Decentralized

Trust needed

Presentation from simple to complex

https://cs.uwaterloo.ca/~m2nagapp/courses/CS446/1195/slides/Deliverable2_4.pdf
https://cs.uwaterloo.ca/~oadesina/slides/D2_Peer_to_Peer.pdf
https://www.student.cs.uwaterloo.ca/~cs446/1171/Arch_Design_Activity/Peer2Peer.pdf

https://www.techopedia.com/definition/454/peer-to-peer-architecture-p2p-architecture
https://www.sciencedirect.com/topics/computer-science/peer-to-peer-architectures
https://stackoverflow.com/questions/12660754/java-peer-to-peer-thread-model-everyone-wait-for-job
https://wiki.p2pfoundation.net/Supernode_P2P_System
https://pdfs.semanticscholar.org/8d9f/e736de5c987f9139061b77c6bcd0c99deb27.pdf
Where the delegation model has a boss thread that delegates tasks to worker threads, **in the peer-to-peer model all the threads have an equal working status. This could be made**

Using Paper for group presentation

1. Centralized … failed
    a. Delay,not responding, network failed
    b. Problems with it and how p2p can be better
2. P2P -> exchange files
    a. Somebody died
    b. File can not be sent fully and properly
3. P2P → 1 … 10 EXAMPLE send chunk
4. P2P -> personal connection failed,
    a. J -> A -----> J->L->A
5. Personal bandwidth not enough, except one of them
    a. Supernode
6. Blockchain example
    a.
7.
8.
9.
10. Non-functional property
    a.
11. Concern -> efficiency, security

Dog Analogy

I lost my dog! How could I find him/her again?

Presentation order:

1. Act 1 (Intro to the scenario)
    a. Jialin call irvine
    b. Irvine calls Johnny + Alex
    c. Jialin calls Alex
2. Jialin - Server, p2p difference
3. Johnny - scalability, maintainability (advantage)
4. Irvine - availability (disadvantage)
5. Act 2 (Still finding one person gives up)
    a. Johnny gives up
6. Johnny - robustness (advantage)
7. Act 3 (Dog found, privacy lost)
    a. Alex finds dog
    b. Tells me
    c. Tells irvine
    d. Irvine tells Johnny
    e. Privacy
8. Jialin - topological (disadvantage)
9. Johnny - security and privacy (disadvantage)
10. Irvin - Ways to P2P
11. Alex - Attacks in P2P
12. Alex - Applications that use P2P

Peer to Peer Architecture

Group Name: PitchPerfect
Group Members: Jialin Shan (j6shan), Alex Lai (x7lai), Johnny Gao (z53gao), Irvine Yao(b6yao)

Definition

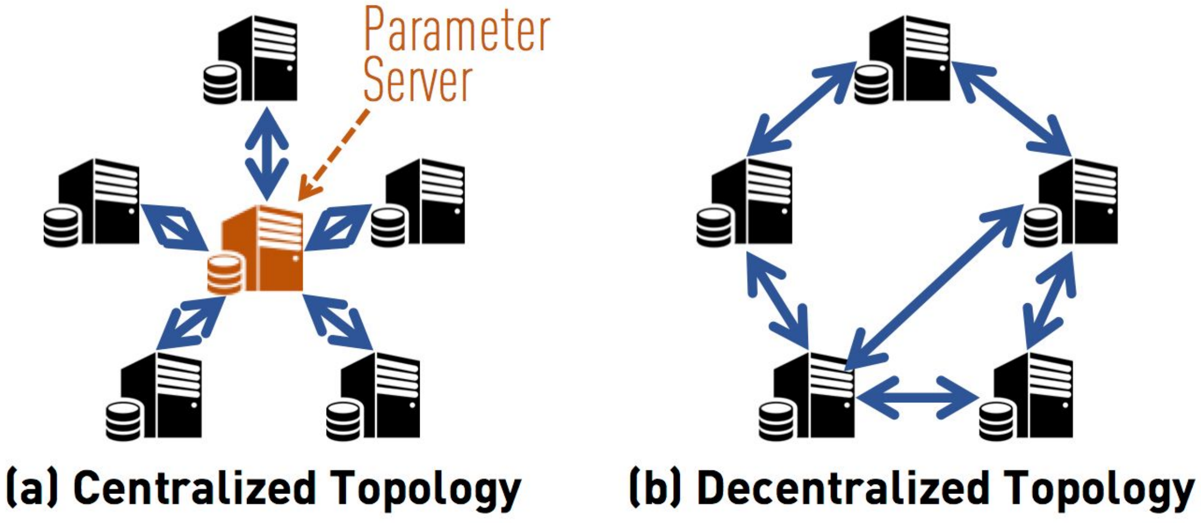
Peer to Peer (P2P) Architecture is a way of structuring **distributed systems** so that they can share a workload or tasks between them.

