

# Project 3: Simple xFS

---

Group member:

Chendong Wang [wang9235@umn.edu](mailto:wang9235@umn.edu)

Xuefei Li [li001008@umn.edu](mailto:li001008@umn.edu)

## Component description

---

- Node

```
Node(int, int, int, string);
\\ The Initialization function of Node class, the input is the ports used
by the Node and the name of the Node.
string checkip();
\\ Used to check the ip of the node itself
void onListen();
\\ A thread function used to listen to the request from other nodes
int Connect();
\\ A function used to connect the trackServer
int SetServer(string, int);
\\ A function used to set the IP and Port of trackServer
string listen(int);
\\ A function wrapped from recvfrom for convenience
int sendall(int, char *, int *);
\\ A function wrapped from sendto for convenience
int FileList();
\\ The function used to send the file list to the track server.
int getfd();
\\ The function used to get the protected node socket fd
names getFileList();
\\ a function used to get file list from the node directory
int update();
\\ The function used to update the file list to the track server.
names Find(int, string);
\\ The function is used to find the nodes that upload the request file
string GetLoad(int, string);
\\ The function is used to get load from a certain node
int Download(string);
\\ The function is used to create a download thread
void uploader(string, int, string, string);
\\ This function is used to create an upload thread
```

```

int handleUpload(string, int, string, string);
\\ This function upload the requested file to the target IP and port
int handleDownload(string);
int downloadfrom(int,int, string, string );
\\ The above fucntions are given file name as input, then call the find
and getload function to choose the most efficient uploader and download the
file from there.
int handleheartbeat();
\\ A thread function used to maintain heartbeat between node and
trackserver, it can detect if the trackserver is down and handle the recovery
process.

```

- Peer selection

```

int prior_value(int load, float latency){
    return (load+1) * latency;
}

```

This function takes the load and the average latency of the target uploader as input, then the prior value of the uploaders are naturally derived from the overall latency of the uploader. The average latency can be computed and stored from the configuration file. The load value can be retrieved from the target with Getload function.

- trackServer

```

class trackServer
{
private:
    struct sockaddr_in _server_addr, _client_addr, _tmp_addr;
    int _server_socket_fd, _client_socket_fd, _slen, _port;
    char _buf[BUFLLEN];
    typedef vector<string> names;
    map<string, names> fileLists;
    names tmpfiles;
    names hbList;
    string sIP;

public:
    trackServer(int);
    \\ Init the track server with a certain port
    string checkip();
    int OnConnect();
    \\ A function used to handle connection from multiple nodes
    string listen();
    int sendall(char *, int *);
    \\ The above two fucntions are used to wrap the sendto and recvFrom
function
    int handleCreate(string);

