**serverv5.c:**

This is a concurrent server that can handle requests from multiple clients simultaneously.

This program gets a file name from the client, checks if the file is available. If the file is available then sends the file to client. If it is not able to serve the file, then it sends the corresponding error number to client. This program will run on every node in the background.

Path on node: */home/umkc\_sc9v9/rtt/serverv5.c*

**startClient.sh:**

This script starts the client application (*clientv10*). This script takes *nodesList.txt* as input file which contains hostnames of all nodes. This script is present on all the nodes.

**startStopServer.sh:**

This script is used to start or stop the server application (*serverv5.c*). This script will take one command line parameter *start* or *stop*. This script reside on all the nodes.

**stopClient.sh:**

This script stops the client application (*clientv10.c*). Checks for any process related to client is running and kills it. This script is present on all the nodes.

**clientv10.c**

This program is used to connect to server and request for service. After the connection is successful, client (*clientv10.c*) receives the requested file from server and closes connection after completion of file receiving. This program reside on all the nodes. This program collects 5 files from server and also calculates the time taken to collect these files. This program logs the following details.

*Source* – Hostname of the source node from which the file is being collected.

*Destination* – Hostname of the destination node on which the file is being collected.

*Filename* – Name of the file that the client collected.

*Time Stamp* – Time at which the file is collected.

*Delay* – Time taken by client to collect the file from server.

Syntax of log file: *RTT\_log\_<destination node name>.log*

Path on node: */home/umkc\_sc9v9/rtt/client/clientv10.c*

**server.sh**

This script reads the nodes list from *nodesList.txt* file and calls *startStopServer.sh* script on each node to start or stop the server application. This script takes start or stop as command-line parameter. We run this script on our local machine.

**downloadRTT.sh**

This script collects all log files from all nodes and merges them into one file. This script takes *nodesList.txt* file as input.

**loadDownloadDetails\_bkp.sql**

This sql file is used to load the log file that is generated from *downloadRTT.sh* script into IGOD database.

**client.sh**

This script calls *stopClient.sh* and *startClient.sh* on the remote node to stop or start client application. It takes *nodesList.txt* file as input. And this script takes start or stop as command-line parameter. We run this script on our local system.

**We have created below tables in our database IGOD:**

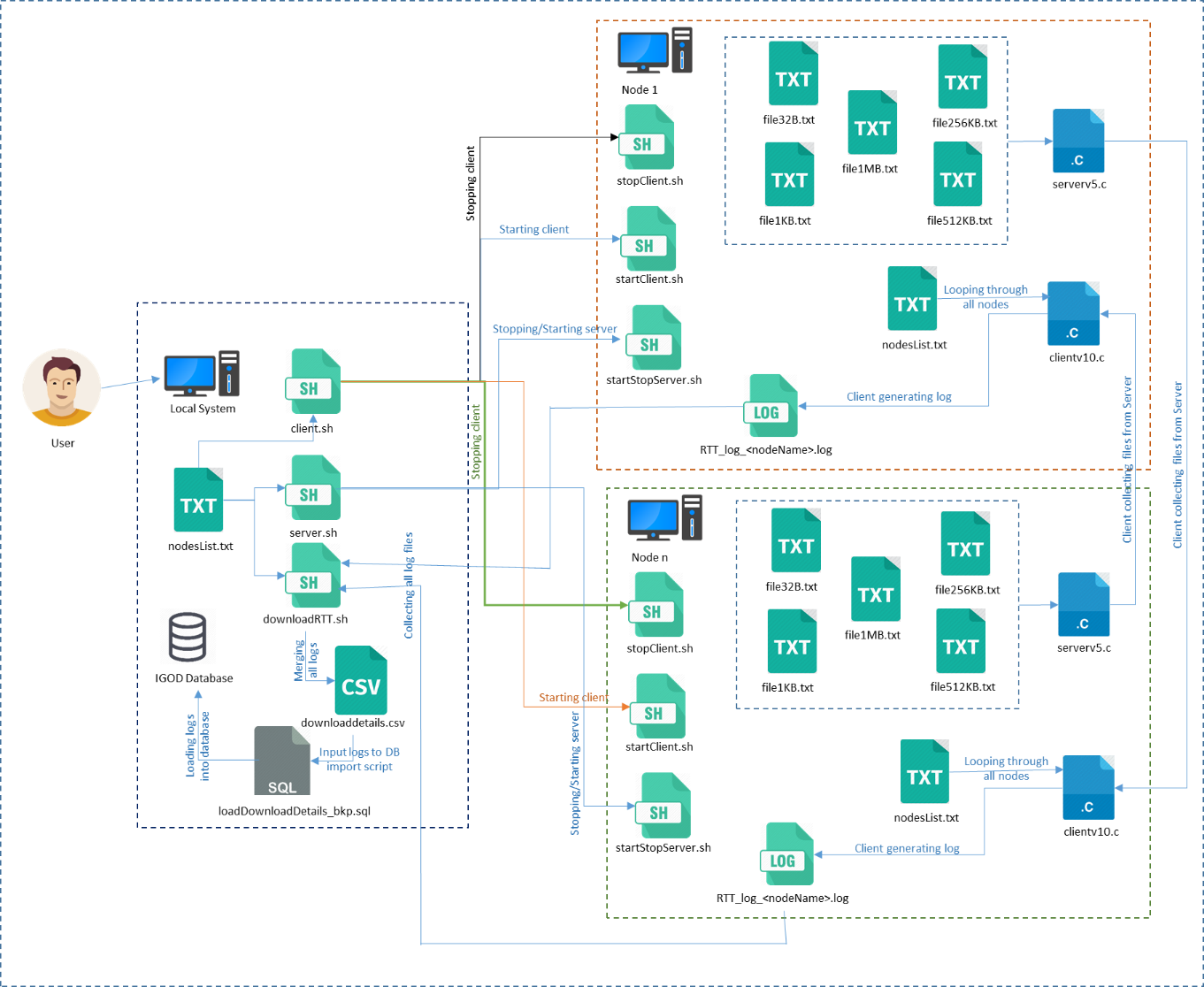
**downloaddetails\_bkp:** which stores source node, destination node, file name, start times, end times, delays and the average delay. This table will be updated daily using *loadDownloadDetails\_bkp.sql* script file.

**distofnodes:** This table stores the distance between any of the two nodes.

**nodes:** This table stores the sitename, hostname, latitude and longitude of all nodes.

**DDR:** This is a view that is generated from *downloaddetails\_bkp* and *distofnodes* tables to stores average DDR for all nodes of all files. This view will be updated whenever *downloaddetails\_bkp* table is updated.

**Below diagram show an overview of where each program/script resides:**

****