mCommunicator.com

An Overview of Peer−to−Peer Technology and Services

For more information , please contact::

mCommunicator.com (India) Private Limited
117 Mittal Chambers, Nariman Point,
Mumbai 400 021, India
Phone +91 (22) 202 3015, 202 5837 Fax +91 (22) 284 3963
http://www.mCommunicator.com, info@mCommunicator.com

Contents

Peer−to−peer: An Overview

The Peer−to−Peer phenomenon has once again caught the fancy of the world, thanks to implementations like Napster that managed to mobilize the ‘power of the masses’ to shake the foundations of a well−established recording industry. Napster, Gnutella, etc. work as file trading systems, these are systems where users can search for and retrieve files from each other’s computers. As such they provide a massive search engine for the ‘global hard disk,’ unifying the data that lies distributed on each one of our computers.

However they fulfill only a part of the peer−to−peer promise.

Peer−to−peer technology is about using resources available with each user and generating a greater good. The resources that envisaged peer−to−peer models intend to harness are:

· Distributed Computing Power

· Distributed Information

· Distributed People

Distributed Computing

People have been using arrays of large machines for computational tasks for sometime. Examples can be found in ‘network rendering’ features that many graphics applications provide, and in clusters of servers. However the traditional distributed computing looked at using tens or even hundreds of closely connected machines, the new peer−to−peer models of distributed computing are being deployed at ‘Internet Scale’ with thousands and millions of computers from all over the internet participating. SETI@Home, the most popular of these applications, currently manages to yoke together some impressive computing power, larger than many supercomputers. Such is the power of using the Internet for Distributed Computing.

Distributed Data

Napster and Gnutella have done a great job of unifying the world’s hard disks. However searching for information on these systems is as yet not perfect. There is only a little information you can query for; also there is also misrepresentation (and irregular categorization) that impedes the search. Napster is slightly better than Gnutella because the MP3 files that Napster trades already have a cataloging system built into them, and using this little bit of ‘application logic’ one can search for music by artist name, title or even genre, which is definitely of more help than the single text box of Gnutella.

There is also another issue: the information that one can trade with a globally dispersed and divergent counter parties is limited to some pirated music tracks and copied digital images. However, the file−sharing model is the best way for a closed group of users to share information with one another. Consider a group working a project together; they can have an information repository where they can store all the common files, and everybody on the project can now access these, as if they were on their local computers. Think of a geographically dispersed family they will derive significant value from share their pictures with each other, among other things.

Internet based file−sharing has a lot of added potential, especially for businesses and impromptu groups,

especially if the applications improve by adding more sophisticated search and find heuristics.

Distributed People

Often information is available with people rather than with computers, file−sharing systems are good for exchanging information that is easily codified. However for unstructured information that people carry around with them, a person−to−person communications method is required. Email used to fill that void in the ‘client−server days’ and it is instant messaging with its promise of immediate delivery that supersedes email in the peer−to−peer world.

mCommunicator: Peer−to−peer Technology Offering

mCommunicator offers a suite of products and services that deliver on different parts of the P2P technology promise. The aim is to build custom solutions and products that use P2P technology concepts to attack real−world issues.

mCommunicator has built a peer−to−peer instant message exchange system that is based on custom protocols, client−side components and central servers that provide directory and fallback routing services, this package is collectively referred to as the messaging engine, which is responsible for exchanging information between different clients. This engine core will be leveraged for building various applications and solutions that leverage peer−to−peer technology.

Anatomy of a Peer−to−peer application

Traditional Client Side Applications

A traditional client side application consists of:

· A Presentation Layer that takes care of the User Interface and input and output from the user.

· The Business Logic (or Application Logic) Layer that contains the special logic that is required for the particular application.

· The Database Layer that would save and retrieve the information generated in the application.