```

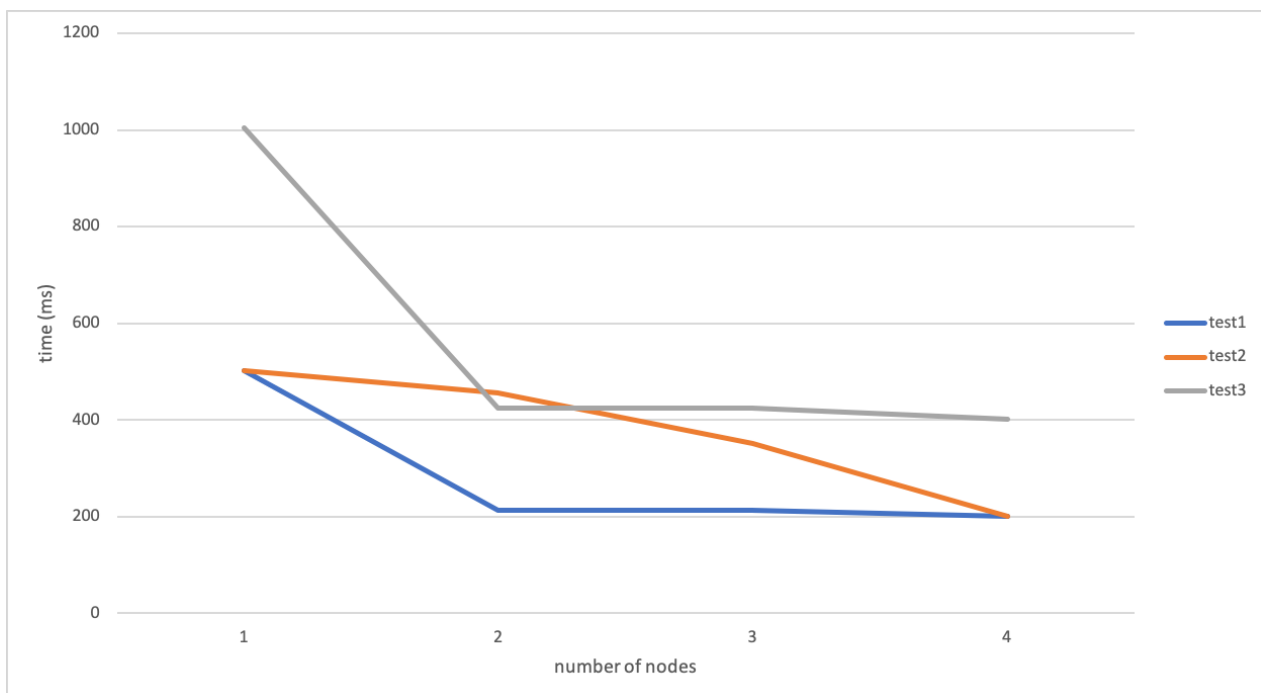
```

int handleFind();
int handleUpdate(string);
\\ The above function are used to handle certain request from nodes.
int heartbeat();
\\ The thread used to send heartbeat to all the nodes periodically
};

```

## Analysis

From the graph below, we can see the average file download time will decrease as the number of peers hosting a file increases. If a node hosting a file is added, our node will Getload from it, then consider the load and latency of the peer. If it is better than any of the previous node (lower latency, low load), we will choose this peer as the uploader and send Download request to it, therefore the download time will decrease. If it is not a good choice, we will skip it and send to the best peer, download time will at least remain the same. Above all, the average file download time will decrease.



## Usage

```

cd pj3_directory
make clean
make

```

- Up the trackServer

The trackServer will be up automatically at Port 4999, which can be modified in the servermain to meet different need.

```
./trackServer
```

- Up the Node

This executable will take multiple arguments with init.

The first argv is the main Port of the Node, like 5000, 6000, 7000

We recommend to set the port with interval more than 100 since a node can use more ports after the main port such as 5001, 5002, 5003, ... etc. to handle multithread downloading , uploading and trackserver related communication

The second argv is the name of node, like node1, node2, node3, which should be the same as defined in the configuration file.

The third argv is the address of trackserver with format like 127.0.0.1:4999.

```
./Node 5000 node1 134.84.182.30:4999
```

```
li001008@cse1-kh4250-30:/home/li001008/Desktop/pj3 $ ./TrackingServer
System IP Address is: 134.84.182.30
```

```
li001008@cse1-kh4250-30:/home/li001008/Desktop/pj3 $ ./Node 5000 node1 134.84.182.30:4999
System IP Address is: 134.84.182.30
load config
3node1,node2,node3,
Enter command: █
```

```
li001008@cse1-kh4250-30:/home/li001008/Desktop/pj3 $ ./Node 6000 node2 134.84.182.30:4999
System IP Address is: 134.84.182.30
load config
3node1,node2,node3,
Enter command: █
```

```
li001008@cse1-kh4250-30:/home/li001008/Desktop/pj3 $ ./Node 7000 node3 134.84.182.30:4999
System IP Address is: 134.84.182.30
load config
3node1,node2,node3,
Enter command: █
```

