Peer to Peer （P2P）
Sharing Application

COMP480 report

Ujjwal Kumar
Instructor's Name : Joseph Vybihal
November 24, 2020

Abstraction :

Peer-to-peer (**P2P**) computing or networking is a distributed application, architecture that partitions tasks or workloads between peers. Peers are equally privileged, equipotent participants in the application. They are said to form a peer-to-peer network of nodes.[1]

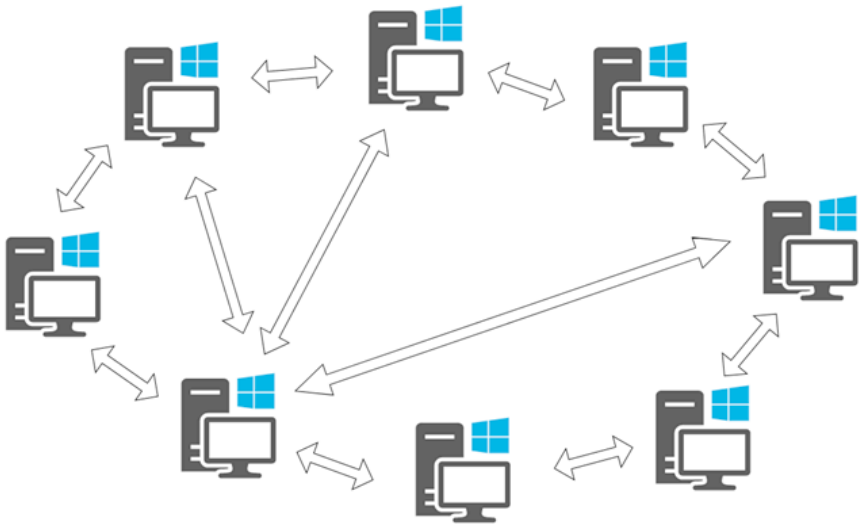
Peers make a portion of their resources, such as processing power, disk storage or network bandwidth, directly available to other network participants, without the need for central coordination by servers or stable hosts. Peers are both suppliers and consumers of resources, in contrast to the traditional, client-server model in which the consumption and supply of resources is divided. Emerging collaborative P2P systems are going beyond the era of peers doing similar things while sharing resources, and are looking for diverse peers that can bring in unique resources and capabilities to a virtual-community thereby empowering it to engage in greater tasks beyond those that can be accomplished by individual peers, yet that are beneficial to all the peers.

While P2P systems had previously been used in many application domains, the architecture was popularized by the file sharing system Napster, originally released in 1999. The concept has inspired new structures and philosophies in many areas of human interaction. In such social contexts, P2P refers to the egalitarian social networking that has emerged throughout society, enabled by Internet technologies in general.

In the comp480 Course, I mainly focused on studying P2P distrusted systems and Later focused on developing a P2P file sharing application that takes it's implementation abstraction from BitTorrent protocol where central server called Tracker is responsible for storing meta-info regarding all the peers in the network and is used by peers of the overlay network to share files between each other.

This report shed the light on how useful the P2P systems can be in terms of data privacy as P2P doesn't rely on some central Data service and the implementation detail of the project.

Introduction :



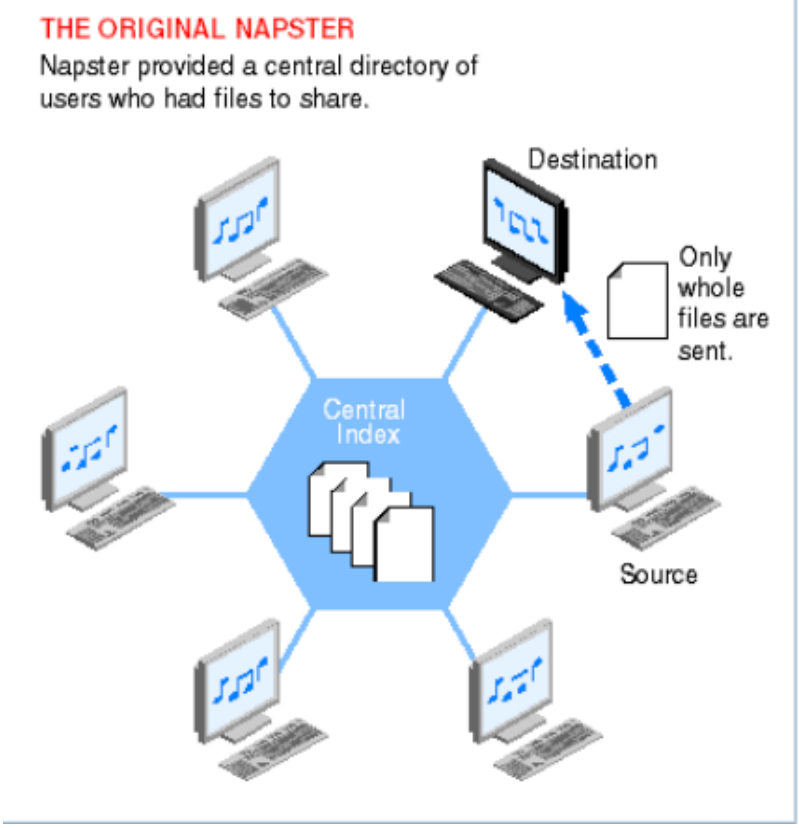
- Peer to Peer network is a distributed infrastructure in which many computers systems (nodes) are connected in order to share resources. There are many ways to create the P2P network , we can either connect the nodes by physical link or by using virtual overlay networks.
- Each Device (computer, mobile, … etc) is consider to be a peer in the network and behaves as node that can share its content and also have access to available resources in the network.
- In its simples form , a peer to peer (P2P) network is created when two or more PCs are connected and share resource without going through a sparse server computer. A P2P network can be an ad hoc - connection couple of computers connected via a Universal Serial Bus [3] to transfer files.

