

## **CS 425 MP4**

**Karan Shah (khshah3)**

**Martin Wozniewicz (wozniew1)**

### **Key Value Store**

We used the Hashing approach for splitting up the ring of machines. The reason for this was because it requires less reshuffling when a machine joins or leaves. Instead of shuffling all of the data in order to equally space out the members when the member list changes, all we need to worry about is the machine in question's successor or predecessor (depending on whether it is joining or leaving).

The entire time, all machines are randomly gossiping their rings to one another in order to keep the system in sync and hopefully avoid these conflicts in the first place.

### **HeartBeating**

Heartbeats are used for failure detection. At every interval of a gossip a machine updates the receiving machines heartbeat acknowledging that he is alive. This helps maintain a list of live machines in the group and helps in replication during failures as described below.

### **Replication**

Since we used a ring and chord like key value store algorithm. Whenever a machine fails the data is now meant to be held by its successor. Hence for any given machines on the ring its N successors are replica for its data. Here 'N' is determined as a constant based on the scalability needed and spatial choice. Hence when it fails its replica immediately has all the data. Yet what's more important is that when the successor realizes that it failed (by Gossiping) it will ensure that its Nth replica has all the data that was passed from its predecessor hence it ensures that there are always N replicas for all pieces of data. We used passive replication so that all data first goes to the primary data-holder and then is applied to its replicas.

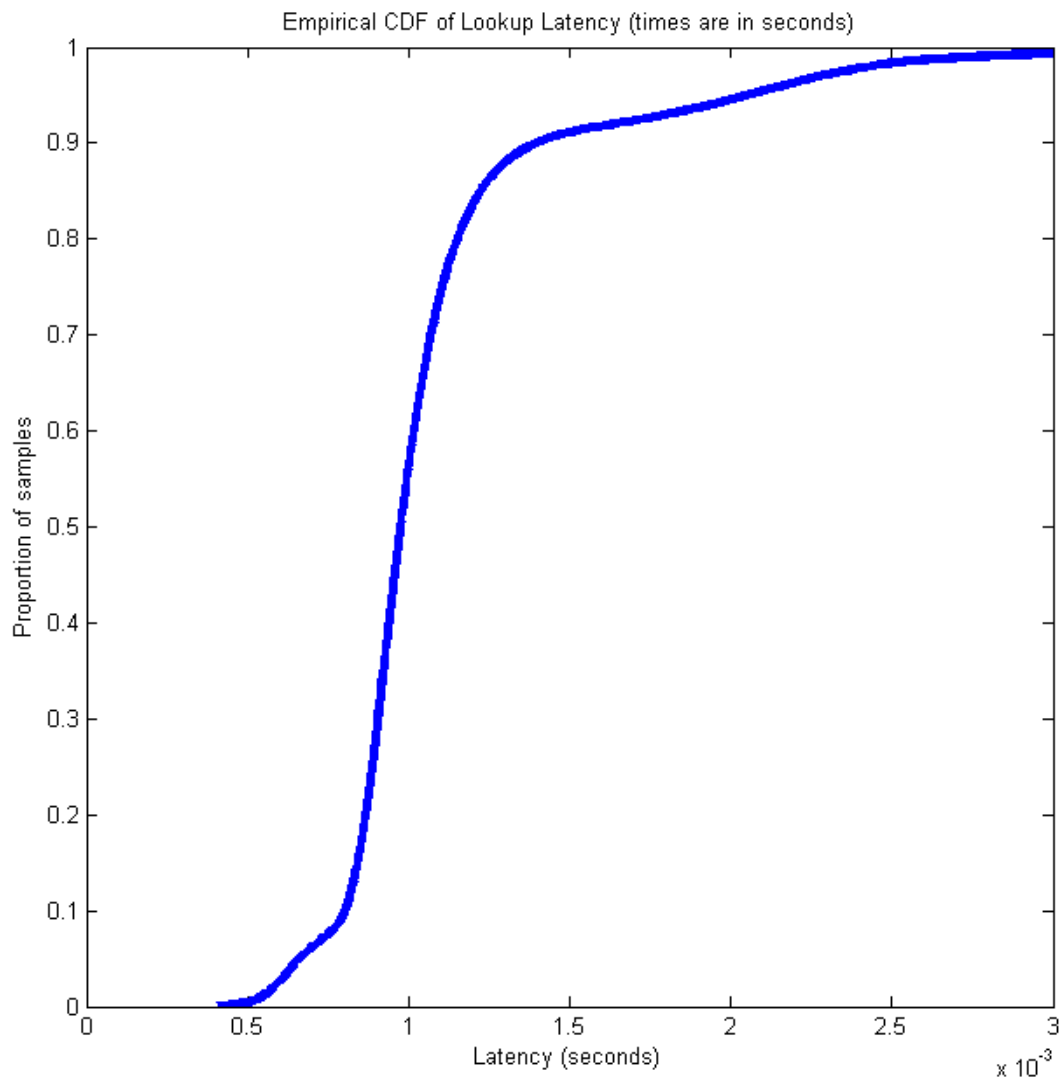
### **Concurrency**

Based on our model when a machine needs to ensure writing to 1 replica we write to our primary data holder that is determined by the hashed key. If Quorum is needed then the primary data holder contacts the quorum number of machines and ensures that all can write if they all can write then the machine writes to all and user is notified of success. For all a similar algorithm except now it will be written to all replicas. Furthermore, for reads a ONE read will happen from the primary and since our mechanism is bottlenecked that is all writes have to go through the primary it means that if one can be reading from the primary then it can read from the others as well and hence Quorum and All are immediately satisfied

## Cool Application

We choose our key value store to be an English dictionary application. A client can connect to the server and for a given word in the dictionary return its definition.

Below is the CDF of latencies, mean is 0.0013 seconds.



Below is the Read latency CDF, mean was .0014 sec.

