**Peer to Peer Parallelism File Transfer**

CS 550 Programming Assignment 4

Abheek Mondal

A20438046

CS 550

Professor Sun

09 April 2023

**Overview**

In this assignment, you are required to change PA2 to adding parallelism in data transfer within a single file. In PA2, a peer node downloads a file from a single node. However, a file could be broken into a set of fixed size chunks while transferring. Therefore, a peer node could download these chunks from multiple peer nodes in parallel and then make the original source file. You will need to check the integrity of the data for each chunk to ensure the file is the same as the original one.

**Code**

I implemented the following algorithms:

1. central\_indexing.py: Monitors all the peers, keeps track of what information each node has.
2. peer\_node.py: Is the peer node, communicates with CIS to register its files and download other files from other peer nodes.

central\_indexing.py

The CentralIndexingServer class consists of the following methods:

* **\_\_init\_\_(self, host, port):** Initializes the server with the specified IP address and port.
* **load\_file\_data(self):** Loads the file data from a persistent pickle file, if it exists.
* **print\_pickle\_file(file\_name):** Prints the contents of the given pickle file.
* **save\_file\_data(self):** Saves the current file data to a persistent pickle file.
* **handle\_client(self, conn, addr):** Handles individual client requests and sends the appropriate response.
* **start(self):** Starts the server and listens for incoming connections.

The if \_\_name\_\_ == "\_\_main\_\_": block is executed when the script is run as the main program. It initializes a Central Indexing Server instance and starts the server.

peer\_node.py

The main functions of the PeerNode class are as follows:

* \_\_init\_\_(self, host, port): Initializes the PeerNode with the given host and port.
* create\_folder(self): Creates a unique folder for the PeerNode to store its files.
* user\_interface(self): Provides a simple user interface to interact with the system.
* register\_files(self): Registers the files in the PeerNode's folder with the Central Index.
* start\_server(self): Starts a server to listen for incoming file requests.
* run\_server(self): Handles incoming connections and starts a new thread for each connection.
* handle\_client(self, conn, addr): Processes file requests from other peer nodes.
* request\_file(self, file\_name): Queries and downloads a file from other peer nodes.
* list\_local\_files(self): Lists the local files in the PeerNode's folder.
* download\_file(self, file\_name, file\_size, peer\_nodes): Downloads a file from multiple peer nodes in parallel.
* check\_file\_integrity(self, file\_path, original\_size): Verifies the integrity of a downloaded file.
* send\_handshake(self): Sends a handshake message to the Central Indexing Server to establish a connection.

There is a generic structure of the deployment script I used. It is a .bat extension file whose only purpose was to open all the terminals at the same time so I could test the download times and record them. However, it is unique to my environment, so it will need to be modified to different environments.

Chart, scatter chart

Description automatically generated**Data Results**

|  |  |
| --- | --- |
| **N** | **Time (s)** |
| **2** | 0.04 |
| 3 | 0.02 |
| **4** | 0.02 |
| 5 | 0.02 |
| 6 | 0.008 |
| 7 | 0.02 |
| **8** | 0.008 |
| 9 | 0.008 |
| 10 | 0.008 |
| 11 | 0.01 |
| 12 | 0.008 |
| 13 | 0.012 |
| 14 | 0.008 |
| 15 | 0.008 |
| **16** | 0.008 |

As we can see there is a near exponential drop for the time taken vs. the number of peers which already had the document. However, after a certain point, regardless of the number of processes that had the file, the speed for the download did not decrease. I would love to look into why. I hypothesize that it is due to the fact that 1MB is already a small enough file to download, and after a while no matter how small the file was split up, if it was split into any more pieces, it would just be inconvenient to communicate with all those peers to download the file. In fact, I believe that it would slow down the process.

**Conclusion**

My program worked, and I listed all the improvements I can make, and the trade offs that I did make in the design doc. It is titled PA4 design doc.