Distributed systems – Technologies
2. DS case study: P2P Architecture

A wave of digital revolution

- Appears around 2000 as
  - a new paradigm in network computing,
  - a new technology for connecting people, and
  - effectively utilizing untapped resources anywhere on the Internet and the Web
- a network of equals to replace? traditional client/server partitioning of functionality
- new paradigm of servents - a term formed from server + clients, partners in computing opportunity
- rise once again to the "power to the people" mantra of the 60s.
  - rather than protesting for personal individuality, the protest was for machine and silicon equality
  - a new generation of protesters took to the digital highways and soon caught the attention of the world.

Key factors of P2P appearance

- increased availability of inexpensive computing power, bandwidth, and storage.
- explosion of content and subscribers on the Internet
- widespread adoption of Internet-based protocols

-> where the content resided was no longer an issue.

It resided everywhere, within servers, within clients, within the fabric of the network itself.

Liberal dissemination of data coupled with unlimited total access.

Drawbacks

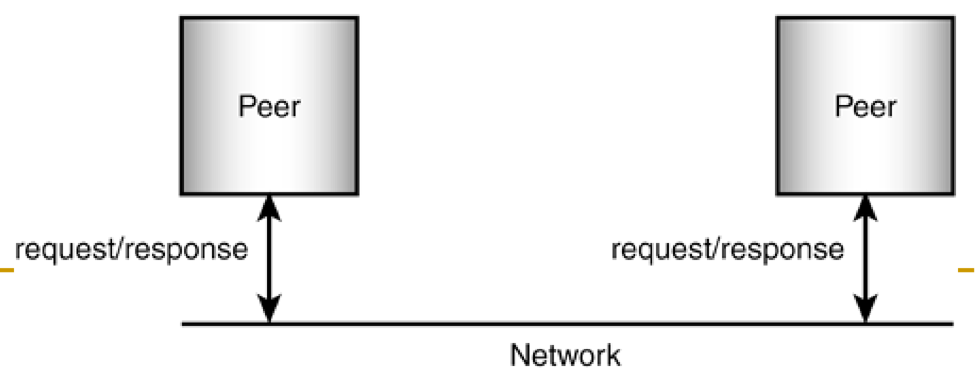
- digital asset ownership infringement
- copyright abuse

E.g. Napster - popular MP3 file sharing program.

Technology disrupted the status quo and brought into question the very laws that digital world.

Definition

- Peer: "a person who has equal standing with another or others, as in rank, class, or age."
- relationship between two computing entities as something-2-something
- fundamental to P2P is node equality,
  - a node is defined as any processing entity that exists as a particular and discrete unit.
- Peers are assumed to possess an equal capability in functions or services that they provide
  - contrary to traditional client/server models, in which the server possesses greater functionality and control than the client.



Napster example

- Not pure P2P:
  - MP3 files were indexed on a central server; only the files remained on and were accessed from peer components.
- large networks of cooperating nodes could be assembled,
- nodes were common PCs that dynamically assembled to form a distributed file system
- network was constantly changing.
- nodes of the network were transient by nature:
  - they would enter and leave the network at will

Types of P2P applications (1/2)

1. Managing and sharing information
   - Files, documents, photos, music, videos, and movies all want to be shared
   - More advanced sharing enables one machine to act as a general task manager by collecting and aggregating results
   - for example,
     - Google.com is an example of distributed task-managing system.
     - Gnutella is an example of a personal P2P file-sharing system.
2. Collaboration
   - Individual users find that address book, scheduler, chat and email software improves their productivity.
   - For example, Java developers use OpenProjects.net to collaborate.
   - On a broader scale, hundreds of thousands use instant messaging, which may be the most popular P2P application to date.

Types of P2P applications (2/2)

3. Enterprise resource management
   - Coordinating workflow processes within an organization leverages the existing infrastructure of networked desktop computer systems.
   - For example, Groove enables an aerospace manufacturer to post job order requests to partner companies and route the completed requests from one department to the next.
4. Distributed computation
   - A natural extension of the Internet's philosophy of robustness through decentralization is to design P2P systems that send computing tasks to millions of servers, each one possibly also being a desktop computer.

