

Using SLAs to guide database transition to NoSQL on the cloud: a systematic mapping study

Fabio Leal - sousaleal.fabio@gmail.com

December 8, 2014

1 Abstract

Cloud computing became a reality over the last years, and many companies are now moving their data-centers to the cloud. A concept that is often linked with cloud computing is Infrastructure as a Service (IaaS): the computational infrastructure of a company can now be seen as a monthly cost instead of a number of different factors. Recently, lots of systems started to replace their relational databases with hybrid solutions (NoSQL DBs, Search Engines, OR-DBs). These changes are motivated by i) performance improvements on the overall performance of the system and ii) inability to a RDBMS to provide the same performance of a hybrid solution given a fixed monthly infrastructure cost. However, not always the companies can exactly measure beforehand the future impact on the performance on their services by making this sort of technological changes (replace RDBMS with another solution). The goal of this systematic mapping study is to investigate the use of Service-Level-Agreements (SLAs) on database-transitioning scenarios and to verify how SLAs can be used in this processes.

2 Introduction

2.1 Cloud Computing

On the early 90's it was commonplace for every Information Technology (IT) company to have its own Data Center with lots of huge servers and main-frames. IT costs were high, and high-performance computing was available only for big companies, as data centers required a lot of physical space and high costs for maintenance.

The regular way of building a web application was to use client-server approach, where the server was an extremely powerful (and expensive) machine. However, new companies, such as Google, were rising with bigger missions: *“to organize the world's information and make it universally accessible and useful”*. It was *just* impossible to store the petabytes of daily-generated data in a single server.

From this point, they also realized that it was much cheaper to build and maintain several low-performance servers than a single high-performance machine. This approach, however, was incompatible with the traditional way of building applications, as they were designed to work with a single server and database.

Several researches were conducted in this area and a common solution was rising: to distribute data storage and processing. Google, Yahoo and other big IT players helped to build open source tools to make this approach possible, like Hadoop.

This revolution brought to life new concepts, such as Infrastructure as a Service (*IAAS*), Platform as a Service (*PAAS*) and Software as a Service (*SAAS*).

[2] says that *Cloud computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the data centers that provide those services.*

2.2 Data Integration & Polyglot Persistence

On the last years, the number of Data Base (DB) Engines grew like never before. Along with the NoSQL movement and expansion of Social Networks, new concepts for Database Models appeared, like Document Store, Search Engines, Key-Value store, Wide Column Store, Multi-Model and Graph DBMS.

[8] presents a ranking of the most popular DB engines.

Today, instead of having a single Relational Database Management System (DBMS) for the whole application, it is efficient and cost-effective to have several Data Base Engines, one for each type of data that the application handles. This concept is called Polyglot Persistence.

Take for instance a classic e-commerce application that deals with a catalog, user access logs, financial information, shopping carts and purchase transactions. These data have different management requirements, for instance, the catalog has a lot of reads and very few writes assuming that the catalog of a shop is more or less stable. Information about user sessions require rapid access for reads and writes but they do not need to be durable. The shopping carts need high availability across multiple locations, and can merge inconsistent writes. User activity logs imply high volume of writes on multiple nodes, while recommendations for users must provide rapid link traversals between friends, product purchases and ratings. [10]

As computing services started to decentralize, developers started to build applications that depended of several data-sources. By this time the use of Web Services and Service Oriented Architecture (SOA) became more popular.

2.3 Service Level Agreements (SLA)

According to *ITILv3's* official glossary [1], a Service Level Agreement (SLA) is *an agreement between an IT service provider and a customer. A service level agreement describes the IT service, documents service level targets, and specifies the responsibilities of the IT service provider and the customer.*

The agreement consists on a set of measurable constraints that a service provider must guarantee to its customers.

2.4 Systematic Mapping

According to [7], "A software engineering systematic map is a defined method to build a classification scheme and structure a software engineering field of interest". Systematic Mapping studies provide a global view of a given research field and identify the quantity, results, and the kinds of researches in this field.

