

Peer to Peer Architectures

Definition

The term “peer-to-peer” has come to denote a system where computers (or nodes) are autonomous and interact directly and on an equal basis with each other, sharing resources such as messages, files, storage, CPU cycles, workspaces. In contrast, client-server systems assume that some nodes have the role of providing resources and the other nodes have the passive role of consumers.

History

The idea of computers playing the roles both provider and consumer of resources has been around for a long time. The original concept of the ARPANET, the precursor of the Internet was designed with this idea in mind, as well as was the original WWW.

Theoretically, E-mail and telephony have also a peer-to-peer nature, since each user can be both an originator and a recipient of a message. However, the current Internet, Web and E-mail rely heavily on servers to provide web-content and relay messages. These servers impose a hierarchical structure and inequality, reducing some nodes to the role of passive consumers (clients). Maintaining resource-rich servers implies some times significant costs. The term “peer-to-peer” (P2P) became popular in 2000 with the simultaneous emergence of the NAPSTER music-swapping system and numerous applications aimed at harnessing the resources of networked personal computers. A P2P system is self-maintaining, once it reaches a critical mass of users.

Examples

Depending on the type of resource being shared, one can distinguish between file-sharing, instant messaging, CPU power sharing and other P2P applications.

NAPSTER allowed users to share and download music files from each other in a very efficient way. Each node (a personal computer) could both download files from others and provide files to others. After NAPSTER was forced to close in 2001, various new file-sharing P2P applications became popular, mostly based on variations of the GNUTELLA protocol, for example, KaZaA, LimeWire, eDonkey, BearShare etc.

Though these applications do not guarantee anonymity of the participating users, the entirely decentralized protocol used in these applications makes them robust (no central point of failure); and also makes it much harder to find an identifiable target for lawsuits.

FreeNet and Tapestry are also P2P file-sharing applications, based on a different protocol, which ensures full anonymity of the participants in file-sharing is.

Several instant messaging applications, like Jabber and AVAKI, and collaboration applications like Groove, among others, use P2P architecture to send messages and create shared workspaces directly among nodes, rather than routing them via servers.

A classic example of CPU power sharing application is SETI@home, which utilizes the unused computational resources of the participating nodes (the computers of users that have downloaded and installed the SETI screen-saver program) to discover patterns in radio-telescope data that may indicate the presence of intelligent life in space. The same idea is used by other P2P systems to harness computing resources for extremely time-consuming computing tasks, like analysis of DNA for cancer research.

P2P Architectures

Ideally, a “pure” P2P architecture includes no servers, i.e. the exchange happens entirely on the “fringes of the Internet”, between autonomous nodes. In reality, however, often some kind of centralization is required to ensure scalability and better performance. For example, name servers can be introduced to ensure that the nodes are independent of their DNS names (which are linked to the IP of the machine on which the node is running).

This is necessary, for example, in instant messaging systems, since often the same user will be connecting from different machines or mobile devices, so the system needs to link a node with a user, not with a machine.

NAPSTER uses a *centralized index* for the shared music stored by different users, which facilitates greatly the search in the system. A node that searches for a file sends a request to the central index to locate peer-nodes that have this resource. However, once a list of nodes that have the required resource is generated and sent to the requesting peer node, the download happens directly between the peers, without any server involvement.

SETI@home also uses a centralized architecture. Even though the actual computing tasks are executed by the participating nodes (peers), the task-decomposition and the combination of the results coming from the nodes is done by a server.

The first versions of the Gnutella protocol assumed an entirely decentralized system.

Each node sent search queries for files to its neighbours, which sent the request further to their neighbours etc, until a node generates a “hit”, i.e. sends back a message that it has the requested resource. However, this *flooding approach* generates a lot of network traffic, which slows down the network. Sometimes queries can not reach remote regions in the network. Therefore, the newer versions of Gnutella use “super-peer nodes”, which

cash search results for frequent queries that have passed through them, thus creating indexes like NAPSTER, but on a much smaller scale. Each user can choose to be a super-peer when joining the network. This emerging hierarchical structure helps to speed the traffic and to reach further in the network.

Systems like FreeNet, Tapestry, Pastry, CAN and Chord are based on a *document routing* approach. Documents are stored on peers whose names are closest (with respect to a hashing algorithm) to the name of the document. Therefore peers who contribute documents typically do not keep these documents, but documents contributed by other peers. The matching between files and peers is kept in a routing hash table. Anonymity is preserved in this way and search is very efficient. The disadvantage is that to be able to locate a document one needs to know its exact name.

Research Directions

There are a variety of problems that current P2P systems need to address varying from technical (e.g. performance, scalability, security), to human- and community-related (e.g. anonymity, trust, and participation). Most relevant to the area of HCI are the issues of trust and user participation.

The issue of trust arises due to the realization that a peer that allows access to its resources may become vulnerable. However, users may still be inclined to allow users who have behaved well in the past and have earned a good reputation to access their resources. Many researchers are currently investigating trust and reputation mechanisms in the area of multi-agent systems, e-commerce and P2P networks. In order to develop trust and reputation, however, peers, should be identifiable. Thus identity and

authentication become desirable features in a P2P community. Assuring both anonymity and identity is difficult.

Another important issue in P2P systems is user participation. Studies of Gnutella use show that most (75%) of the users are free-riders, who do not contribute or share files, but only download files shared by others (Adam & Huberman, 2000). While some have argued that this is not a problem for file-sharing P2P applications (Shirky, 2000), because it does not cost anything to replicate files, it is a serious problem for P2P applications where other resources are shared, like CPU or human attention (e.g. instant messaging, collaboration). Research on motivating users to participate through incentives, for example artificial currency used in MojoNation, and status used in virtual communities like Slashdot help to solve this problem.

Further Reading

The area of P2P systems is new and rapidly developing. It has established recently its own research forums, the P2P journal and the IEEE International Conference on P2P Computing. However, these forums focus currently mostly on the technical issues, while many of the social and HCI-related issues are discussed in other forums, like the Workshop on Agents and P2P computing, the User Modelling Conference, the E-Commerce Conference, and others. Clay Shirky (www.shirky.com) publishes popular articles on the future of P2P technologies. Also several monographs have been published on the topic recently.

Julita Vassileva

Adar, E. and Huberman, B.A. (2000). Free Riding on Gnutella. First Monday. Retrieved on January 19, 2004 from http://www.firstmonday.dk/issues/issue5_10/adar/

Barkai D. (2002) Peer-to-Peer Computing: Technologies for Sharing and Collaborating on the Net. Intel Corporation.

Golle, Ph., Leyton-Brown, K., Mironov, I. (2001) Incentives for Sharing in Peer-to-Peer Networks. Proc. EC'01, October 12-17, 2001, Tampa, Florida, ACM Press, 264-267.

Leuf B. (2002) Peer to Peer: Collaboration and Sharing over the Internet
Boston, MA: Addison Wesley.

Miller M. (2001) Discovering P2P, Marina del Rey, CA: Cybex.

Moore D. and Habeler J. (2002) Peer to Peer: Building Secure, Scalable and Manageable Networks, Berkley, CA: McGraw Hill /Osborne.

Oram A. (2001) Peer-to-Peer : Harnessing the Power of Disruptive Technologies.
Sebastopol, CA: O'Reilly Press.

Shirky, C. (2000). In Praise of Freeriders. Retrieved on January 19, 2004 from:
http://www.openp2p.com/pub/a//2000/12/01/shirky_freeloading.html

