PhD Thesis Proposal

1 Title: Distributed Data Structure Server

1.1 Background

The appearance of web applications needing to support many thousands of concurrent users, like Twitter and Facebook led to the rise of distributed key-value stores, to enable scalability and fault-tolerance for scenarios in which traditional relational databases are not appropriate.

In particular, two aspects make traditional ACID offering relational databases problematic: the need to reply to many concurrent requests with low latency makes the use of joins over normalized data unfeasible; the desire to offer fault tolerance and always-on availability leads to the use of replication (in some cases across several datacenters). The CAP theorem [1] made clear that under a replicated scenario, if availability and partition tolerance is desired, one must give up strong consistency and end up offering eventual consistency semantics [7].

Since Amazon's Dynamo distributed key-value store [3], research on eventually consistent, non ACID, non relational distributed databases has surged. Many other systems have been designed, like Facebook's Cassandra, Google's BigTable, MongoDB, HBase, Riak, in what has been popularly designated "NoSQL" [4].

These key-value stores typically offer a simple get/put API to retrieve or store data for a given key, as opposed to a traditional query language (e.g., SQL). The concurrent use of gets and puts, without support for transactions creates many problems for the application programmer. These include the loss of updates (overwritten by a concurrent client) or the need to reconcile multiple values that result from conflicting updates.

A common example is the difficulty of maintaining a simple counter, due to the lack of atomicity in the get-increment-put operations. Given its general usefulness, support for counters has been added to Cassandra recently, after a lengthy discussion of the problem. In fact, distributed counting is a classic problem that has been addressed over many years, but mostly with an emphasis on scalability and offering strong consistency semantics (e.g., linearizability [5]), and not availability and partition tolerance.

1.2 Research question

This proposal aims to address the problem of whether it is possible to support richer operations (as opposed to just a simple get/put) on replicated, highly available, fault tolerant, weakly consistent distributed datastores, in a reasonably efficient manner and offering some sensible semantics.

Much past research has addressed strong consistency (e.g. atomic concurrent objects [5], Database State Machine [8, 6]), and this thesis aims to explore some similar questions in the weakly consistent setting.

On the other hand, the idea of providing a NoSQL database with richer operations on datatypes has been implemented in the Redis key-value store, for this reason also called a “data structure server”. Redis [4, 2], however does not support replication (apart from a master-slave scheme), using distribution mainly for partitioning the data onto several machines. This proposal can also be regarded as a generalization of this for the fully symmetric replication case.

1.3 Goals for the Pre-Thesis

- Research the state of the art in distributed NoSQL datastores.
- Evaluate possible candidate data structures to be supported by distributed datastores;
- Explore the design space of consistency semantics to be offered;

2 Free Option

This course is going to be a “Supervised Study”. In the topic of supporting data structures in distributed datastores, distributed counting is a classic case study. Therefore, it is proposed that the student researches this specific topic and writes a survey about the state of the art in “Distributed Counting”.

3 External Option

The proposed external option is a course in the Statistics Master Program. The specific course will be *Probabilidades e Modelação Estocástica*

4 Research Unit

HASLab / INESC TEC,
Universidade do Minho,
Portugal

Braga, March 4th, 2012,

The Supervisors:

(Paulo Sérgio Almeida)

(Victor Francisco Fonte)

The Student:

Ricardo Jorge Tomé Gonçalves

(Ricardo Jorge Tomé Gonçalves)

References

[1] Eric A. Brewer. Towards robust distributed systems (abstract). In *PODC '00: Proceedings of the nineteenth annual ACM symposium on Principles of distributed computing*, page 7, New York, NY, USA, 2000. ACM.

[2] Sponsored by VMware. Redis: an open source, advanced key-value store.

[3] Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall, and Werner Vogels. Dynamo: amazon's highly available key-value store. In *SOSP '07: Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles*, pages 205–220, New York, NY, USA, 2007. ACM.

[4] Jing Han, E. Haihong, Guan Le, and Jian Du. Survey on nosql database. In *Pervasive Computing and Applications (ICPCA), 2011 6th International Conference on*, pages 363 –366, oct. 2011.

[5] Maurice P. Herlihy and Jeannette M. Wing. Linearizability: a correctness condition for concurrent objects. *ACM Trans. Program. Lang. Syst.*, 12:463–492, July 1990.

[6] Fernando Pedone, Rachid Guerraoui, and André Schiper. The database state machine approach. *Distributed and Parallel Databases*, 14:71–98, 2003. 10.1023/A:1022887812188.

[7] Yasushi Saito and Marc Shapiro. Optimistic replication. *ACM Comput. Surv.*, 37:42–81, March 2005.

[8] A. Sousa, F. Pedone, R. Oliveira, and F. Moura. Partial replication in the database state machine. In *Network Computing and Applications, 2001. NCA 2001. IEEE International Symposium on*, pages 298 –309, 2001.