A Systematic map is composed by a number of steps:

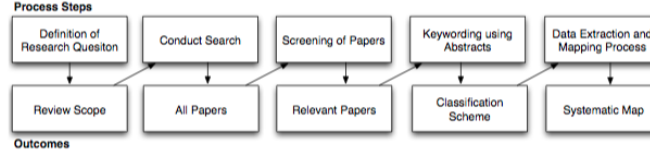


Figure 1: Systematic Mapping Steps

3 Systematic Mapping

3.1 Definition of the research Scope

As we want to investigate how SLAs can be used in database transitioning processes, we proposed three research questions to be answered by our study:

RQ1) How can SLAs be used to guide database transitioning to NoSQL databases in cloud-based apps?

RQ2) How can an SLA be used to monitor the performance improvements promised by a database transitioning process?

RQ3) Is there a standard representation of SLAs in cloud services?

3.2 Conduct Search for Primary Studies (All Papers)

We searched for papers on IEEE, ACM and Elsevier, using Google Scholar as a meta-search engine. We also tried to include industry-related conferences, but few of them had online videos available and the vast majority had no written publications.

We used

"changes AND database AND nosql AND sla AND cloud"

as a boolean search string on Google Scholar search engine. This resulted in a total of 88 publications.

3.3 Screening of papers

Inclusion and Exclusion criteria were used to filter the list of publications:

1. Inclusion Criteria:

- English written publications
- Works that make use of SLAs

2. Exclusion Criteria

- Access-restriction to the original publication
- Non-technical publications
- Publications that don't make any use of SLAs

After applying the inclusion and exclusion criteria our systematic mapping study ended up with 15 relevant publications.

3.4 Classification of the Papers

We classified each selected papers in three facets: **Contribution**, **Technology** and **SLA representation**. On the conceptual map conceptual map we have listed just a subset of the several technology types that were found on the review.

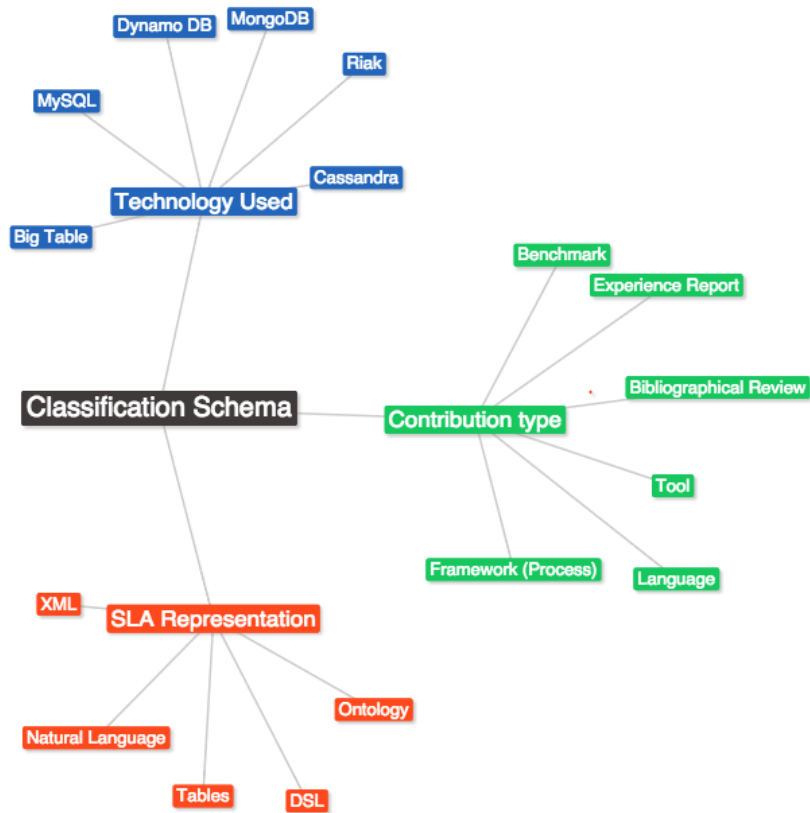


Figure 2: Classification scheme for selected papers

4 Outcomes

As mentioned on "Screening of papers" sub-section, 15 publications were analyzed, and the answers to the research questions are presented below:

RQ1) How can SLAs be used to guide database transitioning to NoSQL databases in cloud-based apps?