Motivation to Adopt P2P

- Decentralization—
  - Businesses realize greater efficiency and profits by attaining a flexible state.
  - From mainframes, to a client/server model, to Internet computing, and now to P2P.
- Cost and efficiency—
  - Hardware and software will continue to be inexpensive and powerful.
  - New systems that increase the efficiency or utilization of hardware or software present a compelling case for making the investment.
  - P2P additionally has the capability to exploit resources that in the past went unrecognized.
- *Pervasive computing*—
  - Imagine information systems everywhere: computer chips in clothing, appliances, automobiles, devices, anything you can think of.
  - Not only are they everywhere, but they are connected.
  - The market for network-connected devices continues to grow, and P2P systems are being designed to support the device market.

P2P Architectural Details

- easiness with which large networks of cooperating nodes can be assembled, and that these nodes live on the edge of the network.
- nodes are common PCs that dynamically assemble to form a distributed file system.
- the network is constantly changing
- many nodes are behind firewalls
- the computers may be turned off at night, and they enter and leave the network at will.
- antithetical to the network organization and typical filesystems found on business networks.

Forming Dynamic Networks

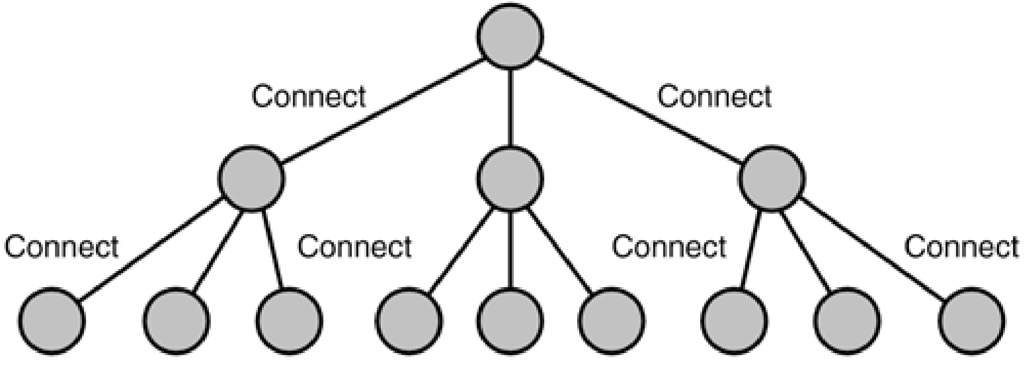
- Internet is a dynamic network with a number of static properties: each machine that connects to the Internet is assigned a unique IP address (IPv6, DNS, NAT)
- Use virtual namespaces that provides a method for persistent identification (similar to an e-mail address that identify the person, despite the access point)
- the nodes within the network "find" or "discover" each other using IP and DNS as a navigational aid to build a dynamic or virtual network.

Discovery on two levels

- discovery process is associated with finding a peer.
  - a peer refers to a computing entity that is capable of understanding the protocol of the messages being exchanged.
  - It is an entity that "speaks" the same language—it understands the semantics of the dialogue.
  - Peer discovery is required to find a service or help divide and conquer many problems associated with information processing.
- finding resources of interest.
  - The early P2P applications dealt with file sharing and searching.
  - In contrast to popular search engines, P2P applications define new techniques to discover files and information on the Internet
    - The massive amount of information available on the Internet and its exponential growth is outpacing traditional information indexing techniques.

Gnutella example – broadcast storm

- file sharing and search program
- uses an unconventional broadcast mechanism to discover peers
  - the broadcast technique grows exponentially—the more users, the more broadcasts
  - when the size of the user base grew too quickly, the system came crashing to a halt, flooding networks with Gnutella requests.



Simple broadcast

- sends a request to every participant within the network segment or radius
- used for discovery, it can reach a large number of potential peers or find a large number of resources
- Drawback:
  - as the user base grows linearly, the number of requests grows exponentially -> huge bandwidth requirements.
  - security and denial-of-service implications
    - A malicious peer can start flooding the network with a number of requests disproportionate to the true size of the user base.

-> Simple broadcast is only viable in small networks.

Selective broadcast

- peers are selected based on heuristics such as quality of service, content availability, or trust relationships
- requires to maintain historical information on peer interactions.
- discovery requests are sent to selected peers,
- the response is evaluated against the criteria defined for peer connections.
- For instance,
  - send discovery requests to peers that support a certain minimum bandwidth requirement.
  - send requests for resources only to peers likely to have that content.
- The more you need to know about the participants, the less dynamic the system can become -> eliminate the benefits of P2P if fixed and static relationships are not mitigated.
- Security is still a concern with selective broadcast

Adaptive broadcast

- minimize network utilization while maximizing connectivity to the network
- Selection criteria can be augmented with knowledge of your computing environment.
- For instance,
  - you can set the amount of memory or bandwidth you will consume during discovery operations.
  - you can limit the growth of discovery and searching by predefining a tolerance level for resource exploitation that if exceeded will begin to curtail the process.
  - excessive resources are not being consumed because of a malfunctioning element, a misguided peer, or a malicious attack.
- requires monitoring resources, such as peer identity, message queue size, port usage, and message size and frequency.
- can reduce the threat of some security breaches, but not all.

