriate version from the menu, and clicks on 'Download MSI.' Once the setup file is saved, the user can execute this, follow the instructions, and complete the steps. After this installation, the MongoDB server can be run from the Windows command prompt. However, the setup process does not complete all the requirements; the user needs to manually create a subfolder named 'db' in the 'data' folder of the hard drive to avoid potential error messages and run MongoDB.

Although a database engineer can interact with this basic installation using the command line interface, the next recommended and useful task is to install a front-end graphical user interface application to manage the databases. Figure 2 shows the configuration of the front-end application called NoSQL Manager. The default port number for the MongoDB server is 270017.

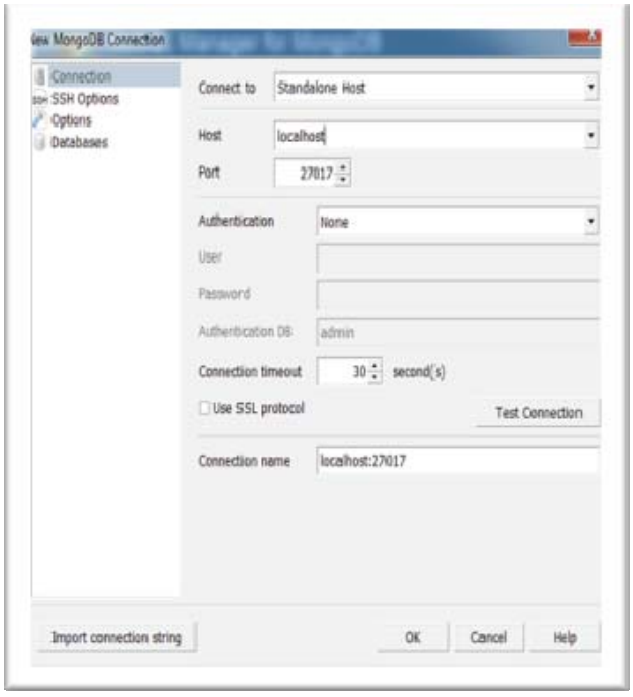

Fig. 2. Connecting to MongoDB using a front-end application.

Another optional tool for database students and trainees to be aware of is called MongoDB Management Service (MMS).

This is a cloud based tool, and does not require an installation. MMS offers a sophisticated database backup service, and can monitor up to thousands of online database deployments [7].

The second important area for learning MongoDB involves the storage concepts and terminology. Most trainees will already be familiar with relational database concepts and terms; therefore a good approach would be to explain to them the corresponding MongoDB terms along with the similarities and differences. As seen in Figures 3 and 4, a MongoDB database is made up of collections, which may be viewed as two dimensional tables. On the other hand, given a similar business domain, relational database structures necessitate the creation of multiple tables to represent the data whereas the same design can be done with fewer collections in MongoDB. Not having to create joining tables for many-to-many relationships and handling one-to-many relationships within the same collection with are examples of reasons for this. Each MongoDB document is similar to an SQL row; however MongoDB's BSON (binary java script object notation) document format provides recursive functionality and allows more efficient database scanning.

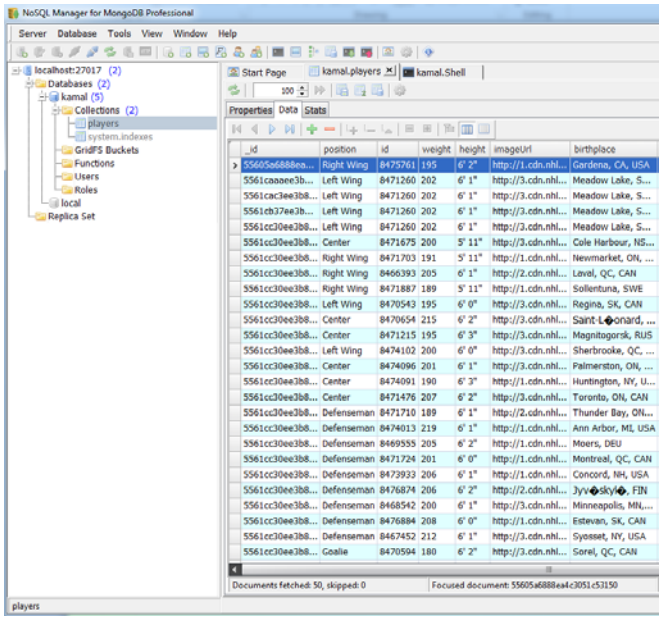

Fig. 3. Viewing sample data from MongoDB.

