COMSM2001-Coursework 2 Report

There are two sockets for handling connections on control port and data port respectively. We need to design a 'data port solution' which is able to handle multiple simultaneous connections. Roughly, **one pthread_mutex** (*lock*)**, one condition variable** (*handle_condition*) **and three global variables** (*worker_con_fd[], handling_worker_count* and *global_run*) are used to solve the problem of thread synchronization. Brief introduction is following.

Data port

In order to handle multiple simultaneous connections, we create a 'worker thread pool' including four workers which allows four connections to be handled at the same time. Here, *handling_worker_count* is used for counting the number of workers dealing with connections right now. Before accepting a new connection in data socket, we need to check whether *handling_worker_count* is below four. In other words, the main function want to know if there is an available worker. If no workers are currently free, we will wait the connection until one becomes available by using pthread_cond_wait to wait the broadcast of *handle_condition*. Otherwise, the main function will accept the coming connection, assign this connection fd to one worker (put this connection fd into *worker_con_fd[workerID]*) and send a broadcast of *handle_condition*. Each worker thread cannot read a new command until it has been delegated a connection by looking at its element of *worker_con_fd*, which is also signaled by a broadcast of *handle_condition*. Obviously, after finishing the task and closing the connection, each worker thread has to decrease *handling_worker_count* by one and send a broadcast of *handle_condition* to 'tell' the main thread that it is free now!

In addition, all those four worker threads will be finished if *global_run* equals 0, since it means that a SHUTDOWN command is executed. The value of *global_run* will be set after control socket received a SHUTDOWN command. Furthermore, all the key/value operations and global variables are protected by mutex (*lock*) because they are not thread-safe.

Control port

Poll function is used for checking whether a connection available on data and control socket. If there is a connection on control socket, we will tackle it first. It means that the data socket cannot accept new connections when a connection on the control port is open. If the control command is SHUTDOWN, the main thread will close both data and control socket until all the current connections on data socket have been closed. And then a broadcast of *handle_condition* will be sent, which can make sure that all the worker threads will be finished even if they are waiting for a new connection fd, since each worker thread will check the value of *global_run* after they received the broadcast signal..