4. Application Areas

Detlef Schoder, Kai Fischbach, Christian Schmitt
(University of Cologne)

The Peer-to-Peer paradigm provides an alternative way of managing resources in various application domains. The primary emphasis of this chapter is placed on presenting an overview of possible approaches for managing the various types of resources, i.e., information, files, bandwidth, storage, and processor cycles, with Peer-to-Peer networks.

4.1 Information

The following sections explain the deployment of Peer-to-Peer networks using examples of the exchange and shared use of presence information, of document management and collaboration.

– *Presence information:* Presence information plays a very important role in respect of Peer-to-Peer applications. It is decisive in the self-organization of Peer-to-Peer networks because it provides information about which peers and which resources are available in the network. It enables peers to establish direct contact to other peers and inquire about resources. A widely distributed example of a Peer-to-Peer application which essentially uses presence information is instant messaging systems. These systems offer peers the opportunity to pass on information via the network, such as whether or not they are available for communication processes. A more detailed description of the underlying architecture of instant messaging systems can be found in [311].
  The use of presence information is interesting for the shared use of processor cycles and in scenarios related to omnipresent computers and information availability (ubiquitous computing). Applications can independently recognize which peers are available to them within a computer grid and determine how intensive computing tasks can be distributed among idle processor cycles of the respective peers. Consequently, in ubiquitous computing environments it is helpful if a mobile device can independently recognize those peers which are available in its environment, for example in order to request Web Services, information, storage or processor cycles. The technological principles of this type of communication are discussed in [627].

R. Steinmetz and K. Wehrle (Eds.): P2P Systems and Applications, LNCS 3485, pp. 25-32, 2005.

- *Document management:* Customarily Document Management Systems (DMS), which are usually centrally organized, permit shared storage, management and use of data. However, it is only possible to access data which has been placed in the central repository of the DMS. As a result, additional effort is required to create a centralized index of relevant documents. Experience shows that a large portion of the documents created in a company are distributed among desktop PCs, without a central repository having any knowledge of their existence. In this case, the use of Peer-to-Peer networks can be of assistance. For example, by using the NextPage-NXT 4 platform, it is possible to set up networks which create a connected repository from the local data on the individual peers [447]. Indexing and categorization of data is accomplished by each peer on the basis of individually selected criteria.
  In addition to linking distributed data sources, Peer-to-Peer applications can offer services for the aggregation of information and the formation of self-organized Peer-to-Peer knowledge networks. Opencola [380] was one of the first Peer-to-Peer applications to offer their users the opportunity to gather distributed information in the network from the areas of knowledge which interest them. For this purpose, users create folders on their desktop which are assigned keywords which correspond to their area of interest. Opencola then searches the knowledge network independently and continuously for available peers which have corresponding or similar areas of knowledge without being dependent on centrally administered information. Documents from relevant peers are analyzed, suggested to the user as appropriate and automatically duplicated in the user's folder. If the user rejects respective suggestions, the search criteria are corrected. The use of Opencola results in a spontaneous networking of users with similar interests without need for a central control.
- *Collaboration:* Peer-to-Peer groupware permits document management at the level of closed working groups. As a result, team members can communicate synchronously, conduct joint online meetings and edit shared documents. In client/server based groupware a corresponding working area for the management of central data has to be set up and administered on the server for each working group. To avoid this additional administrative task, Peer-to-Peer networks can be used for collaborative work. Currently, the best-known application for collaborative work based on the principles of Peer-to-Peer networks is Groove Virtual Office [261]. This system offers functions (instant messaging, file sharing, notification, co-browsing, whiteboards, voice conferences and data bases with real time synchronization) similar to those of the widely used client/server based Lotus products, Notes, Quickplace and Sametime, but does not require central data management. All of the data created is stored on each peer and is synchronized automatically. If peers cannot reach each other directly, there is the option of asynchronous synchronization via a directory and relay server. Groove

Virtual Office offers users the opportunity to set up so-called shared spaces, which provide a shared working environment for virtual teams formed on an ad-hoc basis, as well as to invite other users to work in these teams. Groove Virtual Office can be expanded by system developers. A development environment, the Groove Development Kit, is available for this purpose [187].

