

An Introduction to Peer-to-Peer Computing

David Barkai
Peer-to-Peer Architecture Group
Microcomputer Research Lab
Intel Corporation

Table of Contents

(Click on page number to jump to sections)

AN INTRODUCTION TO PEER-TO-PEER COMPUTING.....	3
OVERVIEW	3
MORE THAN CLIENT/SERVER.....	3
WHAT IS P2P?.....	4
WHY P2P?.....	4
WHY NOW?	5
WHAT IT'S GOOD FOR	5
WHAT'S NEEDED	6
LOOKING FORWARD.....	6
SUMMARY/SURVEY.....	7
MORE INFO.....	7
AUTHOR BIO	7

DISCLAIMER: THE MATERIALS ARE PROVIDED "AS IS" WITHOUT ANY EXPRESS OR IMPLIED WARRANTY OF ANY KIND INCLUDING WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT OF INTELLECTUAL PROPERTY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT SHALL INTEL OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION, LOSS OF INFORMATION) ARISING OUT OF THE USE OF OR INABILITY TO USE THE MATERIALS, EVEN IF INTEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME JURISDICTIONS PROHIBIT THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATION MAY NOT APPLY TO YOU. INTEL FURTHER DOES NOT WARRANT THE ACCURACY OR COMPLETENESS OF THE INFORMATION, TEXT, GRAPHICS, LINKS OR OTHER ITEMS CONTAINED WITHIN THESE MATERIALS. INTEL MAY MAKE CHANGES TO THESE MATERIALS, OR TO THE PRODUCTS DESCRIBED THEREIN, AT ANY TIME WITHOUT NOTICE. INTEL MAKES NO COMMITMENT TO UPDATE THE MATERIALS.

Note: Intel does not control the content on other company's Web sites or endorse other companies supplying products or services. Any links that take you off of Intel's Web site are provided for your convenience.

An Introduction to Peer-to-Peer Computing

David Barkai
Peer-to-Peer Architecture Group
Microcomputer Research Lab
Intel Corporation

Overview

There is a revolution underway that represents a new computing model for the Internet. Its impact will be comparable to the introduction of the graphical-interface browser.

This revolution is being sparked by a phenomenon known as peer-to-peer (P2P) computing, a usage model that has captured the popular imagination with the highly publicized success of Napster*, Scour*, the Freenet Project, Gnutella*, and SETI@home*.

P2P computing can be simply defined as the sharing of computer resources and services by direct exchange. It's an idea that has achieved considerable traction with mainstream computer users and members of the PC industry. Here are three examples:

- The Napster MP3 music file sharing application went live in September 1999, and attracted more than 20 million users by mid-2000.
- By early 2001, the SETI@home program, which uses distributed processing to analyze radio telescope data, had attracted more than 2.6 million users who had donated over 500,000 years of CPU time to the hunt for extraterrestrial intelligence.
- In October 2000, more than 350 individuals representing about 250 companies and organizations gathered for the first meeting of the Peer-to-Peer Working Group.

P2P computing is enabling a new wave of applications with the potential to revolutionize the way computers are used. Developers should keep this new computing model in mind when creating new applications. Some ways to keep up with the latest technical developments include participating in the Peer-to-Peer Working Group, and attending the Peer-to-Peer panels and tracks at the Intel Developer Forum Conference (IDF) Spring 2001, as well as other conferences dedicated to P2P.

More than Client/Server

P2P computing provides an alternative to the traditional client/server architecture. While employing the existing network, servers, and clients infrastructure, P2P offers a computing model that is orthogonal to the client/server model. The two models coexist, intersect, and complement each other.

In a client/server model, the client makes requests of the server with which it is networked. The server, typically an unattended system, responds to the requests and acts on them.

With P2P computing, each participating computer, referred to as peer, functions as a client with a layer of server functionality. This allows the peer to act both as a client and as a server within the context of a given application. P2P applications build on such functions as storage, computations, messaging, security, and file distribution, when handled through direct exchanges between peers.

A peer can initiate requests, and it can respond to requests from other peers in the network. The ability to make direct exchanges with other users liberates P2P users from the traditional dependence on central servers. Users have a higher degree of autonomy and control over the services they utilize.