A number of works were found relating SLAs with Quality of Service (QoS) and Quality of Experience (QoE). Several publications, such as [11], [6] and [5] propose a SLA-centric approach to monitor and control the performance of cloud-based apps. However, no publication was found addressing the problem of switching from one database technology to another and guiding this process through an SLA.

RQ2) How can an SLA be used to monitor the performance improvements promised by a database transitioning process?

[9], [13], [4] and [12] propose SLA-centric/User-Centric solutions to monitor the performance of web applications. All these solutions are technology-agnostic and could be used to monitor the performance improvements promised by a database transitioning process.

RQ3) Is there a standard representation of SLAs in cloud services?

The selected publications did not present a standardized representation for SLAs. In fact, 60% of the selected publications did not mention how the SLA was internally represented.

[3] proposes SYBL: *An Extensible Language for Controlling Elasticity in Cloud Applications*. SYBL allows specifying in detail elasticity monitoring, constraints, and strategies at different levels of cloud applications, including the whole application, application component, and within application component code. Based on simple SYBL elasticity directives, our runtime system will perform complex elasticity controls for the client, by leveraging underlying cloud monitoring and resource management APIs.

As a result of the Systematic Map, 5 kinds of representation for SLAs were discovered: DSLs, Relational tales, ontologies and XML variations.

References

- [1] Axelos. Best management practice portfolio: common glossary of terms and definitions, October 2012.
- [2] A. et al. CONNOLY, D. FOX. A view of cloud computing, April 2010. [Online; posted 01-April-2010].
- [3] G. Copil, D. Moldovan, Hong-Linh Truong, and S. Dustdar. Sybl: An extensible language for controlling elasticity in cloud applications. In *Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on*, pages 112–119, May 2013.
- [4] M. Klems, D. Bermbach, and R. Weinert. A runtime quality measurement framework for cloud database service systems. In *Quality of Information and Communications Technology (QUATIC), 2012 Eighth International Conference on the*, pages 38–46, Sept 2012.
- [5] Markus Klems, David Bermbach, and Rene Weinert. A runtime quality measurement framework for cloud database service systems. In *Proceedings of the 2012 Eighth International Conference on the Quality of Information and Communications Technology, QUATIC '12*, pages 38–46, Washington, DC, USA, 2012. IEEE Computer Society.
- [6] Ioannis Konstantinou, Evangelos Angelou, Dimitrios Tsoumakos, Christina Boumpouka, Nectarios Koziris, and Spyros Sioutas. Tiramola: Elastic nosql provisioning through a cloud management platform. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, SIGMOD '12*, pages 725–728, New York, NY, USA, 2012. ACM.
- [7] Kai Petersen, Robert Feldt, Shahid Mujtaba, and Michael Mattsson. Systematic mapping studies in software engineering. In *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering, EASE'08*, pages 68–77, Swinton, UK, UK, 2008. British Computer Society.
- [8] DB Ranking. Db engines ranking, May 2014.
- [9] S. Sakr and A. Liu. Sla-based and consumer-centric dynamic provisioning for cloud databases. In *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*, pages 360–367, June 2012.
- [10] Genoveva Vargas Solar. Addressing data management on the cloud: tackling the big data challenges, May 2014.
- [11] Pengcheng Xiong. Dynamic management of resources and workloads for rdbms in cloud: A control-theoretic approach. In *Proceedings of the on SIGMOD/PODS 2012 PhD Symposium, PhD '12*, pages 63–68, New York, NY, USA, 2012. ACM.

- [12] Pengcheng Xiong, Yun Chi, Shenghuo Zhu, Junichi Tatemura, Calton Pu, and Hakan Hacigümüş. Activesla: A profit-oriented admission control framework for database-as-a-service providers. In *Proceedings of the 2Nd ACM Symposium on Cloud Computing*, SOCC '11, pages 15:1–15:14, New York, NY, USA, 2011. ACM.
- [13] L. Zhao, S. Sakr, and A. Liu. A framework for consumer-centric sla management of cloud-hosted databases. *Services Computing, IEEE Transactions on*, PP(99):1–1, 2013.