4.2 Files

File sharing is probably the most widespread Peer-to-Peer application. It is estimated that as much as 70% of network traffic in the Internet can be attributed to the exchange of files, in particular music files [579]. (More than one billion downloads of music files can be listed each week [457].) Characteristic of file sharing is that peers which have downloaded the files in the role of a client subsequently make them available to other peers in the role of a server. A central problem for Peer-to-Peer networks in general, and for file sharing in particular, is locating resources (lookup problem) [52]. In the context of file sharing systems, three different models have developed: the flooded request model, the centralized directory model and the document routing model [416]. These can be illustrated best by using their prominent implementations - Gnutella, Napster and Freenet.

Peer-to-Peer networks which are based on the Gnutella protocol function without a central coordination authority. All peers have equal rights within the network. Search requests are routed through the network according to the flooded request model, which means that a search request is passed on to a predetermined number of peers. If they cannot answer the request, they pass it on to various other nodes until a predetermined search depth (ttl=time-to-live) has been reached or the requested file has been located. Positive search results are sent to the requesting entity which can then download the desired file directly from the entity which is offering it. A detailed description of searches in Gnutella networks, as well as an analysis of the protocol, can be found in [517] and [515]. Because the effort for the search, measured in messages, increases exponentially with the depth of the search, the inefficiency of simple implementations of this search principle is obvious [328]. In addition, there is no guarantee that a resource will actually be located. Operating subject to certain prerequisites (such as non-randomly structured networks), numerous prototypical implementations (e.g. [146, 182, 469, 505, 138, 397, 3, 446, 642]) demonstrate how searches can be effected more 'intelligently' (see, in particular, [181], but also [8] for a brief overview). The FastTrack protocol enjoys widespread use in this respect. It optimizes search requests by means of a combination of central supernodes which form a decentralized network similar to Gnutella.

In respect of its underlying centralized directory model, the early Napster [437] can be viewed as a nearly perfect example of a hybrid Peer-to-Peer system in which a part of the infrastructure functionality, in this case the index service, is provided centrally by a coordinating entity. The moment a peer logs into the Napster network, the files which the peer has available are registered by the Napster-server. When a search request is issued, the Napster-server delivers a list of peers which have the desired files available for download. The user can obtain the respective files directly from the peer offering them.

Searching for and storing files within the Freenet network [123, 122] takes place via the so-called document routing model [416]. A significant difference to the models which have been introduced so far, is that files are not stored on the hard disk of the peers providing them, but are intentionally stored at other locations in the network. The reason behind this is that Freenet was developed with the aim of creating a network in which information can be stored and accessed anonymously. Among other things, this requires that the owner of a network node does not know what documents are stored on his local hard disk. For this reason, files and peers are allocated unique identification numbers. When a file is created, it is transmitted, via neighboring peers, to the peer with the identification number which is numerically closest to the identification number of the file and is stored there. The peers which participate in forwarding the file save the identification number of the file and also note the neighboring peer to which they have transferred it in a routing table to be used for subsequent search requests. The search for files takes place along the lines of the forwarding of search queries on the basis of the information in the routing tables of the individual peers. In contrast to searching networks which operate according to the flooded request model, when a requested file is located, it is transmitted back to the peer requesting it via the same path. In some applications each node on this route stores a replicate of the file to be able to process future search queries more quickly. In this process, the peers only store files up to a maximum capacity. When their storage is exhausted, files are deleted according to the least-recently-used principle. This results in a correspondingly large number of replicates of popular files being created in the network, whereas, over time, files which are requested less often are removed. In various studies [416], the document routing model has been proven suitable for use in large communities. The search process, however, is more complex than, for example, in the flooded request model. In addition, it can result in the formation of islands - i.e., a partitioning of the network in which the individual communities no longer have a connection to the entire network [376, 123].

