Brief Announcement: Reconfigurable Byzantine-Fault-Tolerant Atomic Memory*

Rodrigo Rodrigues and Barbara Liskov
MIT Computer Science and Artificial Intelligence Laboratory
{rodrigo,liskov}@csail.mit.edu

Introduction

Quorum systems are valuable tools for building highly available replicated data services. A quorum system can be defined as a set of sets (called quorums) with certain intersection properties. The use of quorum systems in data services is based on the idea that read and write operations need be performed only at a quorum of the servers, since the intersection properties will ensure that any read operation will have access to the most recent value that was written.

Traditionally, quorum systems assumed that servers fail benignly, i.e., by crashing or omitting some steps. Recently, Reiter and Malkhi extended quorum systems to provide data availability in the presence of arbitrary (Byzantine) faults [3].

One important limitation of standard Byzantine fault tolerance techniques, quorum-based or otherwise, is that they assume a fixed set of servers that provide the data service throughout the entire system lifetime. This assumption is problematic, since in a long-lived deployment machines may have to be added and removed from the system.

This paper addresses this limitation by presenting an algorithm for a reconfigurable Byzantine quorum system. Our algorithm ensures atomicity with an asynchronous network despite Byzantine failures of servers, crash failures of clients, and reconfigurations.

Design Overview

Our work builds on existing Byzantine quorum algorithms [3, 4] that provide atomicity in a static system. We are the first to extend these algorithms to support a dynamic membership (Alvisi et al. [1] describe a scheme that allows the failure threshold to change in a Byzantine quorum system, but still considered a fixed replica set).

Our algorithm can be summarized as follows. We use a logically-centralized reconfigurer (implemented as a Byzantine agreement group [2]) that generates new configurations with associated epoch numbers. We tag all messages with epoch numbers. In each epoch, we execute reads and write as in existing atomic Byzantine quorums [3, 4]. If the client discovers that a server it contacts is in a new epoch, it upgrades its configuration, and restarts the operation in the new epoch. If a server discovers that it is in an old epoch, it upgrades its configuration, stops accepting requests for the old epoch, determines if it became responsible for storing objects, and then performs a state transfer from a quorum of nodes in the old epoch.

Other Contributions

This high-level description hides several interesting aspects of the protocol:

- State transfer requires retaining information about old configurations. We present a new garbage-collection protocol that allows this information to be deleted when it is no longer needed, even in the presence of malicious nodes that may, for instance, pretend that they have not received recent configurations.
- The system relies on limiting the number of failures in a replica group that occur within a window of vulnerability. (Outside this time window, all servers in that group may fail arbitrarily.) We present correctness conditions that determine the size of this window, expressed in terms of the occurrence of certain events. Additionally, we prove that our system satisfies these conditions.
- We address the issue of slow clients trying to contact old replica groups whose failure threshold has been exceeded.
- We present an analytic performance model for a heterogeneous deployment (variable inter-node latency), and experimental results from our implementation that validate this model.

References

[1] L. Alvisi, D. Malkhi, E. Pierce, M. Reiter, and R. Wright. Dynamic Byzantine Quorum Systems. In *International Conference on Dependable Systems and Networks (DSN 2000).*

[2] M. Castro and B. Liskov. Practical Byzantine Fault Tolerance. In *Proc. of the 3rd Symposium on Operating Systems Design and Implementation (OSDI)*, Feb. 1999.

[3] D. Malkhi and M. Reiter. Byzantine Quorum Systems. *Journal of Distributed Computing*, 11(4):203–213, 1998.

[4] D. Malkhi and M. Reiter. Secure and scalable replication in Phalanx. In *Proc. of the 17th IEEE Symposium on Reliable Distributed Systems*, Oct. 1998.

[*] This work is supported by the NSF under cooperative agreement ANI-0225660. Rodrigo Rodrigues is supported by a fellowship from the Calouste Gulbenkian Foundation, and was previously supported by a Praxis XXI fellowship.

PODC'04, July 25–28, 2004, St. John's, Newfoundland, Canada.
ACM 1-58113-802-4/04/0007.