# 5105 Project 3 Report

Group: QiuChen Yan, Hoon Kim, Francis Liu

## How to use

Step1 start Tracking Server ‘./tsvc’, this will give you ts\_ip, ts\_port and ts\_ping\_port.

e.g.

My IP is :128.101.34.203 🡨 ts\_ip

Listen on port 8888 🡨 ts\_ip

Ping on port 9999 🡨 ts\_ping\_port

Step2 start each peer ‘./peer <pid> <dir> <ts\_ip> <ts\_port> <ts\_ping\_port> <latency\_matrix\_file>’, where pid is a number from 1 to 100 (depend on how big the latency\_matrix\_file is generated, the example latency.txt included in our submission has a latency matrix of 100), dir is the directory of this peer, latency\_matrix\_file is generated by latency\_mat\_gen <maximum # peers>

Step3 go to look at Tracking Server’s output, find filename and peer number, download your interested files from peer side, ‘download <filenameA> [filenameB] [filenameC] …’.

## Optional functions implemented

1. Multiple download
2. Periodically call UpdateList
3. Peer notify Tracking Server of other Peer down

## Design

Tracking Server

When started, server first read 2 port numbers from file cofig.txt, which are the port for listening requests and the port for replying ping, respectively. Server starts 3 threads when it is on. They are listen thread, ping thread and check thread.

Listen thread accept find, update and register from peers. To begin with, a peer first sends a register request, containing its id, IP address and port number for download. While received this request, server while first check whether this peer is in its list. If it already in, it must be that this peer has crashed and now restarted. Server will remove the old information of this peer and add it again in terms of newly received register request. If it’s not in list, server will simply add it to list. After register, peers will first send an update request to add all its files to server’s file list. Peer also update it file list every time it has downloaded a new file. To find whether somebody has a certain file, peers can send find request to server. Server will reply the id, IP address and download port of every peer that shares this file. If nobody has this file, server replies a special byte as a signal of fail finding.

Ping thread replies every ping message from peers that are on the peer list of server, and ignore the received ping message from unknown peers. The content of ping message is the global id of sender peer.

The goal of check thread is to check whether each peer in list has ping server recently. In more detail, it run periodically to check how many times has a peer fail to send a ping message. And if a peer fail to ping server for 3 times, server will regard this peer as a crashed one and remove it from peer list. The cycle of checking is 5 seconds in default.

Peer selection Algorithm

Peer support concurrent download. Whether a peer is trying to download a file from another peer, or another peer is trying to download file from me, both are count as workload. Upon receiving Find() result from Tracking Server, a Peer will first check if itself is in the list, if it does, it further check if its local file is correct, in which case it will not download from peer. After checking, a peer will call GetLoad from each of the Peers in returned list to get their latency. This operation could be failed due to peer down. In which case, this peer will report this to Tracking Server. The algorithm is to estimate the cost of downloading a file, by multiplying p2p latency and workload of the other peer. Also, we record peers that have been failed before, either due to file corruption during network transfer or other temporary failures. These failures case will also count as cost when choosing from which peer to perform download with. The total retry factor is adjustable, (although set as 3 in our code).

## Fault tolerance

When server crashed, a peer can immediately detect that if server doesn’t receive the response of a ping. Then peer will keep sending register request until server has recovered and replied its request.

When a peer crashed, server has 2 ways to detect that. The first one is ping. If a peer didn’t ping server for more than 15 seconds (by default), server will know that it has crashed. The second one is by duplicated register request. If server received 2 register requests with the same id number, it will know that this peer has crashed and now recovered.

## Special Test case

ST01: terminate peer during download, check fault tolerant, peer that receiving file should automatically try another peer.

ST02: Support many download simultaneously.

ST03: manually delete existing file from the nearest peer, check a peer can download the file from other peers.

## Result analysis

Bellow is the diagram of the relationship between number of peer s hosting a file and the time downloading this file.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| number of peers | 1 | 2 | 3 | 4 | 5 |
| download time | 33.513929s | 17.146398s | 9.796822s | 7.196216s | 1.778424s |

According to the graph above, time cost decrease as more peers share a file. This is because the range of choosing a peer with lowest workload and latency become bigger as more peers hosting a file. Sometimes time cost may also increase if the peer who requests a file have a very high latency toward all peers sharing this file.