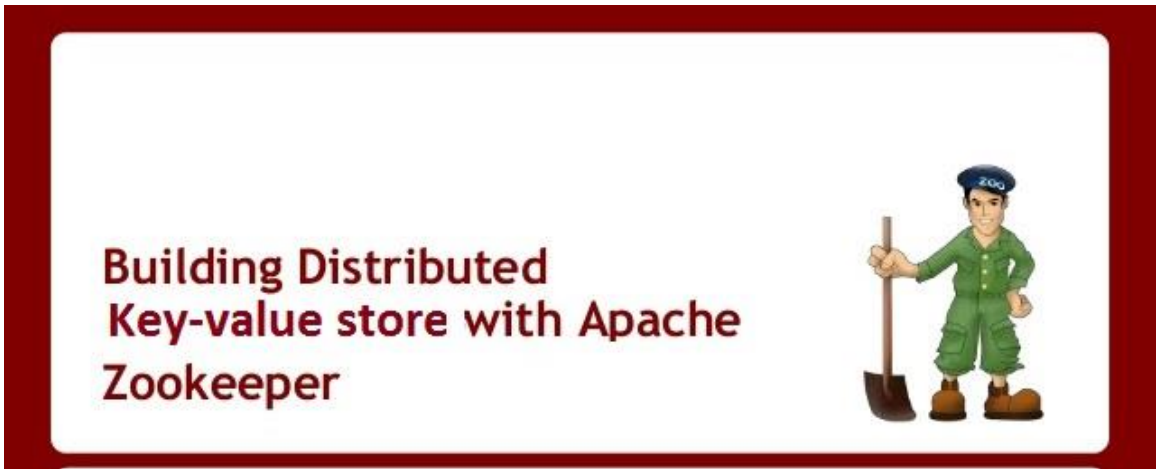



# *BUILDING A DISTRIBUTED KEY VALUE STORE*

*BIG DATA 2016 - CLASS PROJECT*

Course Title | Date


| S.No | Name | USN | Class/Section |
|---|---|---|---|
| 1 | Anmol L Patil | 01FB14ECS035 | CSE 5 A |
| 2 | Chetan G Ternamakki | 01FB14ECS059 | CSE 5 A |
| 3 | Chetan Kumar | 01FB15ECS416 | CSE 5B |
| 4 | Rachan G | 01FB14EEE032 | EEE 5A |

## Introduction

*A distributed data store is a computer network where information is stored on more than one node, often in a replicated fashion. It is usually specifically used to refer to either a distributed database where users store information on a number of nodes, or a computer network in which users store information on a number of peer network nodes.*

*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 presence:

- Register with the Master
- Wait for the start signal
- Store the configuration data on which key-range server is responsible for.

## 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.
- This random number is used to assign the last 8 bytes of the IP Address for the server. Example: If the hash code returns a value 1234.
  Then required value = (1234) %255
  Required value = 214

So, 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
- When server comes back u

## 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

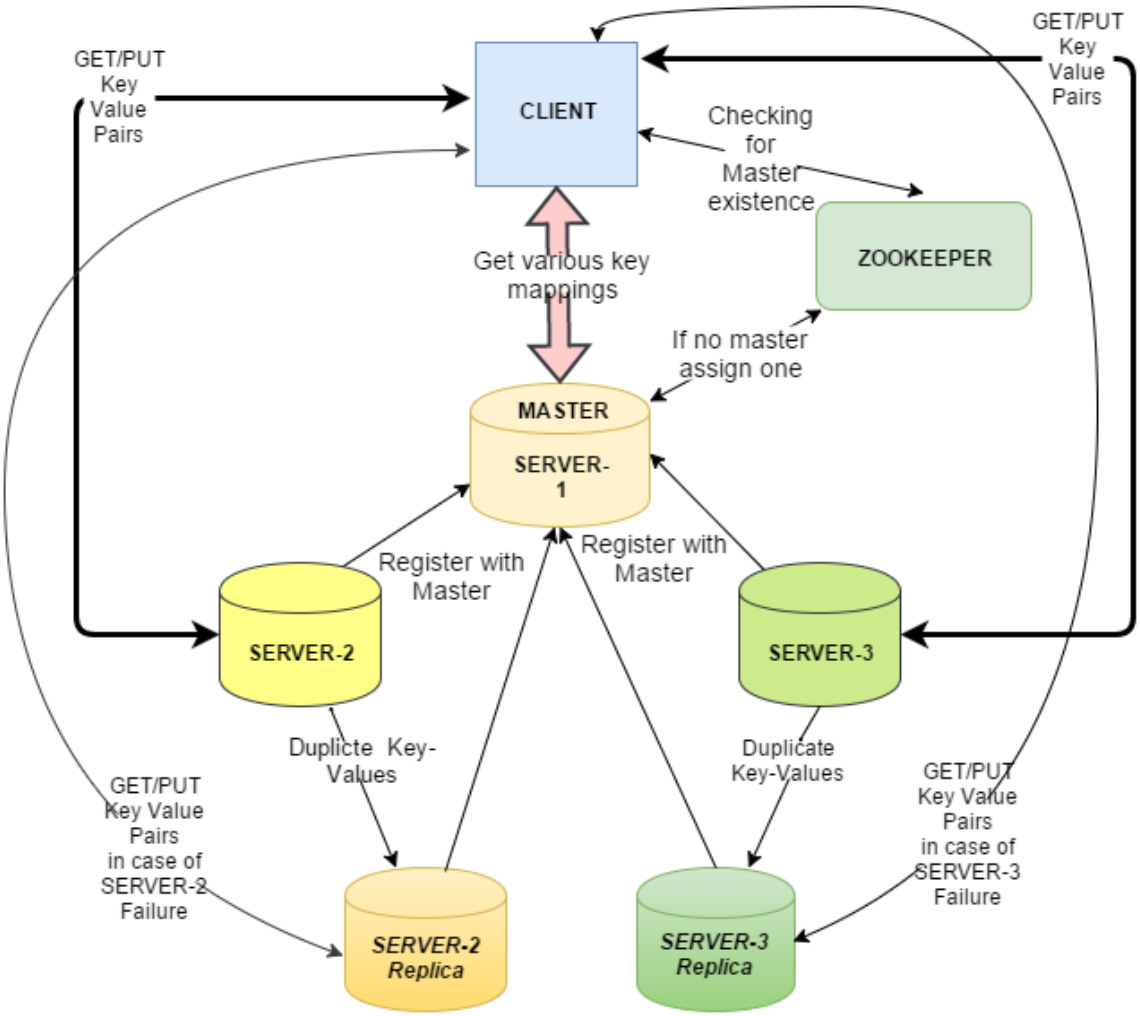
- Handling additional servers, more than three.

## REFERENCES

Reference:


- ZooKeeper: Distributed Process Coordination – Flavio Junqueira & Benjamin Reed
- StackOverflow

# Block Diagram.



## EVALUATIONS (Leave this for the faculty)

| Date | Evaluator | Comments | Score |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

## CHECKLIST

| SNo | Item | Status |
|---|---|---|
| 1. | Source code documented | |
| 2 | Source code uploaded to CCBD server | |
| 3 | Recorded video of demo | |
| 4 | Instructions for building and running the code. Your code must be usable out of the box. | |
| 5 | Dataset used for project uploaded. Please include a description of the dataset format. This includes input file format. | |