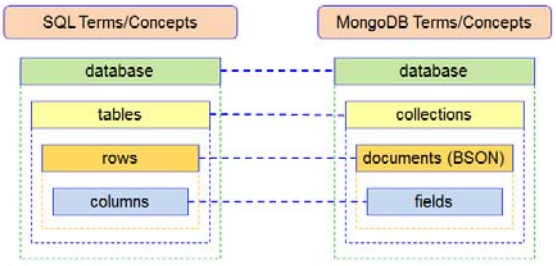

Fig. 4. MongoDB structure vs traditional SQL.

Learning the syntax is essential for creating a MongoDB database. Advanced students and database professionals are already proficient in programming with SQL using relational database applications, e.g. Microsoft SQL Server or MySQL. Figures 5 and 6 compare statements in SQL versus MongoDB.

In the case of an INSERT command that creates new data records, the mapping of the values is easier to understand. It can be explained as a transposition from SQL's horizontal coding (i.e. items and values are listed from left to right) to a vertical coding in MongoDB where the document items and values are listed from top to bottom. In SQL, the input row names are grouped together while their input values are also grouped together separately; these groups are implicitly corresponded based on their order within that table's structure. In MongoDB, the input values are written next to the field name, similar to XML (Extensible Markup Language), and these fields can be listed vertically, each one on a new line. In the case of a READ command (Figure 6) that fetches data records, the difference is greater, between traditional SQL SELECT statements and MongoDB (which uses the FIND method). First, the same kind of transposition or projection difference applies, going from horizontal groupings to vertical listings. On the other hand, in MongoDB, the user needs to be careful to separate the fields involved in the query criteria from the fields included in the query output. Furthermore, SQL queries use intuitive Boolean operators to define query criteria while MongoDB queries use abbreviated tags, such as $gt (greater than). The value of '1' also needs to be specified next to the field name(s) to show them in the output result.

The publicly available instructional information on coding with MongoDB is limited, with many of the resources written in a technical language that is not conducive to easier and faster learning. For this reason, the explanations offered above will be useful in a future training course by filling an educational gap.

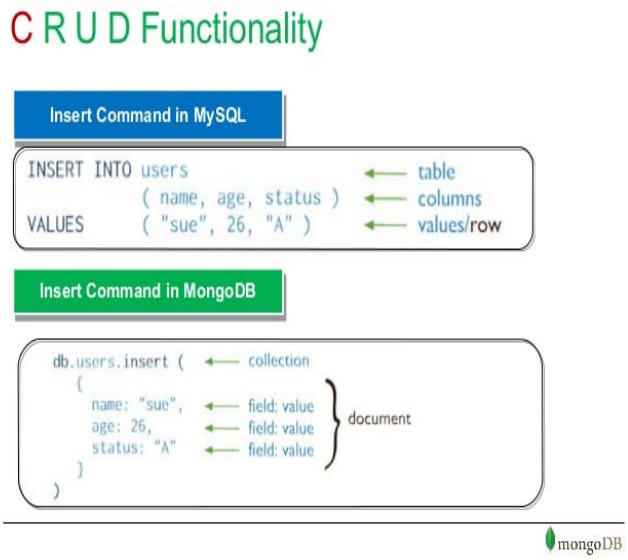

Fig. 5. NoSQL vs traditional SQL syntax.