Architecture :

- A peer to peer network is designed around the notion of equal peers nodes simultaneously functioning as both clients and servers to the other nodes in the network. This model of network arrangement differs from the client-server model where communication is usually to and from a central server. A typical example of a file transfer that uses the client-sever mode is the "File Transfer protocol " (FTP) service in which the client and server programs are distinct. The clients initiate the transfer and the server satisfy these requests. [4]

- In a P2P network when one peer makes a request, it is possible that multiple peers have the copy of the requested object. Now the problem is how to get the IP address of those peers. This is a decided by the underlying architecture supported by the P2P systems.
- There are different architecture for P2P networks that allows a single client / peer to get information about all the peers that are requesting its resource and then can directly connect to the peers in the networks for P2P file transfer. Mainly the address of a peer in the network is its IP address + port number at which it's running it process for listening incoming connection.

- **Centralized Directory :**
  - Similar to the client -sever model because in this P2P architecture we need a single or distributed servers to maintain the directory service (meta-information about files and peers in the network).
  - When a new Peer /client wants to join the overlay P2P network it simply registered itself (using it's IP address) with the central Directory (also known as trackers in some architecture) then tracker will register this new Peer in the network in it's index file with the list of available at this peer. [5]
  - Tracker sends heartbeat message to the peers to check for the availability or node failure and then updates it's index file accordingly
  - This type of architecture was first popularized by the Napster for music files distribution.
    - [4] Napster, launched in 1999, was the first widely available implementation of peer to peer model. A central database contained information about all the music files send by member You would search for a song from this central server, but to download it, you would actually connect to anther online user/peer in the network and copy the file from them. In turn, once you had that song in your Napster library, it become available as a source for others on the network.

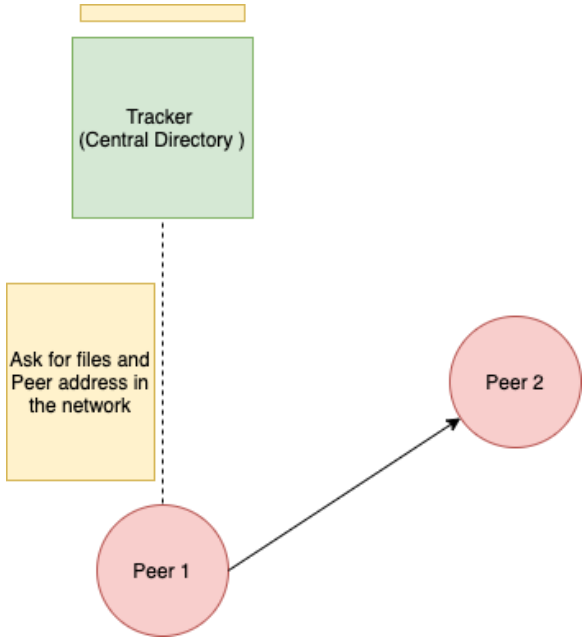


- But like any distributed architecture a single point of failure is a huge issue and as in the case of this model , the central tracker is over node that if fails can bring down the whole network. This issue can be resolved by replicating this tracker over different servers and in case of failures peers can simply use the second replica meanwhile the first original / main tracker server recovers from its failure.
- Whenever a new peer X wants to request a file available in the network it will send the it's query to the server and since the central server (tracker) has all the information about peers, it returns the IP address of all the peers having the requested file to the peer that request the file.
- Now that peer X has all the information regarding address of peers contains the chunks of given file. It can request the file directly from the peers in the network and " voila Peer X has the requested file".

|| SuperNova ||

A P2P network for file sharing

- SuperNova is the P2P file sharing network that I created for course comp480.
- It is heavily influenced **Centralized Directory** P2P file sharing architecture with some limited features and scalability.
- SuperNova architecture contains one Tracker , Which acts like a Centralized Directory, that is available to all the peers in the network.
- Same as Centralized Directory Architecture, Tracker is responsible for storing all the meta-information regarding the available peers and shared files in the network. For database (storing meta-information) SuperNova uses simple Json file format (peers.json). This gives peers and trackers to rejoin the networks after leaving at any given time.