Overview

Each system (referred to as a **peer** or a **node**) in the architecture has equal privilege and similarity in contributing power. They can share resources such as processing power, disk storage, files, and network data without the need of a central system.

The architecture is **decentralized** meaning that a task is dispersed into peers instead of being handled by a single server. And each peer can enter or get out of the P2P "network" with freedom.

This architecture is commonly used in content delivery networks (e.g. Netflix), and peercasting (e.g. Ace Stream).



Centralized vs. Decentralized
(Source of picture: https://www.ibm.com/blogs/research/wp-content/uploads/2017/12/topology.jpg)

Advantages (NFP):

- **Robustness**
    - Decentralization can prevent single point of failure. If a peer fails, an alternative way will be recalculated by the system. Thus it can be regarded as a **highly robust** architecture.

- **Scalability**
    - This architecture is **highly scalable** because the power of computing increases exponentially as more peers are introduced to the network.
    - There can be no server. Or only little information is demanded from the server when there is a component that stores the indexing of other components in its charge.
    - Comparing with client-server architecture:
        - In client-server architecture, the total transferring speed is limited by the server's upload speed.
        - In peer to peer architecture, the server only needs to upload one file with pieces to separate clients. The clients as peers can spread their piece copy over the entire network. In the end, all peers are able to get the entire file.

- **Maintainability**
    - There is no maintenance in p2p architecture since the system doesn't have a server. In other words, p2p is self organized.
    - The program employing the P2P architecture doesn't need to maintain the personal devices in the network.

Disadvantages (NFP not in P2P)

- **Availability**
    - If there are no peers on the current network then there is no benefit gained from the network (Same processing power or in the case of a torrent, you cannot get the file). In a client server architecture the server is paid for and should be available in most of the time.
- **Security/Privacy**:
    - Anyone connected to you in the network knows your IP
        - Thus there is a sense of trust that must exist to all peers in the network
          OR
        - The data that the initial person is sharing should be information that they are willing to give away to malicious third parties
    - Vulnerable to many types of attacks
        - Poisoning
        - Tracking Attack

Inherent Problems

- **Topological**:
    - When a peer needs to find out information but don't know who has it, it might potentially have to go through a lot of other peers causing useless traffic.

Ways to P2P

Before we talk about ways P2P is implemented we should take a moment to understand ways routes can be made in between peers in P2P. P2P could be structured or unstructured.

- In an unstructured P2P architecture peers connect to each other randomly.
- In a structured P2P architecture peers connect based on an algorithm.

Decentralized

- This is the purest form of p2p where both discovering peers as well as sending information are all done on a peer to peer basis.

Centralized

- This strays from the pure P2P. This type of P2P sacrifices the robustness of decentralized P2P for efficiency. With centralized P2P all peers are indexed and stored in a central location (A server). Then whenever a peer wants to send any information to another specific peer they can ask the server where they are and send the information correctly. This makes it so no searching information is passed through multiple peers for no reason.