One of the greatest benefits of P2P computing is community. P2P makes it possible for users to organize themselves into ad hoc groups that can efficiently and securely fulfill requests, share resources, collaborate, and communicate. As P2P computing evolves, we can anticipate the emergence of a wide variety of these online communities.

Figure 1 illustrates the difference between the client/server and the P2P models. It also illustrates how the two models can be superimposed and coexist with each other. In the client/server model every exchange and communication goes through, and is managed by, a central server. In the P2P computing model, the peer systems communicate and exchange directly. Some P2P applications may, at times, also use servers. The overall effect of P2P computing is to take computing to the edges of the net.

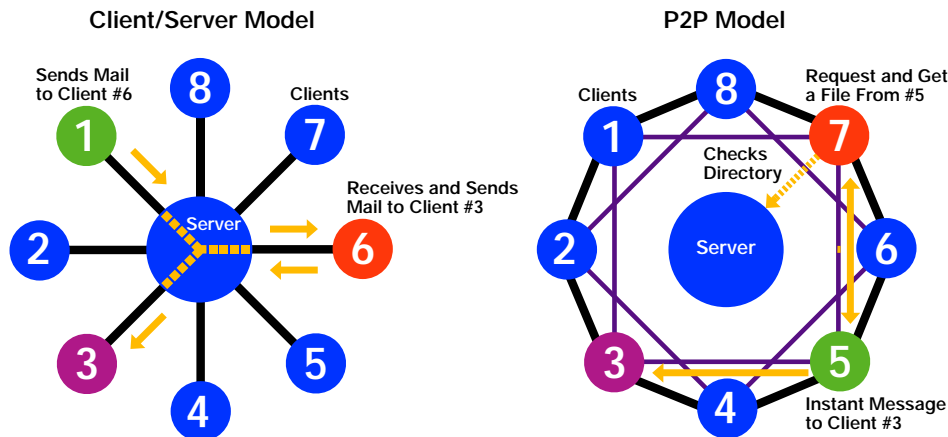


Figure 1. Client Server and P2P Models

What is P2P?

While a peer is a computer that behaves as a client in the client/server model, it also contains an additional layer of software that allows it to perform server functions. The peer computer can respond to requests from other peers. The scope of the requests and responses, and how they are executed, are application-specific.

Typically, there will be a request for access to resources that belong to the other peer. The request may be for information on content and files, or for a file to be read or copied, computations to be performed, or a message file to be passed on to others.

When "computing" in P2P computing is used as a noun, it refers to a framework or computing model. This framework provides the capabilities that allow peers to directly interact. An important characteristic of the direct-interaction capability is that the computing environment becomes decentralized.

When the "computing" in P2P computing is used as a verb, it refers to what we do with P2P framework. Many end-user applications become possible through the P2P services, including storage, computations, messaging, security, distribution, and more. What unifies these application types is sharing of resources with some form of collaboration.

Some P2P evangelists draw a distinction between so-called "pure P2P computing" and "hybrid P2P." The term "pure P2P computing" refers to a model, such as Freenet, where all participating computers are peers. No central server is used to control, coordinate, or manage the exchanges among the peers.

In the "hybrid P2P" computing model, the application relies on a central server to perform some of the required functions. The degree of involvement varies with the application. For example, Napster requires the user to first connect to a control server, where the directory of all available files is stored.

Why P2P?

The P2P computing model offers a number of compelling advantages to individual users and large organizations. These advantages can be divided into technical benefits and social appeal.

Technically, P2P provides the opportunity to make use of vast untapped resources that go unused without it. These resources include processing power for large-scale computations and enormous storage potential.

P2P allows the elimination of the single-source bottleneck. P2P can be used to distribute data and control and load-balance requests across the Net. In addition to helping optimize performance, the P2P mechanism also may be used to eliminate the risk of a single point of failure. When P2P is used within the enterprise, it may be able to replace some costly data center functions with distributed services between clients. Storage, for data retrieval and backup, can be placed on clients. In addition, the P2P infrastructure allows direct access and shared space, and this can enable remote maintenance capability.

Much of the wide appeal of P2P is due to social and psychological factors. For example, users can easily form their own autonomous online communities at the edge of the Net, and run them as they collectively choose. Many of these P2P communities will be ever changing and dynamic in that users can come and go, or be active or not. Other users will enjoy the ability to bypass centralized control. P2P makes users autonomous.