4.3 Bandwidth

Because the demands on the transmission capacities of networks are continuously rising, in particular on account of the increase in large volume multimedia data, effective use of bandwidth is becoming increasingly important. Currently, centralized approaches, in which files are held on the server of an information provider and transferred from there to the requesting client, are primarily used. In this case, a problem arises when spontaneous increases in demand exert a negative influence on the availability of the files since bottlenecks and queues develop. Without incurring any significant additional administration, Peer-to-Peer-based approaches achieve increased load-balancing by taking advantage of transmission routes which are not being fully exploited. They also facilitate the shared use of the bandwidth provided by the information providers.

- *Increased load-balancing:* In contrast to client/server architectures, hybrid Peer-to-Peer networks can achieve a better load-balancing. Only initial requests for files have to be served by a central server. Further requests can be automatically forwarded to peers within the network, which have already received and replicated these files. This concept is most frequently applied in the areas of streaming (e.g., PeerCast [480], Peer-to-Peer-Radio [466], SCVI.net [554]) and video on demand. The Peer-to-Peer-based Kontiki network [361] is pursuing an additional design which will enable improved load-balancing. Users can subscribe to information channels or software providers from which they wish to obtain information or software updates. When new information is available the respective information providers forward it to the peers which have subscribed. After receiving the information, each peer instantaneously acts as a provider and forwards the information to other peers. Application areas in which such designs can be implemented are the distribution of eLearning courseware in an intranet [151], the distribution of anti-virus and firewall configuration updates (e.g. Rumor [406]), and also updating computer games on peer computers (e.g., Descent [489] and Cybiko [416]).
- *Shared use of bandwidth:*In contrast to client/server approaches, the use of Peer-to-Peer designs can accelerate the downloading and transport of big files which are simultaneously requested by different entities. Generally, these files are split into smaller blocks. Single blocks are then downloaded by the requesting peers. In the first instance, each peer only receives a part of the entire file. Subsequently, the single file parts are exchanged by the peers without a need for further requests to the original source. Eventually the peers reconstruct the single parts to form an exact copy of the original file. An implementation utilizing this principle can be found in BitTorrent [127].

4.4 Storage Space

Nowadays, Direct Attached Storage (DAS), Network Attached Storage (NAS) or Storage Area Networks (SAN) are the main design concepts used to store data in a company. These solutions have disadvantages, such as inefficient use of the available storage, additional load on the company network or the necessity for specially trained personnel and additional backup solutions.

However, increased connectivity and increased availability of bandwidth permit alternative forms of managing storage which resolve these problems and require less administrative effort. With Peer-to-Peer storage networks, it is generally assumed that only a portion of the disk space available on a desktop PC will be used. A Peer-to-Peer storage network is a cluster of computers, formed on the basis of existing networks, which share all storage available in the network. Well-known approaches to this type of system are PAST [528], Pasta [430], OceanStore [368], CFS [147], Farsite [13], and Intermemory [254]. Systems which are particularly suitable for explaining the way in which Peer-to-Peer storage networks operate are PAST, Pasta and OceanStore. They have basic similarities in the way they are constructed and organized. To participate in a Peer-to-Peer storage network, each peer receives a public/private key pair. With the aid of a hash function, the public key is used to create an unambiguous identification number for each peer. To gain access to storage on another computer, the peer must either make available some of its own storage, or pay a fee. Corresponding to its contribution, each peer is assigned a maximum volume of data which it can add to the network. When a file is to be stored in the network, it is assigned an unambiguous identification number, created with a hash function from the name or the content of the respective file, as well as the public key of the owner. Storing the file and searching for it in the network take place in the manner described for the document routing model before. In addition, a freely determined number of file replicates are also stored. Each peer retrieves its own current version of the routing table which is used for storage and searches. They check the availability of their neighbors at set intervals to establish which peers have left the network. In this way, new peers which have joined the network are also included in the table.

To coordinate Peer-to-Peer storage networks, key pairs must be generated and distributed to the respective peers and the use of storage has to be monitored. OceanStore expands the administrative tasks to include version and transaction management. As a rule, these tasks are handled by a certain number of particularly high performance peers which are also distinguished by a high degree of availability in the network. To ensure that a lack of availability on the part of one of these selected peers does not affect the functional efficiency of the entire network, the peers are coordinated via a Byzantine agreement protocol [105]. Requests are handled by all available selected peers. Each sends a result to the party which has issued the request.

