ECE656 Project Proposal: Document-Oriented Database

Hai Jiang, Xiyue Zhang, Chang Liu

The most common database implementation today is based on the relational model [1] which uses SQL as its query language. However, NoSQL (Not Only SQL) database solutions [2][3] are becoming more prominent as massive amounts of rapidly growing data are being collected today. More specifically, data in the relational model is usually represented by a database schema [1], in order to capture the semantics of the database. Objects in the database with the same number of characteristics, type and format are grouped together, making it structured data. In comparison, NoSQL can help deal with data that is not structured. Data can be semi-structured, such that similar data objects can be grouped together, but the objects may have different characteristics. The paper presented by [5] demonstrates that MongoDB (NoSQL) performs equally as well or better than the relational database, except when aggregate functions are utilized. Among the NoSQL database concept, document-oriented databases are inherently a subclass of the [key-value store](https://en.wikipedia.org/wiki/Key-value_store), and becoming popular in the current implementation. A document-oriented system relies on internal structure in the document order to extract [metadata](https://en.wikipedia.org/wiki/Metadata) that the database engine uses for further optimization [6].

Our project is going to study, research and implement on the typical NoSQL database: Document-oriented database. We will practice on the leading Document Database Software: MongoDB. More specifically, we will study, research, and implement on Mongo DB to finish the following tasks:

1. To select document data set, and achieve storing, retrieving, and managing document-oriented information on MongoDB.
2. Seek to improve the performance of the Document Database by combining Relational and Semi-structured Databases for a specific application. Our idea can be summarized as follows:

* Problems:

As the huge amounts of data are collected, several types of non-relational data co-exist with the relational ones. In most projects there is either no time for developing a connection driver or it is unsafe to rely on community projects. Also, there is no way to predict the exact amount and structure of data. Only a rough estimate of the structure is known: A list of key-value-pairs. There are parts of the data model which do require ACID transactions and strong consistency as well as parts which do not require this functionality [7].

* Planned Solution:

Based on the specific data set we found, we will study and implement an algorithm to determine which data type is suitable for which kind of data base. Accordingly, we will split the original data into different formats which are suitable for different data base storage. For example, some data are stored in MongoDB, some are stored in MySQL. In this case, a data set containing different data sub-types can be managed in different ways. We will experiment on the method of splitting and managing on different databases, and testing for the efficiency in response to different query language compared to the experiment on all the dataset storing in only one database.

1. Seek to optimize the searching operation of Document Database.

As a result of the initial literature review, several methods are proposed to improve the performance of the search function of the document database. In the patent presented by [8], the performance of searching a document database is achieved according to a representative semantic space. In the patent presented by [9], the operation of searching a document is optimized by indexing each document containing nested fields. In the patent presented by [10], a search plan for a hierarchical structure possessed by a searched document in response to the search request, thereby effectively using information concerning structured document database. The patent [11] proposes a method for re-ranking documents retrieved from a document database, aiming at shorten the time of searching.

We would like to select one or two methods from the above literatures to practice on the MongoDB database system. Some necessary coding script and programming implementations are included. We will also compare the search performance of the document database before and after incorporating our method.

References

[1] Elmasri, R. and Navathe, S. 2011. Fundamentals of Database Systems, Addison-Wesley.

[2] Strauch, C. NoSQL Databases. Selected Topics on Software-Technology, Stuttgart Media University.

[3] NoSQL. http://nosql-database.org/

[4] MongoDB. http://www.mongodb.org/

[5] Parker, Zachary, Scott Poe, and Susan V. Vrbsky. "Comparing nosql mongodb to an sql db." Proceedings of the 51st ACM Southeast Conference. ACM, 2013.

[6] Document-oriented Database. <https://en.wikipedia.org/wiki/Document-oriented_database>

[7] Ebel, Marius, and Martin Hulin. "Combining relational and semi-structured databases for an inquiry application." Multidisciplinary Research and Practice for Information Systems. Springer Berlin Heidelberg, 2012. 73-84.

[8] Sommer, Matthew S., and Kevin B. Thompson. "Method and system for optimally searching a document database using a representative semantic space." U.S. Patent No. 6,847,966. 25 Jan. 2005.

[9] Burrows, Michael. "System and method for search, index, parsing document database including subject document having nested fields associated start and end meta words where each meta word identify location and nesting level." U.S. Patent No. 6,963,869. 8 Nov. 2005.

[10] Hattori, Masakazu, Katsuhiko Nonomura, and Takuya Kanawa. "Structured document search method, structured document search apparatus and structured document search system." U.S. Patent No. 7,054,854. 30 May 2006.

[11] Knepper, Margaret M., Kevin Lee Fox, and Ophir Frieder. "Method for re-ranking documents retrieved from a document database." U.S. Patent No. 7,801,887. 21 Sep. 2010.