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

Research thesis proposal

Florian Dejonckheere

Abstract

Ever since the dawn of dynamic web applications and social networks, there has been an increasing demand for performant storage systems to capture the vast deluge of data that is generated by 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 pertaining to the storage of data in small to medium-size web applications will be summarized. An analysis of the structured data within the context of the Open Webslides (2017) project will be made, and an attempt will be made to match this analysis to a stable, scalable and maintained storage system. A proposed implementation plan for a normalized, efficient data schema will also be presented.

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

Contact: florian@floriandejonckheere.be

Contents

1. Introduction

The Open Webslides (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.

Second, the structure and data provided by the Open Webslides' social notification feed will be interpreted in the context of the aforementioned analysis. 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 Webslides 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 Webslides 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

Hier beschrijf je welke resultaten je verwacht. Als je metingen en simulaties uitvoert, kan je hier al mock-ups maken van de grafieken samen met de verwachte conclusies. Benoem zeker al je assen en de stukken van de grafiek die je gaat gebruiken. Dit zorgt ervoor dat je concreet weet hoe je je data gaat moeten structureren.

5. Expected conclusions

Hier beschrijf je wat je verwacht uit je onderzoek, met de motivatie waarom. Het is **niet** erg indien uit je onderzoek andere resultaten en conclusies vloeien dan dat je hier beschrijft: het is dan juist interessant om te onderzoeken waarom jouw hypothesen niet overeenkomen met de resultaten.

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.

Nayak, A., Poriya, A., & Poojary, D. (2013, March). Type of nosql databases and its comparison with relational databases. *International Journal of Applied Information Systems (IJAIS)*, *5*(4).

Open Webslides. (2017). Open Webslides. Retrieved from http://openwebslides.github.io/

Zhao, Y. (2015). *Event based transient notification architecture and nosql solution for astronomical data management* (Doctoral dissertation, Massey University).