ID2203 Distributed Systems

Advanced Course

Preliminary Report

Bharath Karthikeyan, Nikhil Nadig

KTH Royal Institute of Technology

Problem Statement

The goal of the project is to implement and test - a simple partitioned, distributed in-memory key-value store with linearisable operation semantics.

Infrastructure Definition

The below elements define the infrastructure of our project:

- Support for partitioned key-space among nodes in the configuration
- Assigning of nodes and partitions to replication groups.
- Partitions being distributed over the available nodes such that all value are replicated with a specific replication degree δ.
- GET, PUT and CAS requests which will comprise of read, update and compare between key stores.
- Ability to broadcast information within replication groups and possibly across.
- We will also drive towards the linearisable property for the system

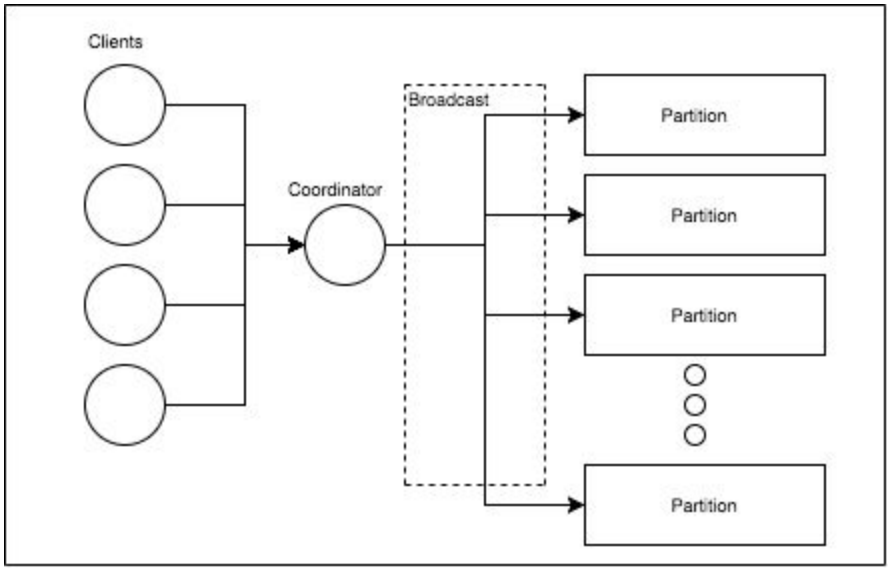
System Details

- Our system is designed to be a **crash-stop** model, i.e. a process is executed correctly, but may crash some time. After the process crashes, it never recovers. Once a process crashes, it is considered faulty and does not process anything. A correct node never crashes (liveness).
- **Perfect links** are point to point links that are characterized by three properties. The reliable delivery property together with no duplication property ensures that every

message sent by a correct process is delivered by the receiver exactly once, if the receiver is also correct.

- **Eventually Perfect Failure Detector** (EPFD) is an abstraction that allows to detect a faulty node eventually which means that the failure detector can perform wrong in the beginning but will eventually be accurate. The assumption for a EPFD is that the time it takes to become accurate is unknown.

Architecture



- During initialization, the coordinator node waits for incoming connections from the server till the degree of replication that has been defined to support the system's functioning is satisfied. It then assigns nodes to the defined partition groups.
- In each partition group, nodes communicate with the nodes in the group and replicate data so that when a node fails the data assigned to that node is not lost. For this to work we need to make sure that when a node fails and the number of nodes in the

group is less than the threshold of replication degree, a node is added to that partition group or the existing nodes join other partition groups and rearrange the data so that data is evenly distributed.

- In order to support this kind of dynamic rearrangement of nodes, the coordinator node needs to use an eventually perfect failure detector (EPFD) to make sure the partition is balanced and there are nodes added to partitions which have lesser number of nodes and divide partition groups that are overpopulated into smaller groups.
- Multiple operations are performed on the KV store which should not cause conflicts and each operation must be in such a way that atomicity is preserved. When a client request is sent for a GET/PUT operation, the coordinator receives this requests and re-routes this request to the partition group responsible for this data based on its key. On receiving a response from the partition group, the coordinator redirects the response to the client.

Conclusion

A highly available, fault tolerant key value store can be created which can tolerate upto (N-1)/2 nodes of failure. We learn the various techniques and algorithms involved in creating and maintaining a distributed key value store.

For sustainability of the nodes we will proceed with the reconfiguration technique to overcome message loss through failing nodes and also at the same time ensuring all correct node operations are still linearisable.