Amaninder Singh

Research Paper

IT 328

Winter 2015

**Document Store: Mongo DB and Couch DB**

This paper is going to discuss and explore about what is document store and what are some of the NoSQL databases related to that. It is also going to discuss how this model works. This paper will also explore what are the some strengths and weaknesses of the model as compared to relational database mode. Mainly in this paper, two types of NoSQL databases will be discussed: MongoDB and CouchDB.

According to Wikipedia, “a document store or document-oriented database is a computer program designed for storing, retrieving and managing document-oriented information.” Document store databases are one of the main classifications of NoSQL databases. These are designed around an intangible idea of a document rather than storing data that differs in radical ways from the means available to traditional relational SQL databases.

This type of data models is flexible in the notion of document structure. Records do not need to have a uniform structure, i.e. different records can have different columns. For example, records can have a nested structure and columns can have more than value i.e. arrays. The types of the values of individual columns can be different for each record. One doesn’t have to define your schema ahead of time, which makes this a schema free database unlike relational database.

In 1970s, when relational databases were introduced, data schemas and data complexity was fairly simple and straightforward. “It made sense to conceive objects as sets of relationships. As the relationships between different type of data were specified in the database schema, these relational databases could be queried with a standard Structured Query Language, or SQL”(MongoDB, 2015).

But everything in the computing world has changed since the development of SQL databases. Cloud computing, web application and agile development methods are raising at an astounding rate every day. As these technologies getting more complex, their data storage necessities incline to grow exponentially and databases play an important role in the way web applications store their information.

Relational databases are great pieces of software that allow records to be stored and retrieved in table structures. Sadly, that is also the limitations of relational database. Tables can’t grow horizontally i.e. column wise because of limited storage space for row to grow. Another drawback of the this kind of database is that in order to find a record, system would have to iterate through the complete file to the find the record which makes it overwhelmingly slow. As Matei (2013) from San Jose State University in a research paper said: “Taking the data and splitting it up into multiple tables or normalizing solves the issue, however, this means data is now scattered around the disk and requires multiple reads in different sectors of the drive to retrieve the information.” This makes the problem more problematic as the data is not longer located at one spot and it takes longer to run the results.

MongoDB/ CouchDB are ways to solve it even though there is no easy or straightforward answer for the problem. There is no single database that solves the entire problem. It all depends on project to project. However, both of these databases have both strengths and weaknesses but overall it perform quite well and it doesn’t have as many restriction and limitations as other NoSQL databases.

MongoDB is one of the leading NoSQL open-source document database written in C++. MongoDB name comes from “hu**mongo**us”. Its model makes it easy to build on as it supports unstructured data natively and doesn’t require costly and time-consuming migration when application requirements change (MongoDB 2015). Another quality of MongoDB as Matei (2013) from San Jose State University in a research paper said:

MongoDB is a document store database that does not have strict schemas that RDBMs require and can grow horizontally without performance degradation. MongoDB brings possibilities for different storage scenarios and allow the programmers to use the database as a storage that fits their needs, not the other way around. Scaling MongoDB horizontally requires tens to hundreds of servers, making it very difficult to afford this kind of setup on dedicated hardware. By moving the database into the cloud, this opens up a possibility for low cost virtual machine instances at reasonable prices.

“Mongo DB’s documents are encoded in a JSON-like format, called BSON, which makes storage easy, is a natural fit for modern object-oriented programming methodologies, and is also lightweight, fast and traversable. In addition, MongoDB supports rich queries and full indexes, distinguishing it from other document databases that make complex queries difficult or require a separate server layer to enable them”(MongoDB, 2015). Users perform queries against the database using procedural API calls, rather than using declarative language like SQL. Basic idea behind the design of MongoDB is to make scaling easier and to make the notion of a row with a more adjustable, the document. MongoDB supports BSON (binary encoded JSON) data structures to store complex data types. Instead of stored procedures, developers can store and use JavaScript functions and values on the serve side. A document is a set of fields and can be thought of as a row in a collection. It can contain complex structures such as lists or even document. Each document has an ID field, which is used as a primary key and each collection can contain any kind of document, but queries and indexes can only be made against one collection.

According to Wikipedia, “Apache CouchDB, commonly referred to as CouchDB, is also an open source database that focuses on ease of use and on being a database that completely embraces the web.” As mention before, this is also a NoSQL database that uses JSON to store data. Also mention in Wikipedia that CouchDB does not store data and relationships in tables. Instead, each database is a collection of independent document. CouchDB is designed for offline operation: it uses multi-master asynchronous replication. In CouchDB, multiple replicas can have their own copies of the same data. “CouchDB comes with a suite of features, such as on-the-fly document transformation and real-time change notifications, that makes web app development a breeze. It even comes with an easy to use web administration console. It is highly available and partition tolerant, but it also eventually consistent. CouchDB has a fault-tolerant storage engine that puts the safety of your data first” (CouchDB, 2015).

As the finance industry is growing at high speed, which led to the major strain on existing IT infrastructure mainly in Databases. That is one real world application where document type databases are very useful. As MongoDB website says, “New finance industry regulation has meant increased volume, velocity and variability of data. This coupled with cost pressures from the business has led these institutions to seek alternatives. In this session learn how FS companies are using MongoDB to solve their problems. The use cases are specific to FS but the patterns of usage - agility, scale, and global distribution - will be applicable across many industries.”

Major disadvantage of this model over relational database is the non-existence of Joins. This means that when you need this type of functionality, you need to make multiple queries and join the data manually, which can get ugly and time consuming. Another major disadvantage is that document type databases tend to use more memory because of the data structures they are using.

**Reference:**

Apache CouchDB™ is a database. (n.d.). Retrieved February 4, 2015, from http://couchdb.apache.org/

CouchDB. (n.d.). Retrieved February 4, 2015, from http://en.wikipedia.org/wiki/CouchDB

Document-oriented database. (n.d.). Retrieved February 4, 2015, from http://en.wikipedia.org/wiki/Document-oriented\_database

Document Databases. (n.d.). Retrieved February 4, 2015, from http://www.mongodb.com/document-databases

Khan, S., & Mane, P. (2013, October 1). SQL Support over MongoDB using Metadata. Retrieved February 4, 2015, from http://www.ijsrp.org/research-paper-1013/ijsrp-p2255.pdf

Matei, T. (2013, May 1). MongoDB Performance In the Cloud. Retrieved February 4, 2015, fromhttp://scholarworks.sjsu.edu/cgi/viewcontent.cgi?article=1300&context=etd\_projects

Rahein, A. (n.d.). That No SQL thing- Document Database. Retrieved February 4, 2015, from http://ayende.com/blog/4459/that-no-sql-thing-document-databases