Saurabh Sanjay Malkar net id-sxm170231
Jitendria Justin net id- jxj172430

Advanced Computer Networks (6390.001)

Project Report

Aim: In this project we will be creating a simplified version of Inter planetary file system(IPFS).

Description: In this project we will create a p2p web network consisting of p2p nodes. A p2p node is made up of 3 modules as follows:

 Content Provider: Content provider publishes the content in its storage, so that the other device can access the content. It achieves this publishing the content to other using the PUBLISH<filename>command. By publishing the content the publisher shares (metadata, endpoints) with others.
o Metadata: hash, content-length, content-type.
o Endpoint: IP address:port pair hosting that content.

There are two types file we are going to share in the scope of this project
o Web pages(text/ HTML)
o Images(png)

 Content Tracker: This module keeps track of the metadata. Whenever a node publishes some content, it is the job of content tracker to store this information in its cache(storage). Exchanging this information with other node is also a responsibility of content tracker. Once the other nodes receives this information it stores it in its cache. Whenever a client receives request for some content it provides the metadata and endpoint to the client.

 Client gateway: Client gateway enables the client (Browser) to access the content from the p2pweb. Whenever it receives a request with for a particular content it looks up with the content tracker and gets the information about the node hosting the file(ip address and port number) and using this information it request the content from the node hosting it gets the content and provide it to the client.

It uses http to access p2pweb. If the content is not found it replies with a 404 error.

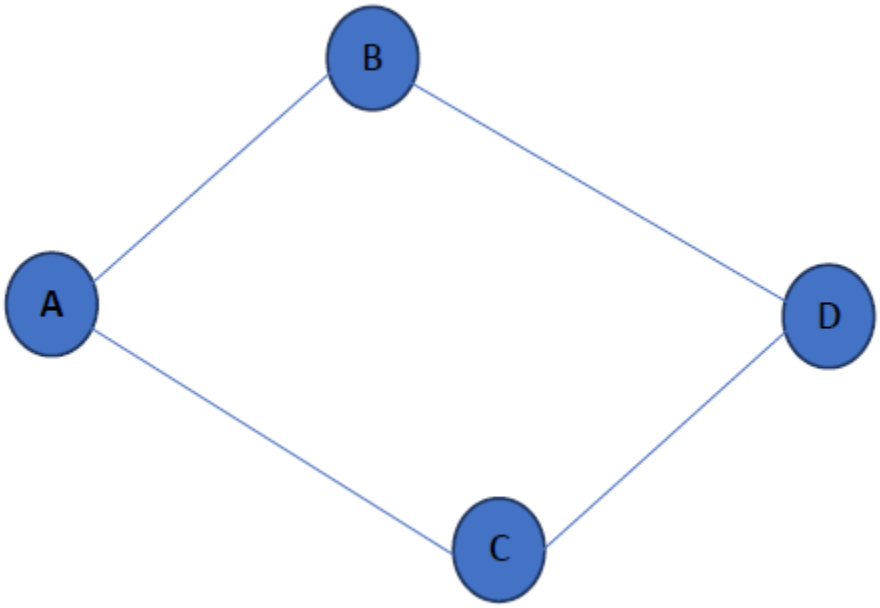
Saurabh Sanjay Malkar net id-sxm170231
Jitendria Justin net id- jxj172430

Capabilities of a node:

- PEER <peer-hostname><peer-port>: This command is used for establishing a direct connection with a node if it doesn't already have a connection.
- PUBLISH <filename>: When a node wants to publish some content it uses this command to do it.
- UNPUBLISH <hash>: When node do not want host the some content anymore it uses this command to remove it from web.
- SHOW_PEERS: This commands shows the peers connected directly to the node.
- SHOW_METADATA: This command shows the metadata stored in the cache of the content tracker.