Sometimes the database system was local, sometimes server based, however the general structure remained the same. The primary purpose of the application was to retrieve data from the database, transform it using the application logic, and present it to the user for viewing or modification. The user may elect to create some new information and that was stored way in the same database.

Peer−to−Peer Applications

A P2P application is essentially a traditional client side application, with just one more component added to the set: the Information Exchange (or Message Exchange) Layer.

To deduct how a P2P application works, take the case of the famous SETI@Home.

The application starts and waits to receive some ‘processing task’ from the message exchange layer. On receiving the request to process a certain chunk of radio reception data, the application then stores the incoming data in its database. Whenever the processor load on the host machine drops below a predetermined level, the application fetches some of the data from its local database system and begins applying transformations to the data as defined in the application logic. Whenever the host computer goes into screen saver mode, the data that has been processed so far is displayed as a cute screensaver display. Whenever the processing task is completed, the application logic forms a special message and sends it using the Message Exchange Layer to the primary server, which then applies its own application logic to it.

The Messaging Layer is the Difference

As explained above, a peer−to−peer application is really a traditional application integrated with a message exchange layer. There is no competitive advantage in “traditional applications” however, we at mCommunicator have the message exchange layer in place and can use it to build peer−to−peer solutions for most requirements.

Sample Application Areas for mCommunicator Peer−to−Peer Technology

The primary benefits of peer to peer computing accrue in environments where data available on multiple machines can be collated and accessed through a single interface. Some envisioned application areas are:

Internal Enterprise Applications (Including EIP's and Intranets)

An area that can benefit greatly from P2P technology is that of communications and knowledge sharing infrastructure within organizations.

· **Project Workgroups:** The potential for value add from a file sharing application for users involved in a project is tremendous. Typically most of the information will be stored in files on the users' local hard disks, a file sharing application will allow them to share this date between their peers. Thus allowing the users to keep all their information local on their machines and edit and update information as required, thus keeping all the shared data current all the time. Also they would benefit immensely from other P2P applications like instant messaging to enable all of them to collaborate on their projects.

· **Distributed Offices:** File sharing applications are also a great value−add for companies with multiple offices. These companies can now link up these offices and enable the users of different offices to share current documents with each other through the Internet, as easily as they would do if they were in the same office. This is much cheaper than the other alternative of having a private WAN between these offices.

· **Distribution Chains and Supply Chains:** It is nearly impossible to cover an entire distribution chain with a private network, as the costs of doing so would be prohibitive however it is possible to enable all the users in a distribution chain to connect to the internet and using P2P applications it is possible to use the internet to enable immediate transfer of current data between an entire distribution chain.

· **Other applications:** Inside corporate environments, information delivered to direct to the users has value over information that stays stored away in a central location and hence may be forgotten.

The primary benefits that an organization gathers by using peer−to−peer for internal knowledge pooling and collaboration are:

· **Newer, More Updated Information:** Because the information is stored with the creaor of the information or is at the point where it is constantly getting changed or added, the data available throughout a P2P system is always more current than thru other systems.

· **Easier Knowledge Creation:** It is clearly understood that the challenge for knowledge management is to have users submit their knowledge into the system. Peer−to−peer models make knowledge creation easier because they store the data with the originator who may use his favorite tools to create the knowledge as opposed to taking on the cost of learning a set of new tools that will be available on the web site.

· **Economic Efficiency:** People use their own idle computing resources for hosting and managing peer to peer networks thus results in better utilization of available resources for the company.

Non−traditional devices (mobile phones, PDA's) and traveling users that are away from their desktop computers may use the system using a web interfaces to these P2P networks.

External Networks for Companies (example: Specialized B2B Exchanges)

Benefits of P2P technology are not limited only to intra−organizational processes. This example details how a B2B exchange can run in P2P mode. Needless to say the primary benefit is reduction of infrastructure cost and hence increasing efficiency of the exchange.