- Command:
  - find(filename): find all peers hosting the file
  - getload(peer): get the load of certain peer after find
  - download(filename): download the file from the best choice of peers according to prior\_value
  - sync: update local file list to tracking server

## Testing description

- Download
  - Single file download
    - Just type in the command and the file you want.

```

li001008@cse1-kh4250-30:/home/li001008/Desktop/pj3 $ ./Node 5000 node1 134.84.182.30:4999
System IP Address is: 134.84.182.30
load config
3node1,node2,node3,
Enter command: download
Which file? test.txt
Try test.txt
Enter command: Find file test.txt at: 134.84.182.30:6001
Find file test.txt at: 134.84.182.30:7001
Port 5003 Download from 134.84.182.30:7001
Total file size:50729 Byte
=====
7 64 %

```

- Multi files download

We test by first download a large file (test.txt), before it complete, enter download again to download a small file (cnm.txt), we can see that "Download cnm.txt done!" is printed first, showing that the two downloads are processing in different threads simultaneously, small file is completed first, although it started later.

```

Enter command: download
Which file? test.txt
Try test.txt
Enter command: Find file test.txt at: 134.84.182.30:6001
Find file test.txt at: 134.84.182.30:7001
Port 5003 Download from 134.84.182.30:7001
Total file size:50729 Byte
download
Which file? cnm.txt
Try cnm.txt
Enter command: Find file cnm.txt at: 134.84.182.30:6001
Find file cnm.txt at: 134.84.182.30:7001
Port 5004 Download from 134.84.182.30:6001
Total file size:8 Byte
content: fuck you
Download cnm.txt done!
Download test.txt done!

```

- Getload
- Find

```

li001008@cse1-kh4250-30:/home/li001008/Desktop/pj3 $ ./Node 5000 node1 134.84.182.30:4999
System IP Address is: 134.84.182.30
load config
3node1,node2,node3,
Enter command: find
Which file? test.txt
Find file test.txt at: 134.84.182.30:6001
Find file test.txt at: 134.84.182.30:7001
Enter command: getload
Which node? 134.84.182.30:6001
node2:0

```

- Update

```

li001008@cse1-kh4250-30:/home/li001008/Desktop/pj3 $ ./Node 6000 node2 134.84.182.30:4999
System IP Address is: 134.84.182.30
load config
3node1,node2,node3,
Enter command: sync
Enter command:

```

The server will show:



```
Updating file list from 134.84.182.30:6000
test.txt
cnm.txt
```

- Fault tolerance
  - Corrupted file
    - Uncomment line 728 & 739 in node.cpp to modify a byte of the message to send.
    - Download a file as above
    - Result:

```
li001008@cseel-kh4250-30:/home/li001008/Desktop/pj3 $ ./Node 5000 node1 134.84.182.30:4999
System IP Address is: 134.84.182.30
load config
3node1,node2,node3,
Enter command: download
Which file? test.txt
Try test.txt
Enter command: Find file test.txt at: 134.84.182.30:6001
Find file test.txt at: 134.84.182.30:7001
Port 5003 Download from 134.84.182.30:7001
Total file size:50729 Byte
The file was corrupted, try download from another node.-----] 100 %
Total file size:50729 Byte
```

- Peer crashes
  - crash the sender node while the download is performing.

```

■ Result:
Total file size:50729 Byte
timeout, try another peer=====] 70 %
Time out, try another node.
No available node. Download fail

```

- Server crashes
  - crash the server after the nodes have connected

- Result:
- 

```
li001008@csel-kh4250-30:/home/li001008/Desktop/pj3 $ ./Node 5000
System IP Address is: 134.84.182.30
load config
3node1,node2,node3,
Enter command: Server timeout, maybe down now, wait for recovery
Server timeout, maybe down now, wait for recovery
Server timeout, maybe down now, wait for recovery
```

- The node should be blocked and wait for the server to recover
- Then up the server again
- Result:

[illegible]

■