PHD Thesis Proposal :
Distributed storage and querying of semantic Web data with NoSQL data stores

David Célestin FAYE*

Keywords : NoSQL, Large scale Databases, Cloud Databases, RDF stores

1 Background

The World Wide Web introduces a new scale for modern Web applications, in terms of: concurrency (millions of requests/second from various users who expect the service to be always available and reliable), data (peta-bytes generated daily by these users), processing (this amount of data have to be processed efficiently), exponential growth of users requests at any moment. The appearance of a new form of traffic profile driven by what might be referred to as Web 2.0 has lead to a growing number of Web applications such as Google, Amazon, Yahoo!, Facebook, Twitter and many more, dealing with high traffic, massive data, large user-base and user-generated content. User-generated content means that sites become more "read-write" balanced. One can note that these Web applications are usually free and they do not have the same thoughts compared to traditional large applications requiring consistency such as financial or telecommunication applications

Relational Database Management Systems (RDMBSs) are the predominant technology for storing structured data in Web and business applications. However, Web sites with very large traffic make seldom, if ever, the use of existing RDBMS solutions(e.g., Oracle, Sybase, MySQL, PostgreSQL, etc) even with a high clustering solution. Indeed, the existing clustering solutions are not really scalable when dealing with exponential growth. Additionally, when dealing with normalized database schema applications require the use of joins, which does not perform well under high volume of data and/or nodes.

On the other hand, guaranteeing the consistency of the data may be very expensive in terms of time and it is often incompatible with the performance. Web applications have different needs than the applications that RDBMS were designed for: low and predictable response time (latency), scalability and elasticity (at low cost!), high availability, flexible schemas, semi-structured data and distribution. They can (usually) do without transactions, strong consistency, integrity and complex queries.

So, because consistency is often incompatible with the performance this type of applications can sacrifice data integrity/consistency so that to scale. For example Amazon observed the negative impact of the increase of response time over their sales while Google noticed that the increase of latency affect dramatically traffic rate. Consequently, for these enterprises with high traffic, *availability* may be more important than *consistency*.

*Thesis supervisor – david-celestin.faye@ugb.edu.sn

NoSQL covers a wide range of emerging technologies, data architectures for managing web-scale data or in the context of big data. The NoSQL concept was defined in 1998, but the movement began really in 2009 and it is a large and expanding field, over 120 solutions known[1], with data stores having the following common features : persistent data (not just caches) non-relational data, no joins, distribution, massive horizontal scaling, no fixed schemas, replication support, individual query systems rather than using a standard query language, consistent within a node of the cluster and eventually consistent across the cluster and simple transactions. Some of them have already gained recognition due to adoption by large projects, or technical features. Among them: Bigtable, Dynamo, Hadoop/HBase, Cassandra, SimpleDB, MongoDB, CouchDB, Membase, Voldemort, RavenDB, Berkeley DB. Many of these systems are application specific. This new kind of database focus on particular classes of problems; be more flexible about stored data (document stores), manage use cases like relationships (Graph Databases), aggregate data (Column Databases) or just consider a database as something that stores a value (Key-value Stores). Considering querying capabilities noSQL databases favour performance and scalability over rich dynamic querying features. Most of the NoSQL systems relay on a distributed architecture in which the data is stored in a redundant manner on several servers.

The Semantic Web extends the principles of the Web by allowing computers to understand and easily explore the Web. In recent years RDF has been a widespread data format for the Semantic Web. There is a real need to efficiently store and retrieve RDF data as the number and scale of Semantic Web in real-word applications in use increase. As datasets grow larger and more datasets are linked together, scalability becomes more important. Efficient data storage and query processing that can scale to large amounts of possibly schema-less data has become an important research topic.

Considering Semantic, RDF/W3C is another community wanting to go beyond SQL, with the main motivation to alleviate the need for a central database by allowing everyone to provide a data source and to link it to the others with the benefits of useful well defined inference mechanisms. Like NoSQL, RDF stores focus on schema-less data. However, at the opposite of NoSQL stores, RDF stores make joins and provide powerful query mechanisms with SPARQL query endpoints allowing for declarative queries into the RDF data model, without performance overhead. A NoSQL based RDF store can be benefic for the following reason : RDF could learn a lot from NoSQL about scaling and ease-of-programming; NoSQL could learn a lot from RDF about data portability, decentralization and inference.

2 Research question

The goal of the the thesis is to propose and evaluate a distibuted NoSQL data store to store and query Semantic Web data or complex ontologies with the ability to clearly define link types. Moreover, creating a SPARQL abstraction layer for NosQL based RDF data stores and handling a semantic query rewritting mechanism would allow these data to be effectively queried. The Integration of an Ontology-Based Data Access Approach in NoSQL Stores would provide semantic conceptual schema over a repository of data and, due to its logical formalism, it is likely to support formal analysis, optimization and distibuted reasoning.

[1] http://nosql-database.org/