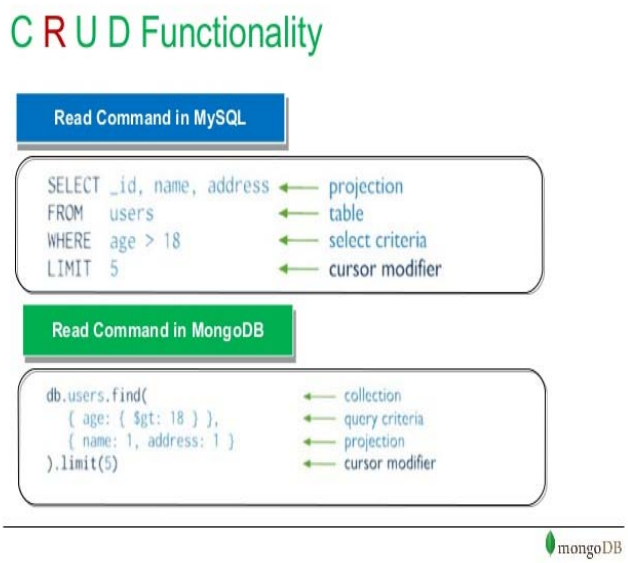

Fig. 6. NoSQL vs traditional SQL syntax.

V. CONCLUSION

In order to implement these learning activities for students, it is necessary for instructors to cover the theoretical content in depth prior to the practical work. Furthermore, the practical tutorials require initial familiarization, trouble-shooting, and self-guided technical learning. Therefore it is important that instructors plan and allocate time to students so that they can understand what they need to do, and become prepared. In later stages of the course, once MongoDB is installed on a server, students can access a cloud based MongoDB using the lab computers and their own computers. In the course (that this paper targets), about thirty enrollments are expected. These students will be guided by one instructor.

A recommendation for further research as well as training material development should be on understanding and improving the security of a MongoDB database. This is important because many MongoDB implementations have been done recently without setting up the correct access controls [8]. After the initial installation, all of the user profiles and authorizations have to be set up by the database administrator. Secondly, hackers may try to exploit MongoDB databases using code injections similar to the way SQL injections have been used to target traditional relational database applications. Future students who are doing additional training on MongoDB [10] security should review the syntax of the possible injections, and learn how these injections can be prevented from posing a risk.

It needs to be emphasized that cloud computing and big data offer additional benefits together: capturing and analyzing much more information, making better business decisions, using the IT infrastructure more efficiently, and enhancing disaster recovery. However, when selecting a database product, it needs to be kept in mind that each company should decide according to its own unique data and reporting requirements. However, in general, capturing and providing fast access to big data will continue to be crucial, with ever increasing numbers of connected mobile applications and new online systems [9].

As emphasized in this paper, the public awareness of big data applications and their benefits is increasing. As a result, more companies are looking for IT professionals with the latest knowledge and skills in this area. These potential employers are not only software development companies but also include diverse organizations and companies from different sectors of the economy.

REFERENCES

[1] V. Beal, "Cloud Cmputing (the Cloud)", http://www.webopedia.com/TERM/C/cloud_computing.html

[2] G. A. Tarnavsky, E. V. Vorozhtsov, "Cloud Computing in Science and Engineering and the "SciShop.ru" Computer Simulation Center", Engineering, Technology & Applied Science Research, Vol. 1, No. 6, pp. 133-138, 2011

[3] V. Inukollu , S. Arsi1, S. Ravuri, "High Level View of Cloud Security : Issues and Solutions", Fourth International Conference on Computer Science, Engineering and Applications, Chennai, India, pp. 51-61, 2014

[4] A. Shields, "Why Traditional Database Systems Fail to Support Big Data", http://marketrealist.com/2014/07/traditional-database-systems-fail-support-big-data/

[5] Solid IT, "DB-Engines Ranking of Document Stores", http://db-engines.com/en/ranking/document+store

[6] S. Khan, V. Mane, "SQL Support over MongoDB using Metadata", International Journal of Scientific and Research Publications, Vol. 3, No. 10, pp. 1-5, 2013

[7] J. Roy, "MongoDB: Characteristics and future", http://www.mongodbspain.com/en/2014/08/17/mongodb-characteristics-future

[8] J. Heyens, K. Greshake, E. Petryka, "MongoDB Databases at Risk: Several Thousand MongoDBs without Access Control on the Internet", Saarland University Center for IT Security, Privacy, and Accountability, https://cispa.saarland/wp-content/uploads/2015/02/MongoDB_documentation.pdf

[9] E. Erturk, "An Intelligent and Object-oriented Blueprint for a Mobile Learning Institute Information System", International Journal for Infonomics, Vol. 6, No. 3/4, pp. 736-743, 2013