Why Now?

The concept that underlies the P2P computing model, that is, direct exchange and sharing of files and cycles between nodes, is not a new computing concept. So it is reasonable to ask why the P2P model has burst on the scene at this time.

Several essential developments have made P2P for the masses possible. The ubiquity of connected computers has come close to enabling anywhere, anytime access to the Net and its resources. In addition, there is a critical mass of computer users. Improvements in communications bandwidth, still on a fast-track growth curve, make it possible to move large amounts of data and rich media content from one location to another. And today's PCs are sufficiently robust, in terms of processing power and storage capacity, to handle the extra services required in a P2P environment.

Another factor that has accelerated adoption is the emergence of complementary technologies, including recent advances in wireless and software agents that provide more avenues for interesting P2P applications.

While all of these are necessary conditions for P2P computing, something more is required. History shows that a trigger is needed for a new technology to really take off. In the computer industry this is often referred to as a "killer app." The electronic spreadsheet triggered the proliferation of the PC, and the Mosaic* browser triggered the transformation of the Internet into the World Wide Web.

The P2P computing model found its trigger with Napster, followed by Gnutella. Their huge popularity got everyone talking about P2P and helped further stimulate other P2P applications such as Freenet and SETI@home. By the fourth quarter of 2000, over 100 companies and numerous research projects were engaged in P2P computing.

What It's Good For

What are some of the areas where P2P application can thrive? Here are a few examples:

- *Community Web network.* Any group with specific common interests, including a family or hobbyists, can use lists and a Web site to create their own intranet.
- *e-Business.* P2P can add new capabilities, including connecting and enabling the links of a supply chain, distributing information, content, or software more effectively, and keeping information items on their original node with a central directory or a search capability.
- *Gaming.* A P2P infrastructure provides a natural foundation for the development of online community games that are not centrally controlled. Developers can focus on game features instead of the interface to the communications protocol.
- *Search engines.* Fresh, up-to-date information can be found by searching directly across the space where the desired item is likely to reside.
- *Virus protection.* Relationships among the nodes on the P2P community allow for collaboration in virus detection and warning, as well as automatic quarantining of the community against further attacks.
- *Edge services.* There are instances when it is desirable to place the data, prior to its use, closer to the client requesting it. Online training modules that contain video segments, for example, provide the desired effect when the large data files are located close to the online trainee. Multiple clients offering storage space can provide more flexible and reliable service compared to a server.
- *Collaborative development.* The scope can range from developing software products to composing a document to applications like rendering graphics.

What's Needed

While the potential of P2P computing has been demonstrated, and the interest level of the industry is extremely high, several issues remain to be resolved. To date, P2P applications cannot easily communicate with other P2P applications, and each has its own set of basic services and plug-ins. Applications that originate from different sources are usually not integrated and cannot communicate with one another. Even more significant is the fact that developers of new applications cannot readily benefit from the experience of other P2P developers.

As a result, application developers spend much of their time and energy developing services that have already been successfully resolved by others. This is time and energy that could be better spent on adding features and functionality to new applications.

This issue will be resolved when the industry collaborates on solutions that enable interoperability between P2P applications. This interoperability can be achieved through a common set of services that provide the functionality needed for P2P computing above the functionality provided by the peer's operating system. These common services can be thought of as a middleware layer. One of the principal advantages of a common middleware is that application developers will no longer be required to keep creating the same basic services over and over again. Interoperability means that P2P applications can communicate across different software environments. Peers running Windows* or Linux* or any other operating system can share P2P applications. Applications using different programming languages can communicate and be integrated via a common middleware. And common middleware provides a mechanism and an infrastructure for incorporating devices other than PCs and servers. Such devices include wireless and handheld products and various network appliances.

Figure 2 shows a P2P middleware layer that fits between a peer's local operating system and application interfaces that the P2P application can directly call. The creation of standards and a common infrastructure has benefits for users and developers alike.

