

Analysis of efficient NoSQL data storage mechanisms for a structured social notification feed

Research thesis proposal

Florian Dejonckheere

Abstract

Since the introduction of dynamic Web 2.0 web applications and social networking sites, there has been an increasing demand for performant storage systems to keep up with the vast deluge of data that is generated by capturing the social activity of users. This data is characterized by its size and inherently structured nature: most types of social networks can be modeled using certain predetermined graph topologies.

In this paper, the current state of affairs related to the storage of non-critical notifications in small to medium-size web applications will be summarized. An analysis of the structured and linked data provided by the Open Weblides (2017) project will be made, and an expandable abstraction will be designed in this context. This design will then be matched to a stable, scalable and maintainable storage system. A concrete implementation plan and roadmap for developing and rolling out the mentioned abstraction mechanism will also be presented.

Keywords

Webapplications — Social feed — Databases — Cloud — NoSQL — Social Graph — Big Data

Contact: florian@floriandejonckheere.be

Contents

1	Introduction	1
2	State of the art	1
3	Methodology	2
4	Expected results and conclusions	2
	References	2

1. Introduction

The Open Weblides (2017) project provides a user-friendly platform to collaborate on webslides - slides made with modern web technologies such as HTML, CSS and JavaScript. One of the core features this application provides is *co-creation*. The co-creation aspect manifests itself in several forms within the application; annotations on slides and a change suggesting system resembling GitHub's pull request feature are the main mechanisms. Because of the inherent social nature of co-creation, a basic notifications feed was also implemented. This feed is tailored to the user, and reflects the most recent changes, additions and comments relevant to the slide decks the user is interested in.

However, the functionality implemented in the system contains only the bare necessities at the moment. The module will be expanded in the future, and doing so requires a structural and conceptual rethinking of how the notifications are generated, stored and queried. This paper has two concrete goals: First, it aims to analyze and summarize the existing

frameworks and software packages commonly used in the industry to store structured non-relational graph or document data. In order to achieve this, an introduction into NoSQL and the different types of NoSQL database management systems will be proposed first, before proceeding into a comparative study between different vendors.

Second, the linked structure and information provided by the Open Weblides' social notification feed will be evaluated, and an abstract, maintainable and expandable structure will be designed. This allows any future development to easily add additional notifications into the system. This data schema will then be evaluated in the context of the comparative study. Finally, a recommendation will be made for a concrete implementation of the described data storage schema.

2. State of the art

In current literature, many existing studies have already described the relation between traditional relational database systems and NoSQL stores. However, since this paper covers a specific use case no further general comparative studies will be referenced.

The 2015 doctoral thesis (Zhao, 2015) describes the development of a messaging system for astrophysical transient event notifications. Part of this analysis is a comparison between document-based NoSQL storage solutions fit for this particular use case. We expect this paper to provide a solid base of reasoning in order to find a scalable and efficient solution for resolving similar computational challenges.

The main difference between the previously mentioned studies and this paper is the specific use case. This paper tries to present a solution tailored to the specific requirements of the Open Weblides project. This entails a different handling of certain key data-structures within the provided platform, particularly concerning querying the stored data.

3. Methodology

First, a comparison of existing NoSQL database management systems will be presented. In this comparative study, both multiple types and multiple vendors of NoSQL systems will be compared against each other. Criteria for comparison include how the database management system concretely stores its data on disk, the query format and specific programming language bindings. Another important aspect is the distributed nature of many NoSQL databases. Using Brewer's conjecture (Gilbert & Lynch, 2002, 2) – often called the CAP theorem – the existing types of data storage systems will be examined and summarized. There is also a practical factor present in the research; this includes the license of the project, its active maintainability and future prospects. Common types of NoSQL databases include key-value store, column-oriented, document store and graph databases (Nayak, Poriya, & Poojary, 2013). This paper will give a short introduction to these types, before proceeding to examine the best fitting types further in detail.

Second, the data model specific to the Open Weblides project will be examined. We will start from the data model that is already implemented in the current iteration of the platform. At the time of writing, the existing base implementation of the social notification feed only contains one or two types of notifications. This paper will try to extrapolate this concept into a more generalized, abstract system where developers can easily plug in additional notification types. The physical properties of the data model will also be taken into account: the data will be written once to the data storage, and read many times later. It is also highly interlinked information, as a notification will always relate to one or more users as a subject, and an object class – most likely a slide deck or collection of slide decks. These links need to be maintained, and efficiently reconstructed when queried.

Finally, a sample data set will be constructed using the aforementioned detailed analysis. Empirical testing will be conducted against multiple database management systems, and the results will be summarized and interpreted. Various information flows will be tested, however the most important process remains efficiently querying the stored data.

Using the comparative study of storage engines, data model analysis and the empirical results an implementation plan will be constructed. This plan will serve as a recommendation for future development.

4. Expected results and conclusions

The NoSQL ecosystem, unlike relational databases, is headed towards specialization, so different solutions are headed in different directions (Maroo, 2013). In this paper, we expect to find one type of NoSQL database that is a better fit for the Open Weblides use case, in clear contrast with the other types of storage engines. Due to the inherently highly interlinked nature of the stored data, we suspect a graph-based database management system to provide the most advantages, and generally the most performant experience.

This expectation is amplified by a good availability of Ruby bindings to the most popular graph database management systems.

Since the platform being discussed only caters to a small to medium user base, we do not expect the need to scale horizontally beyond one instance. However, the vertical scalability is still a topic for discussion, and we expect to determine the computational order of magnitude in order to efficiently query the given dataset during this study.

Finally, the implementation plan should describe a concrete roadmap, stretching over a development period with a baseline expectation of one to three months. Roll-out of this mechanism should also be included in this plan.

References

- Gilbert, S. & Lynch, N. (2002). Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *ACM SIGACT News*, 33, 51–59.
- Kaur, K. & Rani, R. (2013). Modeling and querying data in nosql databases. In *2013 ieee international conference on big data* (pp. 1–7). doi:10.1109/BigData.2013.6691765
- Maroo, T. (2013). *Handling with dynamic, large data sets - nosql a buzzword or savior?* JECRC Foundation.
- Nayak, A., Poriya, A., & Poojary, D. (2013, March). Type of nosql databases and its comparison with relational databases. *International Journal of Applied Information Systems (IJ AIS)*, 5(4).
- Open Weblides. (2017). Open Weblides. Retrieved from <http://openweblides.github.io/>
- Zhao, Y. (2015). *Event based transient notification architecture and nosql solution for astronomical data management* (Doctoral dissertation, Massey University).