Northeastern University
CS6650 Scalable Distributed Systems | **Final Project Report**
Team Member: Zhiwei Xu, Ziliang Zhao, Sijia Zhu, Jing Liu
Session: Spring 2020
Updated: April 23rd, 2020

COVID-19 Information Aggregating System

I. Project Executive Summary

System Overview:

COVID-19 Information Aggregating System aims to collect newly confirmed reported cases of COVID-19 from states of the US and store data in key-value pairs, e.g. <NY, 3031> or <CA, 820>. A cluster of nodes shall respond atomically for operations of PUT / GET / DELETE requests from clients.

Briefly, this distributed system involves three kinds of roles.

- **Central Server**: provides *command-line-based menu* for users to interact with; stores user accounts; performs as a load balancer for info servers; logs operations of each nodes and resumes crash; enables users to add or remove info servers.
- **Info Servers:** operate requests from clients, including PUT /GET/ DELETE and store the data;
- **Clients:** could register and log in with a username and the password; add or remove participant nodes; launch requests to update data.

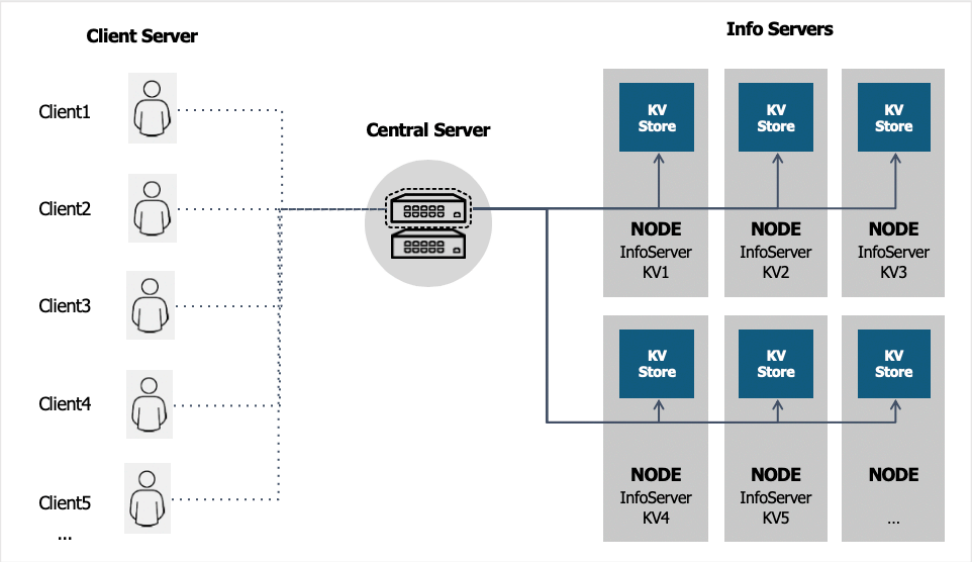

*Figure 1. Covid-19 Info System Overview**
*(*Remark: an info-server processes a request from the client directly once assigned)*

II. Technical Highlights

Functional Design Diagram:

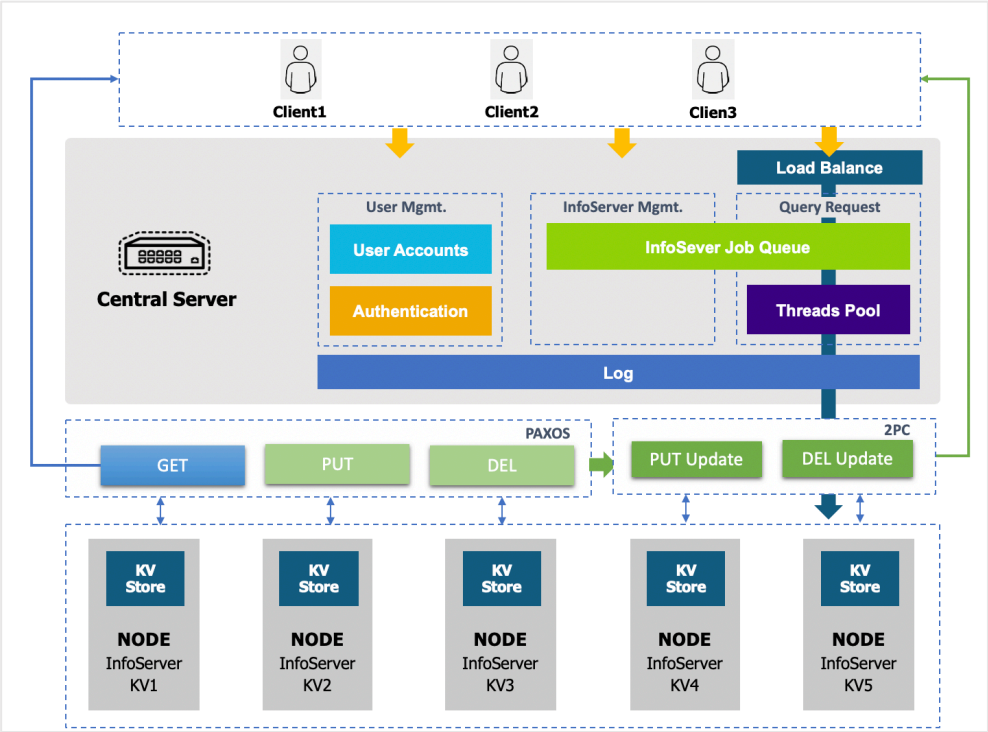

Figure 2. System Functional Diagram

Design Points:

- The **central server** provides **four main services**, including
    - **User Management:** user credentials with a pair of username and password can be stored and read from the *Account file*, so that clients could register and log into the system.
    - **Info Server Management:** according to the menu, clients could add an info server up to the *max of 10* or remove an info server from the *InforServer Queue*. The minimum number of the nodes is *5* so that the majority for Paxos consensus could always be attained.
    - **Query Requests:** clients are able to query and update COVID-19 cases of 50 states in the US. With data from real-time COVID-19 reports populated into info servers, users could connect to a *randomly assigned* participant node for a query.
    - **Logs:** the central server stores a *daily log file* to record and state recovery following a crash.

- **Info Servers** are designed for **processing requests** and **storing** COVID-19 data in pairs.
    - New created info server would be added into the thread pool for tasks.
    - The **Paxos** has been built for reach consensus from **concurrent updates with different values**, and the **Two Phase Commit Protocol** algorithm intends to **ensure synchronization for the data writes.**

Technical Impressions:

- **Basic Infrastructure**
  - This distributed system has been built upon **RPC** with RMI for server-client connections as well as **group communications**.
  - The *master-slave model* has been used to conduct **self-stabilization**

- **Central Server**
  - Central server provides interfaces of *`signup`, `login`, `addInfoServer`, `removeInfoServer` and `search`* for services.
  - User credentials includes *username and password as a pair of key-value*, stored in the `*Account*` file for further authentication.
  - A *thread pool* with core size of 10 and keep-alive servers of 3 has been created for the tasks. Info Servers with the name of "KV#" are *queued* in a list.
  - The central server *load balances* the queries to an info server *randomly*.
  - `Log4j` library has been imported for creating and storing customized logs. Each step of operation has been recorded as in the figure 3. The logs are stored in the file of `CentralServer/2020/04/22.log (e.g.)` in a daily basis.

```
[2020-04-22 21:57:53.053 ERROR Utils:35] - User name already exists!
[2020-04-22 22:10:31.031 INFO  Utils:28] - Server: localhost/KV5 is ready
[2020-04-22 22:10:32.032 INFO  Utils:28] - A new server initiated!
[2020-04-22 22:13:21.021 INFO  Utils:28] - A new server initiated!
[2020-04-22 22:13:21.021 INFO  Utils:28] - Server: localhost/KV6 is ready
[2020-04-22 22:13:29.029 INFO  Utils:28] - A new server initiated!
[2020-04-22 22:13:29.029 INFO  Utils:28] - Server: localhost/KV7 is ready
```

Figure 3. System Logs Screen Shots

- **Info Server**
  - Each Info servers use a *Map as* **in-memory database** to store COVID-19 data.
  - Info Server implements the put/get/delete operations, replication consensus and concurrency control.
  - All info servers are stored in the InforServer Queue, enabling **distributed mutual exclusion**
    - Central server maintains Queue for node server to enter critical section
    - Info server sends requests REQUEST to central server, wait for permission to enter critical section
    - Node send RELEASE to central server
  - The ***Paxos*** has been built for reach consensus from **concurrent updates with different values**.
  - ***Two Phase Commit Protocol*** algorithm intends to **ensure synchronization for the data writes**

- `*ReentrantReadWriteLock*` has been imported for read and write lock to conducts query operations. Write locks are used for PUT and DELETE requests while read lock for GET.