Assume a very simplified model of a B2B Exchange where the functionality is limited to only the following:

· Searching for new Business Partners

· Exchanging Transaction Information (Quotations, Negotiations, etc…)

This kind of an exchange would be a natural for peer−to−peer style networking. Implementation of such a system would not be unlike the current file−trading systems.

There will be a central search facility where users can search published data about other users, this will be information about the companies and their offerings. This information will be stored in XML files at the users machine and will be queried like Napster queries for mp3 files. The application logic would take care of presenting the information in an easily understood manner for the end user.

The benefits for the end user over current 'web−based' B2B exchanges are:

· **Instant Information:** Current web−based B2B exchanges are based on a store and forward model; the users create new information and store it on the web site, when the counter party to the transaction comes to the website to receive the information. Business negotiations are typically of the nature of repeated quick back and forth communications between people, and hence would be aided by an Instant Information Exchange model provided by P2P solutions.

· **Control over data:** In a peer−to−peer model the users would retain control over their data as it resides on their own machines. People currently have their own data analysis and reporting tools that they use with existing business data; the same tools may be leveraged to analyze the exchange data that they have stored on their desktop. Also it is easier to change pricelists, for example, on ones own computer rather than at a website hence more control also ensures more current information for the exchange in general.

· **Reduced Cost:** A web based B2B exchange as currently implemented, increases cost for the market maker significantly as opposed to a P2P model. Cost points that the market maker could avoid in a peer−to−peer environment:

o The central repository where all the data is stored for a web−based exchange can be done away with

o The computation power expended on applying application logic based transformations to this data and also on couching it in a presentable format now need not be expended by the market maker's central computers but this cost gets spread across the multitude of users of the site.

o The market maker need not provide tools additional tools for analysis as the users can now use their favorite tools for the same.

- He also need not provide the computation power for these tools.

The above discussion focuses on users who use conventional PC's for accessing the exchanges. However the peer−to−peer model can also be extended to include non−traditional devices primarily mobile users. This will be done by having a special set of servers called 'public peers' which will run a version of the peer−to−peer software which outputs its presentation layer in format like HTML or WML which is more amenable for display on the mobile device.

Extendible to other business needs: This is just an elementary discussion of the B2B exchange but it serves to show the way for implementing complex peer−to−peer models for business data exchange including captive exchanges like **large company extranets** and **supply chain linkages.**

Appendix: Functioning of the Message Exchange Layer

mCommunicator's Messaging is based on unique user identifiers. A Unique user identifier of the form user@host provides a deterministic pointer to a user. These User Identifiers are globally unique. Other users communicate with these UID's and send or receive messages from them.

If a message is intended for a 'buddy' a direct peer−to−peer connection is setup and they exchange messages, else if the message is destined for another user on the local server then the message is delivered directly to the user. However if the message is sent to a remote user (user@other−host) the message is transmitted directly to the other user, or over a direct link between the servers or through a gateway. So the message gets delivered instantly from user to user, however it may go through one or many hops.

Hence, mCommunicator can support many different configurations such as:

· Peer−to−Peer with the server providing only passive directory services

· Clients−to−Client with the single server providing directory services as well as message queuing and delivery.

· Multiple Client spread across multiple servers, with the server providing directory services as well as message routing and delivery.

Peer−to−peer message delivery is the primary mode of message exchange, however an alternative route is through the server, which is used in case where the peer−to−peer connectivity information is not present at one end, or also in case of corporate firewall scenario's, which typically prevent incoming connections. The third−case of the message traversing multiple servers occurs when the users are from different domains, and are supported by distinct server hardware, and they are forced not to use Peer−to−Peer connections. In such a scenario, the message would be routed across servers and then delivered to the end user.

The client software could be running on a PC for desktop use or even on a web server where a specialized client software will written to use an HTML / WML presentation layer for use from remote web browsers and from WAP enabled mobile devices.

