Building Distributed Key- Value Pair Using Zoo-Keeper

BY

RAJAT TIWARI

VTH SEMESTER

Mentors: **Dr. Dinkar Sitaram**

Dr. K V Subramaniam

Introduction

A distributed data store in a computer network where information is stored on more than one node, often in a replicated manner. It is used for distributed database management where users store information on a number of nodes, across computer networks.

Simply put a store that is capable of storing data indexed by a key

1. *Key is a string of characters*
2. *Value is a string of characters*

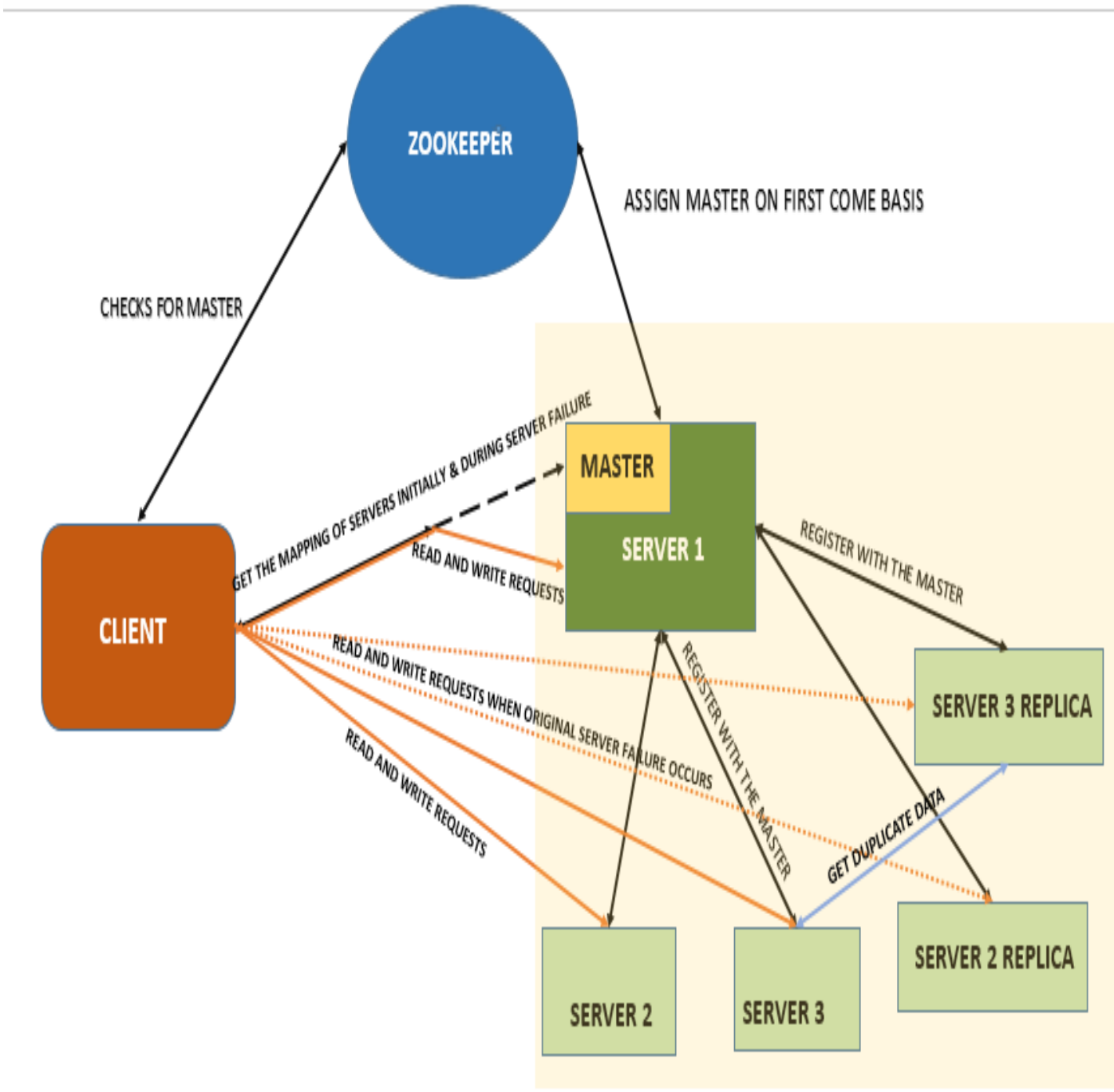
   Value is a JSON object

ALGORITHM/DESIGN

1. Creation of a Client and Server java files using socket programming.
2. Start ZooKeeper and creation of Master ZNode
3. Check for the presence of Master using ZooKeeper
4. In case of absence of Master:
   - Self-declare as Master, the current ZNode.
   - Setting the cluster status to INITIALIZING.

- Wait for a fixed amount of time, for other servers to come up.
- Now, set cluster status to READY
- Send start signal to all existing servers
- Further, send server id back and total number of servers.

5. In case of Master's presence:
   - Register with the Master
   - Wait for the start signal
   - Store the configuration data for which a key-range server is responsible.


ZOOKEEPER
ASSIGN MASTER ON FIRST COME BASIS
CHECKS FOR MASTER
MASTER
SERVER 1
CLIENT
GET THE MAPPING OF SERVERS INITIALLY & DURING SERVER FAILURE
READ AND WRITE REQUESTS
REGISTER WITH THE MASTER
READ AND WRITE REQUESTS WHEN ORIGINAL SERVER FAILURE OCCURS
REGISTER WITH THE MASTER
GET DUPLICATE DATA
READ AND WRITE REQUESTS
SERVER 3 REPLICA
SERVER 2 REPLICA
SERVER 2
SERVER 3

SERVER OPERATION

- Clients will send requests to the server
- Server will determine request type – put, get
- Server will determine if it can process the request or the request has to be serviced by other servers
- For self-served requests – it will process the request and send back status of response
- Remote server – respond with error message

Server storage:

- Data will be stored in memory and not in any file.

SERVER REPLICATION

- Based on the server name a hash code function assigns a random number.
- The last 8 bytes of the IP Address for the server contains the above random number. Example: If the hash code returns a value 1234.

Then required value = (1234) %255

Required value = 214

Therefore, IP Address for the server will be 127.0.0.214

Its replica would be hash code return value of hashcode(*servername+r).*

HANDLING SERVER FAILURE

- Client tries connecting to server with key.
- On server failure, connects to master to get new list of keys-server mapping.
- Talks to the replica to retrieve data

EXPERIMENTAL RESULTS

- Successfully established connection between client and server.
- Successful querying of keys by the client from various servers with distributed key value pairs.
- Server failure handled, its contents replicated in a replica-server and client retrieval from replica-server.

FUTURE ENHANCEMENTS

- Handling additional servers.

REFERENCES

Reference:

- ZooKeeper: Distributed Process Coordination – Flavio Junqueira & Benjamin Reed
- StackOverflow
- Apache Foundation - Zookeeper
- Java2s.com