Peer to Peer Architecture

Group Name: Hippothesis
Group Members: Jesse Hoek (jchoek), Yifan Dai (y25dai), Elisha Lai (ahlai), Tina Zhang (myzhang)

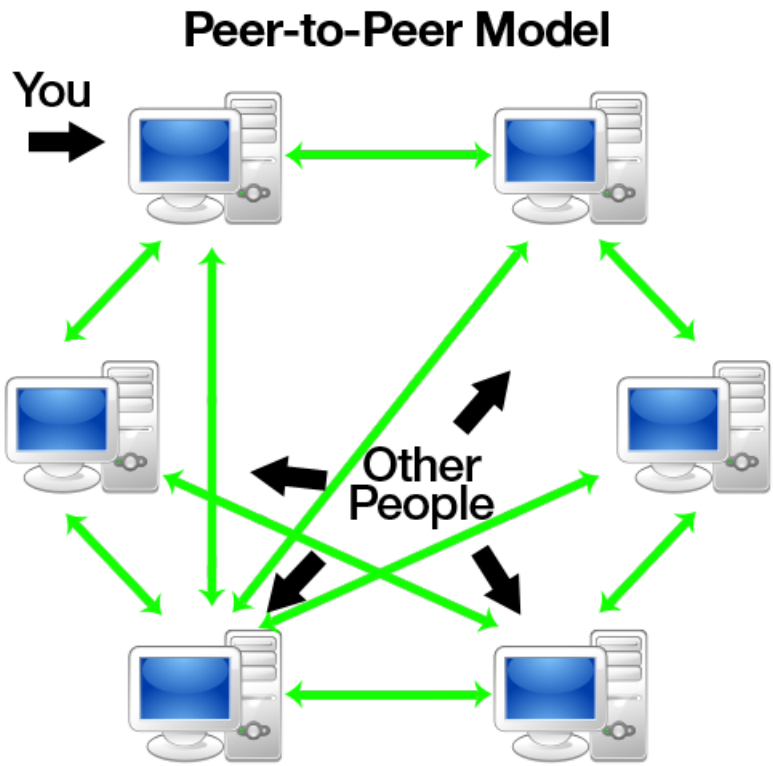
Overview of Peer to Peer Architecture

In the common client-server architecture, multiple clients will communicate with a central server. A peer-to-peer (P2P) architecture consists of a decentralized network of peers - nodes that are both clients and servers. P2P networks distribute the workload between peers, and all peers contribute and consume resources within the network without the need for a centralized server.

However, not all peers are necessarily equal. Super peers may have more resources and can contribute more than they consume. Edge peers do not contribute any resources, they only consume from the network.

In its purest form, P2P architecture is completely decentralized. However, in application, sometimes there is a central tracking server layered on top of the P2P network to help peers find each other and manage the network.

Here's a simple example of small P2P network.



Applications

P2P architecture works best when there are lots of active peers in an active network, so new peers joining the network can easily find other peers to connect to. If a large number of peers drop out of the network, there are still enough remaining peers to pick up the slack. If there are only a few peers, there are less resources available overall. For example, in a P2P file-sharing application, the more popular a file is, which means that lots of peers are sharing the file, the faster it can be downloaded.

P2P works best if the workload is split into small chunks that can be reassembled later. This way, a large number of peers can work simultaneously on one task and each peer has less work to do. In the case of P2P file-sharing, a file can be broken down so that a peer can download many chunks of the file from different peers at the same time.

Some uses of P2P architecture:

- File sharing
- Instant messaging
- Voice Communication
- Collaboration
- High Performance Computing

Some examples of P2P architecture:

- Napster - it was shut down in 2001 since they used a centralized tracking server
- BitTorrent - popular P2P file-sharing protocol, usually associated with piracy
- Skype - it used to use proprietary hybrid P2P protocol, now uses client-server model after Microsoft's acquisition
- Bitcoin - P2P cryptocurrency without a central monetary authority

Advantages/Change Resilience

P2P networks have many advantages. For example, there is no central server to maintain and to pay for (disregarding tracking servers), so this type of networks can be more economical. That also means there is no need for a network operating system, thus lowering cost even further. Another advantage would be there is no single point of failure, unless in the very unlikely case that the network is very small. P2P networks are very resilient to the change in peers; if one peer leaves, there is minimal impact on the overall network. If a large group of peers join the network at once, the network can handle the increased load easily. Due to its decentralized nature, P2P networks can survive attacks fairly well since there is no centralized server.

Disadvantages

P2P networks introduce many security concerns. If one peer is infected with a virus and uploads a chuck of the file that contains the virus, it can quickly spread to other peers. Also, if there are many peers in the network, it can be difficult to ensure they have the proper permissions to access the network if a peer is sharing a confidential file. P2P networks often

contain a large number of users who utilize resources shared by other nodes, but who do not share anything themselves. These type of freeriders are called the leechers. Although being hard to shut down is counted as an advantage, it can also be a disadvantage if it is used to facilitate illegal and immoral activities. Furthermore, the widespread use of mobile devices has made many companies to switch to other architectures. With many people using mobile devices that are not always on, it can be difficult for users to contribute to the network without draining battery life and using up mobile data. In a client-server architecture, clients don't need to contribute any of their resources.

Non-Functional Properties

- Scalability - the network is very easy to scale up
- Efficiency - the network uses available resources of peers, who normally wouldn't contribute anything in a typical client-server architecture
- Adaptability - the network can be used for a variety of use cases, and it can easily start up another network if one is taken down, which means there is no single point of failure

Further References (if you are interested)

- http://www.cim.mcgill.ca/~sveta/COMP102/P2P-4.pdf
- https://en.wikipedia.org/wiki/Peer-to-peer#Architecture
- http://www.makeuseof.com/tag/p2p-peer-peer-file-sharing-works/
- http://www.readwrite.com/2007/03/29/p2p_introduction_real_world_applications/