Presence

"Presence" characterizes the existence of a software or hardware device through which information can be communicated. Presence indicates information is available about the various communications devices that a person may use and their status, such as whether a cellular phone is switched on or if a user is logged on to a PC. Messaging works only in conjunction with the notion of the user's presence because the routing service uses this information for delivery of messages.

Presence information is published by the client's application and is maintained at the server for the length of the session. This presence information allows other clients as well as server side routing to make decisions about the method of delivering information to the client.

Presence is implemented in a Presence publisher (Presentity) and presence subscriber (Watcher) model, in real world terms, "I am a subscriber to the presence information of all my buddies."

Technical overview

The most important elements of the mCommunicator messaging model are *message*s, *clients* and *servers*. A typical mCommunicator message is a fragment of application−defined (or user−defined) content. It is typically delivered right to the recipient client or to the sending clients' server, which then sends it across the mCommunicator network. Messages are addressed to a target by specifying the target's mCommunicator Unique Identifier, which is of the form username@hostname.domain.

Client

The Client Software is broken up into a network layer that handles all the messaging components and the rest of the application that provides custom functions based upon various requirements.

The Application layer as defined above is essentially a traditional client application that has custom business logic, and presentation layers to process the information received form the network and also typically a database system for storing the information created in the system.

For example in case of the mCommunicator instant messenger, this layer shows the entire interface, the messenger and the message boxes, it also supports application specific functions like message histories, previous messages, advertisement display, etc…

The network layer can support

- Outgoing connections over TCP
- Incoming Connections over TCP
- Other protocols (currently HTTP) through both client side and server side adapters

Messaging Servers

Authorized subscribers to users presence information can also request a contact tuple, which provides them with a server−independent way of contacting and messaging each other. However, when either such a direct connection is not possible, or when the contact tuple is hidden due to network topology or security issues, the message routing is server based.

Server lookup

As mentioned earlier, the user identifiers are of the form user@host.domain the system keeps an internal routing table for each host / domain combination. This essentially is modeled on the way the DNS system supports MX records for mail exchange. This system is not integrated with the DNS service but runs as a subsystem of the mCommunicator Software.

Message Delivery

The messages are delivered from server to server by a direct TCP connection and from there they are passed to the client by an inbound message stream. In cases where an inbound stream is not possible, other adapter like the HTTP adapter are used.

Message delivery is a function of presence:

· **Synchronous message delivery** (**online messages**): In a direct peer−to−peer scenario or when both clients are online in a client−server scenario, the networking component contacts the target client and delivers the message. Contact with the remote client may involve routing through intermediate server and/or gateways as explained above. The delivery happens through one or more *protocol adapters*, in order to support various communications protocol, such as HTTP or native TCP/IP.

· **Asynchronous message delivery (offline messages)**: When the recipient is not connected to the Internet, the application passes the message to the server for delivery. The Server delivers the message to the recipient's server (which may be a separate or the same server), which holds the message in queue, until it can be delivered.

References

The sites and services mentioned in this note can be found at:

1. SETI@Home: http://setiathome.ssl.berkeley.edu/
2. Napster: http://www.napster.com/
3. Gnutella: http://gnutella.wego.com/

Other articles and sites that refer to Peer−to−Peer technology and its uses:

1. Peer−to−peer working group: http://www.peer−to−peerwg.org/
2. The O'Reilly Peer−to−Peer conference: http://conferences.oreilly.com/P2P/
3. Red Herring's Trend #1 for 2001: Distributed Computing, http://www.redherring.com/mag/issue86/mag−computing−86.html
4. The Napsterization of B2B, Andrew McAfee, Harvard Business Review, 1st Nov 2000 http://www.hbsp.harvard.edu/hbsp/prod_detail.asp?F00601.html

Some Companies in the Peer−to−peer Space

1. Groove Networks: http://www.groove.net/ and http://www.groovenetworks.com/
2. Centrata: http://www.centrata.com/
3. United Devices: http://www.uniteddevices.com/
4. Proivo: http://www.porivo.com/