Attacks in P2P

DoS and DDoS

- One attacker among peers can insert false entry in indexing server's routing table so as to control peers to query for a targeted address. The target will then be waiting for enormous acknowledgements. This is an illustration of Dos attack.
- NOTE: This type of attack is done WITH the P2P network TO another network

Query Flooding Attack

- One attacker can broadcast nonsense information to peers. Then peers cannot see true information among nonsense information. One possible solution is for the indexing server to add this peer to blacklist. But carefully disguised false message can easily be spread throughout the p2p network in short time.
- NOTE: This type of attack is done to block the p2p network

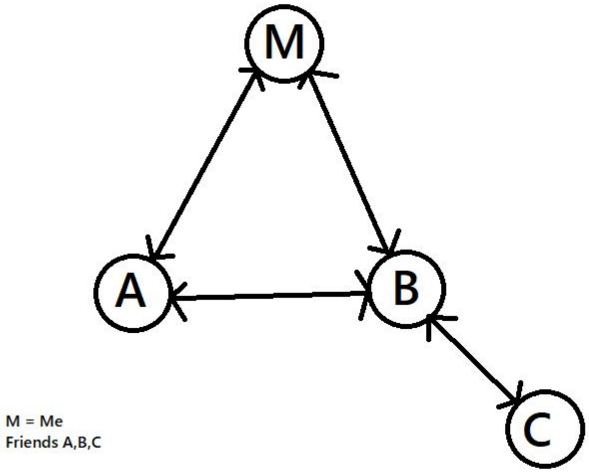
Applications that use P2P

- VoIP (Skype)
    - In the case of skype they use a special type of P2P network called supernode. A node is promoted that basically acts like a host.
- Torrent (BitTorrent, Centralized P2P)
    - Trackers would be the servers that index the peers so peers can find each other.
- Decentralized Currency (BitCoin)
    - Shared computation in a decentralized network
- Distributed File Systems (OFFSystem)

Analogy

In the scenario where you are trying to find your lost dog you have two choices:

- Call animal shelter control services
    - This selection is similar to a client server architecture where you are asking the server (Which doesn't do anything for you till you do) to find your dog.
- Ask your friends (peers) to help you find your dog
    - This choice would be synonymous with a peer to peer network as your friends are just as potent as you at finding your lost dog.



After you decide you want to ask your friend (peer) to help you find dog:

- You would ask your friend A (another peer) to help you find your dog while you are searching for the dog, too.
    - This scenario shows us that in a peer to peer network, the works are being done concurrently, and works between peers are the same and equal.
- After A agreed to help you find your dog, he asks two of his friends B and C to find the dog too
    - This scenario simulated the propagation and scalability feature. More computing power joined the network meaning more work can be done quicker

While we are looking for that dog:

- Friend C decide to drop after searching for a while, the other three people are still searching and still found the dog easily
    - This scenario shows us that robustness of the peer to peer system. Failure of a peer does not mean failure of the network, and more people join the network the more resilient the network is.

When we find the dog:

- Friend B finds the dog, first he called the owner M notifying him that his dog was found and delivered the dog since turns out they know each other all along. Then he called friend A, who gave him the task. Since A still doesn't know whether owner B knows if his dog has been found or not, he called M to let him know again, despite the fact that M already have the dog.
    - Topological disadvantage since there is multiple people calling the same person to inform them of the same information.

References:

https://www.techopedia.com/definition/454/peer-to-peer-architecture-p2p-architecture

https://www.sciencedirect.com/topics/computer-science/peer-to-peer-architectures
https://stackoverflow.com/questions/12660754/java-peer-to-peer-thread-model-everyone-wait-for-job
https://wiki.p2pfoundation.net/Supernode_P2P_System
https://pdfs.semanticscholar.org/8d9f/e736de5c987f9139061b77c6bcd0c99deb27.pdf
https://www.cse.wustl.edu/~jain/cse571-07/ftp/p2p/