Example:

Suppose we want to create the following topology using Four nodes. To do this task following operations are performed:
First ,4 nodes are created.
Node A has to be a peer to both Node B and Node C , Peer command is used on Node A and give Node B's IP address and port number as the input. This command creates a permanent connection between Node A and Node B. Similarly, Node C as added Node A's peer.
Node D is a peer of both Node B and Node C. They are added a peer same as on Node A.
The following topology is created.



There are two main jobs a node does: Publish/ Unpublish a file, Request for the file to other nodes, In this case either the .txt file or .png file. Here in this example let node A publish the data (metadata) and node D be requesting for that data.

Publish:

Saurabh Sanjay Malkar net id-sxm170231
Jitendria Justin net id- jxj172430

1. Suppose A wants to publish a file, it uses the command PUBLISH<filename> to publish the content to the p2p web.
2. At this point at A, the content provider of A will generate a hash value using the SAH1 algorithm and store this metadata which is :< hash value, content type , content length> in a global table, it is table which contain metadata of all the published file by all nodes, and send this table to its peers. The peers update their own global table and send it to their own peers. By doing this, it is made sure that all the nodes on the network knows that Node A has published a file . Whoever has this hash value can access this file from A. And the content tracker keeps the track of this global table on p2p web and is responsible for the exchange of this metadata & IP endpoint between the nodes.

Request :
1. Now, if there is a client connected to node D, which wants to access to the file (eg : .png file) from p2p web. It uses SHOW_METADATA command all the file's metadata (Hash, Host IP address, Host port) is displayed. Since client wants the file published by Node A it sends the request for the file from the browser using http and the hash value of the desired file.
2. At this point, the client gateway gets the information of the node hosting this file from the content tracker of D, which already knows the metadata and IP end-point (that is the node hosting this file) since content tracker is responsible for the exchange of metadata and IP end-point.
3. Now, the request for file by D is fulfilled by the client gateway, since it knows the endpoint, which is hosting the file. It forms a connection with node A using the IP address and port number associated with file, since D is not a peer of A and then fetches the content from the endpoint hosting this content and serves to the client with appropriate http headers. If the file is not found, then it replies to the request with ERROR 404.When the request is fulfilled, the connection between D and A is terminated. The connection is not terminated if the two nodes are peers.

Unpublish:
Suppose A wants to unpublish a file from p2p web, It uses the command UNPUBLISH<filename> to unpublish the content from p2p web. Node A updates it table by removing the meta of the file. Then it sends this updated table to its peer and they follows similar steps as the publish command to make sure every knows on the network that Node A is no longer hosting the data.

Contribution of group members:

Saurabh Sanjay Malkar :

- Implementation of commands.
- Implementation of peering.
- Implementation on DC machines
- Debugging
- Report

Jitendria Justin :

- Implementation of Multiple client Server
- Server functions
- Implementation of http server commands

- Debugging
- Report

Challenges faced:

1.During the course of this project we decide to perform it using Python, being from a Non-Computer Science background we were not familiar with it. But still we decide to use it , this project provided us with a good opportunity to learn a new language and use it to implement something so we get a hand on experience. Now, while learning the language we came across various hardship of a totally new concept. Still we went through all of it and were actually able to successfully complete it.

2.We were unable to handle multiple connection to a node, after having some insight of python we were exposed to the class Threading in python which allows us to do multiple things at a time, So we created a thread that handles the multiple connection to a node.

3.While handling the announcement made by node, we were facing problem like to send periodic announcement to peers, then we came up with a solution, that to send announcement only when someone publishes or unpublish some file

4.our project works on the assumption that, nodes are peered first, then the nodes start publish/unpublish the file, Whose metadata then is circulated in the network by the content tracker, So a new node in the network gets the announcement information only when one of the node publishes something

Conclusion:

Our project works on the assumption that the nodes are peered first and then the commands are operated, It works great on dc machines! It provides a file for a http request from the client like browser, if the file is not present in the network then an “ERROR 404 File Not Found” is displayed onto the browser.

Saurabh Sanjay Malkar net id-sxm170231
Jitendria Justin net id- jxj172430