

## Eventual Consistence Spec

The task is divided into two functions: 1) client.c 2) server.c

Other useful files:

- 1) Makefile is provided to quickly compile all the client and server functions and delete all the programs and txt files.  
Run **make** to create all executable codes.  
Run **make clean** to remove all the executable and txt files
- 2) Config file is provided to denote the delay range and server information

## Brief introduction to server functionalities

### Instruction to run server.c

- 1) Run Make to create server compiled code
- 2) Server takes five arguments: program name, config file, ID, W, R, Number of  
For example **./server config 1 2 2 9**  
Note: Since the number of server is 9, 9 terminal windows are required to create 9 independent servers. Or, the program is seg fault.

### Main function

Initialize all the parameters including mutex, server list, config parser, etc  
It also starts a new pthread to each connection.

### ProcessClient

First the server receives requests from client, the communication protocol is constructed according the MP spec.  
Second the server receives requests from other server. If it is get, it returns the local copy with key and value. Note, the value is fusion with system time when the value was stored. If it is put, update the local copy of value if exist, otherwise append the data to the data structure

### Inter\_server\_channel

This function is to deal with the string passing request among servers.

### Write\_file

Create a file and check track of client request as specified by the MP spec

Other functions are self explanatory, please see function for more detail

## **Brief introduction to client functionalities**

### **Instruction to run client.c**

- 1) Run Make to create server compiled code.
- 2) Client programs takes four arguments: program name, config file, Client\_ID, NumberofServer

For example, ./client config 1 9

Note: Since the number of server is 9, 9 terminal windows are required to create 9 independent servers.

### **Main function**

Connect to the server with given server ID. Send get, put and dump request to connect server.

If the connected server is down, reconnect function is triggered to reconnect to a new server with a higher server ID.

### **Reconnect\_server**

The function is to reconnect a new server only when current connected server is down.