Registration :

- Whenever new peer wants to join the network it starts it's process and tries to connect to tracker. When the tracker connects to the Peer using TCP connection, Tracker will look for the peer in it's peer.json file and check if the peer name already exists or not and if it does it will simply send the "*welcome*" message to the Peer by acknowledging that the peer is returning peer in the supernova network.
- But if peer was not found in trackers peers.json file that means peer was never registered in the network and therefore tracker will simply ask the Peer to give its user name or will assign an iterative username (peer_i). After the peer is being registered in the network , tracker will send "*welcome*" message to the peer and now the connection between tracker and that peer has been established.
- After the peer register it self it will send the details of its config files.
- Detail message contains at which port and IP address Peer will be listening for incoming peer connection and the all the files name Peer has made available for sharing amongst other peers.

After this Tracker will save all the information regarding that peer in its **peers.json** file and peer will store all the details (connection related and process related) in its **config.json** file

After the Peer is up and running it will start a separate thread for listening the incoming connection in case some other peers wants the file from its own directory.

peers.json for server when there are two peers in the network :


```
"PEER_1": {
    "files": [
        ".DS_Store",
        "video.mov",
        "virus_replication.cpp",
        "client1.pdf"
    ],
    "listeningIP": "localhost",
    "listeningPORT": "53784"
},
"PEER_2": {
    "files": [
        ".DS_Store",
        "DistributedComputing512.pdf",
        "521Project4image.png"
    ],
    "listeningIP": "localhost",
    "listeningPORT": "53786"
}
```

```
"peerListeningIP": "localhost",
"peerListeningPORT": 0,
"sharedDIR": "test",
"trackerIP": "localhost",
"trackerPort": 45000,
"user": "PEER_1"
```


config.json for peers when the connection is established between peer and tracker.

Peers Menu :

- Peers have access to some limited number of features.
  - 1. Requesting peers details in the network and files available in the network
  - 2. Requesting a file in the network from another peer in the network
  - 3. Additional feature includes providing new shared directory to the user.

- upon startup Peer specify the directory it wants to share in the network. All the files in that network are available to other peers in the network.
- At the moment there is a limited refresh view of the files in the network that means if new file is added in directory by the user then refreshing system doesn't work as expected. This issue can be resolved in later updates using event based programming and installing a watch under the directory and notifying the process in any changes is made to the directory and immediately refreshing json config files accordingly.
- Peers can ask tracker to send the list of the files available in the network
- After that Peers can choose whatever file it wants from other peers.

File Sharing :

- whenever the peer wants to request a file from another peer. It sends a message to tracker to retrieve the IP address and Port of the peer that contains that particular file.
- When Peer get the information of the another peer that contains the file it creates a separate thread and tries to communicate with the other peer and ask for the file. And if the other peer still has the file it accepts the request and send the file to the peer who requested it over TCP connection.
- After the file requested is received, this peer sends a last command "thanks" specifying the other peers that file transfer was successful.
- Note : Because Peer has separate thread running for incoming connection it listens for incoming connection and as soon as a peer tries to connect to this peer it responds accordingly.

Future work :

- SuperNova is still limited in features and error handlings.
- In future I will try to implement better concurrency control so that multiple peers can connect and transfer multiple files a time thus increasing the network throughput.
- One disadvantage of single **Centralized Directory** architecture for P2P file sharing is that there is possibility that tracker can fail and whole network is brought down in an instant. It's possible that Centralized tracker can recover and continue where it left but this means all the other peers in network can't communicated with each other to share files.
- One possible solution I though about was replicate the tracker using **Paxos Algorithm**.
- So whenever a tracker is down the other one can take it's place and whole network can continue as before.
- Another improvement I can think of in terms of throughput, is by importing the whole project to **Java** and using **Apache Zookeeper** to coordinate between different element of the network.
    - For example a Zookeeper ensemble will allow multiple peers to connection to the replicated trackers (in this they will be Workers servers) and in case a tracker is down other peers can still connect to the replicated trackers without ever knowing that there was failure node in the network.

- Another step I am looking forward to implement is to make my own replicated servers using Raspberry pi and use this P2P service to automatically sync files between two computers (in some specify folder). I can be working on two different computers and if I simply wants to send file to another computer I add the file in the shared Dir and sync files across all my personnel computers connected to the superNova network.This gives us a cheap and modular file sync service across multiple computers.

I have learned so much about distributed systems in past few months and I am so excited to on some new upcoming projects in the same field.

There are infinite possibility with this project. I will try to continue this project where I left it for next few months because I think that will give me huge opportunity to learn about the topic even further and go in details regarding advantages and disadvantages at more deeper level.

Welcome to SuperNova Network ...

- **bibliography :**

[1] - https://en.wikipedia.org/wiki/Peer-to-peer

[2] - https://www.computerworld.com/article/2588287/networking-peer-to-peer-network.html

[3]- https://www.computerworld.com/article/2588287/networking-peer-to-peer-network.html

[4] - https://www.makeuseof.com/tag/p2p-peer-peer-file-sharing-works/

[5] - https://www.digitalcitizen.life/what-is-p2p-peer-to-peer/