

CSE 223B Project Check-in

Connie Chen

Russell Reas

Rakesh Varna

May 23, 2013

1 Introduction

This project aims to implement a backend-agnostic distributed key-value storage system. Our main deliverable includes Chord-based node management software along with basic implementations of drivers for in-memory data and the Redis key-value server. Only a subset of the operations available with Redis will be supported.

In addition to supporting node joining and failure as outlined in the Chord paper, we plan to experimentally explore various methods for reducing the $O(\log n)$ complexity of read operations. Furthermore, we plan to implement a simple replication model on top of the Chord system.

2 Software Artifact

The main software artifact is a multi-threaded server written in Python for the Thrift RPC framework. As noted above, we plan to include example drivers for both in-memory and the Redis key-value stores. A small subset of operations, perhaps just `get` and `put`, will be supported as building a key-value server itself is not the emphasis of the project.

3 Design of System

We will apply the same consistent hashing approach outlined in the Chord paper. Similar to the replication approach used in Amazon's Dynamo, we will copy some data from the key primary to $N \geq 0$ successors where N is a user-controlled parameter.

4 Comparison to Previous Work

Several similar projects exist, and we will discuss differences and similarities:

- Redis Cluster, Redis Sentinel, and twemproxy
- Dynamo
- The Chord paper describes complexity reduction for the average case by expanding the size of the finger table

5 Evaluation

At this point we plan two main experiments: Measure replication in node joining and failure scenarios. At what rate of failure do we start to see data loss in the system? Evaluate the performance of `get` requests under various types of load, for example, accessing the same key repeatedly or accessing random keys repeatedly. Which cases do faster lookup schemes improve?

Access to some type of cluster environment would be helpful. This will allow us to run large numbers of nodes. We currently have the ability to run multiple nodes on a single machine as well.