This party waits until a certain number of identical results are received from these peers before accepting the result as correct.

By means of file replication and random distribution of identification numbers to peers using a hash function, the Peer-to-Peer storage network automatically ensures that various copies of the same file are stored at different geographical locations. No additional administration or additional backup solution is required to achieve protection against a local incident or loss of data. This procedure also reduces the significance of a problem which is characteristic of Peer-to-Peer networks: in Peer-to-Peer networks there is no guarantee that a particular peer will be available in the network at a particular point in time (availability problem). In the case of Peer-to-Peer storage networks, this could result in settings where no peer is available in the network which stores the file being requested. Increasing the number of replicates stored at various geographical locations can, however, enhance the probability that at least one such peer will be available in the network.

The low administration costs, which result from the self-organized character of Peer-to-Peer storage networks, and the fact that additional backup solutions are seldom required are among the advantages these new systems offer for providing and efficiently managing storage.

4.5 Processor Cycles

Recognition that the available computing power of the networked entities was often unused was an early incentive for using Peer-to-Peer applications to bundle computing power. At the same time, the requirement for high performance computing, i.e., computing operations in the field of bio-informatics, logistics or the financial sector, has been increasing. By using Peer-to-Peer applications to bundle processor cycles, it is possible to achieve computing power which even the most expensive super-computers can scarcely provide. This is effected by forming a cluster of independent, networked computers in which a single computer is transparent and all networked nodes are combined into a single logical computer. The respective approaches to the coordinated release and shared use of distributed computing resources in dynamic, virtual organizations which extend beyond any single institution currently fall under the term 'grid computing' [220, 48, 247, 213, 224]. The term grid computing is an analogy to customary power grids. The greatest possible amount of resources, particularly computing power, should be available to the user, ideally unrestricted and not bound to any location - similar to the way in which power is drawn from an electricity socket. The Proceedings [51] provide an overview of diverse aspects of grid computing.

One of the most widely cited projects in the context of Peer-to-Peer which is, however, only an initial approximation of the goal of grid computing, is SETI@home (Search for Extraterrestrial Intelligence) [28]. SETI@home is a scientific initiative launched by the University of California, Berkeley, with the goal of discovering radio signals from extraterrestrial intelligence. For this purpose, a radio telescope in Puerto Rico records a portion of the electromagnetic spectrum from outer space. This data is sent to the central SETI@home server in California. There, they take advantage of the fact that the greater part of processor cycles on private and business computers remains idle. Rather than analyzing the data in a costly supercomputer, the SETI-Server divides the data into smaller units and sends these units to the several million computers made available by the volunteers who have registered to participate in this project. The SETI-Client carries out the calculations during the idle processor cycles of the participants' computers and then sends the results back. In the related literature, SETI@home is consistently referred to as a perfect example of a Peer-to-Peer application in general, and, more specifically, a perfect example of grid computing [414]. This evaluation, however, is not completely accurate, as the core of SETI@home is a classical client/server application, due to the fact that a central server co-ordinates the tasks of the nodes and sends them task packets. The peers process the tasks they have been assigned and return the results. In this system there is no communication between the individual nodes. SETI@home does, however, have Peer-to-Peer characteristics [416]. The nodes form a virtual community and make resources available in the form of idle processor cycles. The peers are, to a large extent, autonomous, since they determine if and when the SETI@home-Software is allowed to conduct computing tasks [28, 29]. The shared accomplishment of these types of distributed computing tasks, however, is only possible if the analytic steps can be separated and divided into individual data packets.

The vision of grid computing described earlier, however, extends far beyond projects such as SETI@home. At an advanced stage of development, it should not only be possible for each network node to offer its own resources, but it should also be possible for it to take advantage of the resources available in the Peer-to-Peer network. A currently influential initiative, the Globus Project [590], which is working on a standardized middleware for grid application, has been greeted with wide acceptance throughout the grid community. The project is being supported by important market players, such as IBM, Microsoft, Sun, HP and NEC.