NoSQL: The Death of the Star

Alberto Abelló*

*Omega-125 Campus Nord UPC
Jordi Girona 1-3, 08034 Barcelona (Spain)
aabello@essi.upc.edu,
http://www.essi.upc.edu/ aabello

Abstract. In the last years, the problems of using generic storage techniques for very specific applications has been detected and outlined. Thus, some alternatives to relational DBMSs (e.g. BigTable and C-Store) are blooming. On the other hand, cloud computing is already a reality that helps to save money by eliminating the hardware as well as software fixed costs and just pay per use. Thus, specific software tools to exploit the cloud have also appeared. The trend in this case is to use implementations based on the MapReduce paradigm developed by Google. The basic goal of this talk will be the introduction and the discussion of these ideas from the point of view of Data Warehousing and OLAP. We will see advantages, disadvantages and some possibilities it offers.

1 Introduction

Nowadays, most companies externalize as many services as possible to reduce costs and be more flexible in front of fluctuations of the demand. Thus, with the cloud, the time has arrived to IT infrastructures. The National Institute of Standards and Technology (NIST) defines *cloud computing* as "a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction".

Cloud computing, in general, is a good solution for medium to small companies that cannot afford a huge initial investment in hardware together with an IT department to manage it. With this kind of technologies, they can pay per use, instead of provisioning for peak loads. Thus, only when the company grows up (if at all), so the expenses will. The only problem is that they have to trust their data to third parties.

In Abadi (2009), we find an analysis of pros and cons of data management in the cloud. It is found completely inappropriate for transactional processing mainly due to the problems to guarantee ACID properties in such environment. However, it is adequate for analysis environments, since those properties are not needed. It also outlines the problem of having data in an untrusted environment, which would again be unacceptable in transactional processing, but can be easily solved in analytical systems by just leaving out some sensitive data or using an anonymization function. On the other hand, what cloud data management can offer to an analytical environment is elastic compute power (in the form of parallelism), replication of data

(even across different regions of the world), and fault tolerance (a new machine automatically taking over from a fallen one without re-executing the whole query or process).

Data Warehouses (DW) and On-Line Analytical Processing (OLAP) tools where defined by Bill Inmon in 1992 and Edgar Codd in 1993, respectively. Thus, they are almost twenty years old and have evolved to maturity by overcoming many limitations in these years. Huge (Terabytes) relational DW exist today benefiting from techniques like materialized views, bitmap indexes, etc. Nevertheless, some challenges remain still open. Mainly, they are related to the management of ETL processes, unstructured data, and schema evolution.

Cloud computing does not mean that we cannot use a relational system. Indeed, well known alliances already exist in the market, like that between Oracle and Amazon. However, as pointed out in Inmon et al. (2008), there is four to five times as much unstructured data as there is structured data. NoSQL engines like BigTable (presented in Chang et al. (2008)) are thought to store this kind of data. For example, with this approach, we could easily incorporate information extraction tools to the management of unstructured data in the sources. A non-formatted chunk of data would be stored associated to the key, and just when the user decides what to do with it and which tool can be used, we do it. Oppositely, in a RDBMS, we format and structure data in a star-shape schema months or even years before it is actually used (or even without knowing whether it will be used or not).

It is at this point that MapReduce (presented in Dean and Ghemawat (2004)) can help, because it was conceived to parallelize the parsing and modification of data as it is being scanned. MapReduce is a framework that hides distribution of data, parallelization, fault-tolerance and load balancing from the programmer. It has been specially designed for scalability, and processing in the cloud. It allows to deal with huge volumes of data, when the schema is not concrete (i.e., variations can eventually appear).

References

Abadi, D. J. (2009). Data management in the cloud: Limitations and opportunities. *IEEE Data Engineering Bulletin 32*(1), 3–12.

Chang, F. et al. (2008). Bigtable: A distributed storage system for structured data. *ACM Transactions on Computer Systems (TOCS) 26*(2).

Dean, J. and S. Ghemawat (2004). Mapreduce: Simplified data processing on large clusters. In *6th Symposium on Operating Systems Design and Implementation (OSDI)*, pp. 137–150.

Inmon, W., D. Strauss, and G. Neushloss (2008). *DW2.0*. Morgan Kaufmann.

Résumé