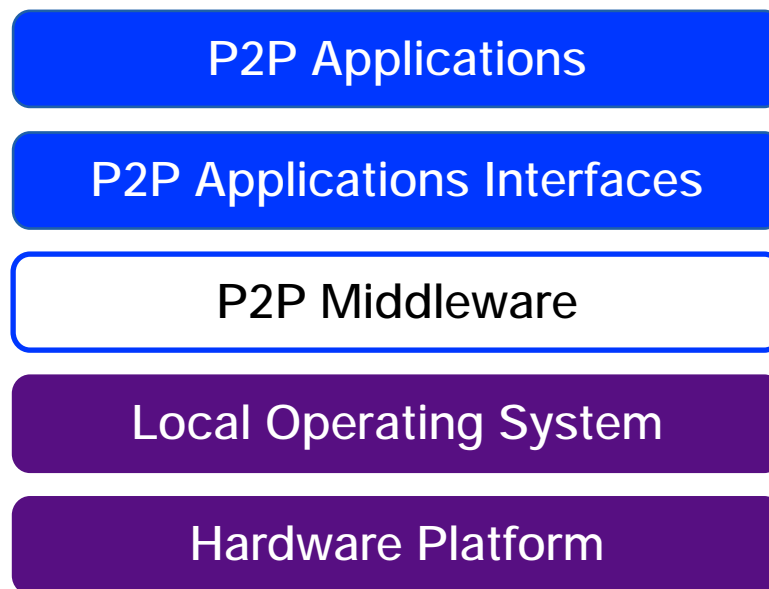


Figure 2. The P2P Stack

Looking Forward

The P2P computing revolution is happening, and it's big. The success of pioneering P2P applications proves that decentralized computing has enormous appeal for millions of users and obvious benefits for application developers. In addition to "thinking peer-to-peer" during the development process, it is important for developers to understand the power of P2P technology. One of the best ways to do this is to get involved with the Peer-to-Peer Working Group.

The Peer-to-Peer Working Group is an industry consortium dedicated to the advancement of infrastructure standards for P2P computing. The goal of the group is to develop infrastructure standards for interoperability, security, performance, management, and privacy that will enable P2P computing to become ubiquitous. The working group will determine areas for standardization, work to develop specifications, and promote adoption of these specifications as standards throughout the computer industry. Participation is open to anyone interested in peer-to-peer computing.

Developers should also stay tuned for the latest information from Intel, made available through the Intel Developer Forum (IDF) Conference and in future issues of *Intel Developer Update*.

Summary/Survey

For 20 years people in academia and research have experimented with peer-to-peer computing technologies, mostly in relatively small and controlled environments. Recently we have seen P2P applications that have captured the imagination of millions of users.

The idea of sharing and collaborating via the Internet extends to dividing the load of performing large computations, collaborating in creating media or software, conversing online and directly, and organizing into online communities. There are many more intriguing possibilities, limited only by the imagination of application developers and users. P2P computing is evolving into a phenomenon whose impact will be no less than that which we experienced when the graphical-interface browser and when the client/server computing model were introduced.

The physical infrastructure of computers and Internet communications is in place, and the concept of P2P in the mainstream of computing has been demonstrated. The time has come to move from the mode of early experimentation to that of a solid computing model.

More Info

Membership in the Peer-to-Peer Working Group is strongly suggested for any developer interested in creating applications for this revolutionary Internet computing model. See the [Peer-to-Peer Working Group](#) home page for membership information and links to information and industry events.

By attending [the Intel Developer Forum Conference Spring 2001](#), you can hear industry leaders share their vision of P2P standards-setting and technical challenges. IDF technical tracks will provide a detailed look at a variety of important issues including security, usage and application examples, and technical challenges.

Future *Intel Developer Update* articles will explore the architecture and infrastructure of P2P computing, and examine some of the technologies and services needed by P2P developers, including security, privacy, resource management, resource sharing, and the advancement of interoperability.

Author Bio

David Barkai joined Intel in 1996. Prior responsibilities at Intel include development of IDF Conference content and work in the Microcomputer Software Lab, where he focused on applications for Intel® Architecture workstations. In addition, David has worked in the area of scientific and engineering supercomputing applications and conducted pioneering work on vector processors. He holds a Ph.D. in theoretical physics and has published more than 20 papers in the areas of physics, numerical methods, and computer applications and architectures.

—End of Intel Developer Update Magazine Article—