NOSQL Databases and Big Data Storage Systems

A short summary by Platon Karageorgis

The following summary focuses on the 24th chapter of the book Fundamentals of Database Systems, written by R. Elmasri and S. Navathe. The standard relational database system is broadly used and is probably the most common one in the modern era. The urge for more data though, has led many organizations to implement a different model that will be able to handle more data in a more efficient and especially faster way. The model is called NOSQL, and it means “No SQL” because its alternate design makes it difficult to implement a high level language like SQL.

A NOSQL database system is designed to maximize availability and therefore methods like replication are quite significant, with the master-slave and master-master model having the lead role. Moreover, the system must be scalable and in most cases prefers horizontal scalability instead of the vertical one, since it allows the increase of nodes and therefore handles well growing numbers of data. Sharding or horizontal partitioning, are also important aspects of a NOSQL model contributing in load balancing, while high performance is also a primary task and it is fulfilled mostly by the methods of hashing or range partitioning. In addition, NOSQL database systems have some additional characteristics that distinguish them from the common RDBMS. Firstly, they do not demand a schema in order to formulate the data, secondly as mentioned before they do not need high level languages like SQL, although they have implemented some that are quite high level. Finally, they use versioning as they choose to have multiple copies of the same data but with different versions of them.

The NOSQL database systems are more and more common nowadays, but they seem to have differences between them. There are four main types of them which will be introduced shortly. To begin with, the document based system which chooses to save data as a document -usually a frequently used format- followed by the key-value stores system, which concentrates on quick data access by assigning every value to a different key. Then there is the column-based system that partitions a table vertically and creates column groups, and lastly the graph-based system that depicts data as graphs. These models are analyzed more thoroughly along with their market implementations later in this text.

Before proceeding to the extended analyses of the NOSQL database versions, the writer discusses the CAP theorem. CAP stands for consistency, availability and partition tolerance and constitute an alter ego of ACID in database systems, but focusing on NOSQL systems. Consistency has a different meaning compared to ACID, as it means that data should have the same value in every copy they have in the database. In addition, the writer implies that guaranteed consistency is arguably a poor choice, as the CAP system states that it is impossible to have all those aspects applied together, and the one that is usually left behind is consistency. Hence, eventual consistency is an important term in these systems. This term means that the system will be consistent, but in a weaker level. Furthermore, availability means that read and write operations should be executed immediately, without latency or error messages, because the data might not be accessible. Lastly, partition tolerance takes care of the welfare of the system when there is a network failure, and makes sure that it will continue to run.

The writer proceeds to the analyses of the first NOSQL database type, the document-based one. This type stores data as documents like JSON files and holds the NOSQL database main attribute discussed before, as it does not demand schemas since the documents are considered self-described data. The basic market implementation that is mentioned in this case is Mongo DB. In a few words, it uses BSON files and has multiple CRUD operations where CRUD refers to create, read, update, delete. It chooses two-phase commit as a method to ascertain that the ACID rules are valid, and puts into effect master-slave replication. Moreover, it uses hash partitioning as well as range partitioning to execute sharding. Subsequently, the writer analyzes key stores which attempt to function without a programming language and use a sequence of operations in order to achieve mainly availability, but also other traits discussed earlier. The basic method is through a unique key that is assigned to a value. The Dynamo DB system and Voldemort data store are good examples of this type's market functionality. Voldemort's implementation has an interesting aspect, as it applies consistent hashing in order to distribute the keys and the values. Moreover, the column-based method that partitions tables vertically in order to create families of columns has an implementation in the Hbase data model. Hbase is a model similar to Google's Big Table and is the type that is most similar to the common RDBMS, as it stores data in table rows. In addition, it uses timestamps to record different data versions. Finally, the graph databases type is examined, a method that uses vertices and edges to represent data and is put into effect by Java. The implementation that the writer presents is Neo4j, a system that uses a map pattern to inspect the properties of the nodes and relationships that appear in the graphs.

Which questions aren't answered by this text?

The text has a great structure overall but I think the CAP analyses should not interfere between the presentation of the NOSQL types and their implementations, it should have been mentioned after the introduction. The explanations though were thorough and all the important terms were well-explained. Specifically the fact that consistent hashing was analyzed, a relatively easy term in comparison to the rest of the text, shows that the writer looks after the reader.

What has changed since this was written?

The text was written 6 years ago so probably nothing has changed. It has to be stated though that the need for more data is ascending so quickly, that I believe the urge for these systems must have risen significantly in this period.