Integrating NoSQL Technologies into the Virtual Observatory to Support Big Data Challenges

José Antonio Magro Cortés

September 10, 2013

Abstract

The volume of data produced at science centers presents a processing challenge that is getting more and more difficult to deal with. Many science centers do not have the computing or financial resources to manage all of the data on site, so it moved into the grid to share the load with computing centers around the world.

Tools for making it easier for astronomers to operate on those larger datasets are needed, and indeed are priorities of those facilities. In order for observatories to store and publish their observations, they have typically relied on relational database management systems (RDBMS). However, RDBMS have their weaknesses, specially for data publishing applications, as they impose the same schema for datasets which might not have all of their metadata in common, and mandate an ingestion phase that converts the metadata in the original format into a format suitable for RDBMS systems.

Today, non-relational, cloud, or the so-called NoSQL (*Not-only-SQL*) database systems are growing rapidly as an alternative model for database management. These systems are tuned for the kind of big data applications that have made possible very large systems such as Google or Facebook, and can incorporate the documents themselves, and query on the existing metadata, without the need for a dedicated, complicated ingestion phase.

Keywords: BigData, Virtual Observatory, RDMS, NoSQL, MongoDB, MapReduce, Atacama Large Millimeter Array.

1 Big Data challenges in astronomy

Astronomical datasets are growing at an exponential rate: the next generation of telescopes will collect data at rates of several terabytes per day. This data deluge presents some critical challenges for the way astronomers can get new knowledge from their data. These extremely large datasets, or datasets with high data rates, are commonly known as *Big Data*.

In this work, we give a short overview of the big data challenges being faced by astronomy, and present an alternative, using one of the freely available NoSQL databases, and how it can be integrated in the Virtual Observatory framework.

2 Non-relational DBMS

A non-relational database just stores data without explicit and structured mechanisms to link data from different buckets to one another. NoSQL architectures often provide limited consistency, such as events or transactional consistency restricted to only data items. Some systems, however, provide all guarantees offered by systems complying with the Atomicity, Consistency, Isolation and Durability (ACID) criteria by adding an intermediate layer. The ACID properties guarantee the reliable processing of database transactions, but they are not needed for system where the data are not subject to transactions, such as read-only systems. Many NoSQL systems employ a distributed architecture, maintaining data redundantly on multiple servers, often using distributed hash tables. Thus, the system may actually scale adding more servers, and thus a server failure may be tolerated. Unlike RDBMS, the NoSQL databases are designed to expand transparently to scale and they are usually designed with low-cost in mind.

3 Non-relational DBMS inside VO

The International Virtual Observatory Alliance (IVOA) was formed in 2002 with the aim to facilitate the international coordination and collaboration necessary for the development and deployment of the tools, systems and organizational structures necessary to enable the international utilization of astronomical archives as an integrated and interoperating virtual observatory.

After studying the main differences between RDBMS and NoSQL DMBS, we addressed two topics linked with the previous sections: the possibility of adapting FITS headers file to the native storage unit in MongoDB (documents) and using a paradigm specifically designed for processing huge amount of data (e.g. raw data from telescopes).

To support our proposal, we mention some other scientific centers which have used the same (or similar) technologies: the CMS at the LHC (MongoDB), the ATLAS Workload Management System (Cassandra) and an example of measuring radiation levels in Seattle (CouchDB).

3.1 FITS file format

When working with FITS files many issues arise: its inadequacy for storing information, the great amount of possible key-value pairs in FITS headers and having multiple FITS formats. We show a logical layout using MongoDB features to solve the previously posed problems.

3.2 MapReduce

MapReduce, developed by Google, is a programming model and its implementation for processing (*e.g.* raw data from telescopes) and producing (*e. g.* representations of graph structure of systems) huge amounts of data. The program executes across several hundred or thousands of nodes and even controlling machine failures. MapReduce, usually, is used to solve problems involving big size datasets, up to several petabytes. For this reason, this model is used in distributed file systems, like HDFS. Many problems in astronomy naturally fall into this model because of the inherent parallelizability of many astronomical tasks.

3.3 ASA scaling to NoSQL

In order to scale ALMA Science Archive to MongoDB, we have defined a MongoDB schema for ObsCore, a data insertion plan using a FITS alternative in MongoDB and found the necessary changed in code to use MongoDB methods.

4 Conclusions and future work

4.1 Conclusions

- Relational approach are not always suitable for any problem. Non-relational systems are not cure-all, but it has been shown they can face some problems in a more efficient way (*e.g.* MapReduce) and, in some situations, NoSQL can be a complement for existing RDBMS.
- Any new proposal should be inside Virtual Observatory frame.
- NoSQL database systems, specially, not exclusively, those document-oriented can reduce system analysis and design and can also succeed in boosting the performance of data management.

4.2 Future work

- Focusing in a workgroup inside Virtual Observatory instead of treating several aspects.
- The use of formal metrics (CoCoMo, Function Point Analysis) to plan the software design and development and the costs involved.
- Benchmarks support in order to obtain accurate data in performance improvements.
- Deciding which language to use to connect the selected DBMS.

References

[1] Kristina Chodorow, *MongoDB: The Definitive Guide*. O'Reilly, 2010.

[2] Abraham Silberschatz *et al.*, *Database System Concepts Sixth Edition*. McGraw-Hill, 2010.

[3] SKA Telescope, *The age of astronomy Big Data is already here*. 2013.

[4] Centre for Astrophysics and Supercomputing, *Scientific Computing and Visualisation*. Winburne University of Technology, 2013.

[5] Juan de Dios Santander Vela, *Integration of tools and radioastronomical archives in the VO architecture* (PhD Thesis). Universidad de Granada, 2009.