Resource Indexing

- Finding resources is closely tied to finding peers
  - the difference is that peers have intelligence: they are processes capable of engaging in digital conversations through a programming interface.
- resource is much more static, and only requires identity
- “presence” = the ability to tell when a peer or resource is online
- discovering resources can be done using centralized and decentralized indexing.

Centralized vs. Decentralized indexes

- good performance
- Drawbacks:
  - bandwidth and hardware requirements of large peer networks can be expensive
  - hit the scalability wall at some point, regardless of the amount of software and hardware provided

- overcome the scalability limitations of centralized systems
- every document or file stored within the system is given a unique ID
- E.g. in FreeNet
- Drawback:
  - searches have to be exact
  - problems with keeping cached information consistent (major detriment to scalability).

Node Autonomy

- Peers are independent and self-governing.
- P2P systems are built under the assumption that services are distributed over a network, and that the network is unreliable
- Failures are not always detected immediately.
- Response time and latency issues introduced as a result of remote communication can be unpredictable
  - The network can have good days and bad days!
- Peer-to-peer interaction can become unstable as error paths and timeouts get triggered excessively.
- Synchronization often strains available bandwidth.

P2P and Web Services

- Common vision:
  - Web services enable developers to build loosely coupled, self-describing, highly scalable systems that provide interoperability between software on different platforms
- P2P technology look very attractive as a tool for designing high performance, scalable, server-based systems
- For example: a stock trading application in which
  - the trades are handled by a centralized server using SOAP, and
  - the stock charting and history functions come from a network of information sources using P2P.

Examples of P2P Applications

- Instant messaging
- Managing and sharing information
- Collaboration
- P2P services
  - Distributed processing (grid computing)
  - Distributed storage
  - Distributed network services
- From former distribute stolen copyrighted music and video files, to currently e-market hubs, corporate infrastructure, and Internet-enabled applications!!!
- Single-function P2P applications are giving way to multifunction service-based architectures.

Instant Messaging (IM)

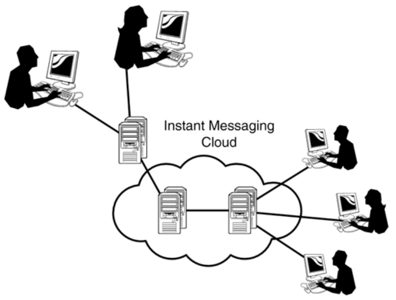
- killer application for P2P
- enables online users to communicate immediately and in real-time, one-to-one or in a group
- The user
  - activate a special piece of client software that communicates with a central server and registers the user as being online.
  - its registration is mapped to an identity, such as a nickname or screen name.
  - is then able to invite others to a conversation, or can be invited.

IM characteristics

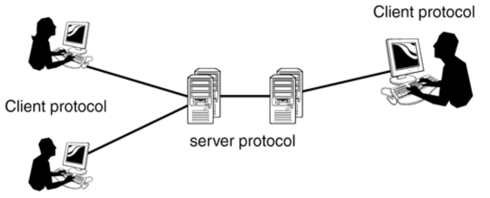
- IM servers communicate using an IM server protocol that enables messages to be relayed across the Internet.
- interconnected IM network forms an IM cloud, or backbone
- Most servers support a proprietary protocol, which has made it difficult for IM users to communicate across multiple IM systems.
- Unlike email, in which a message is stored and delivered once the user has connected to an email server, IM systems provide immediate end user delivery.
- To avoid this uncertainty in delivery, IM systems provide a "buddy list" (online status)
- Applications of presence (online status) and IM currently use independent, nonstandard, and non-interoperable protocols developed by various vendors

IM cloud or backbone

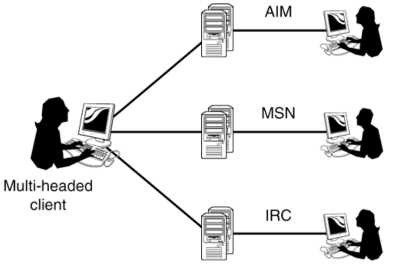
- proprietary IM cloud



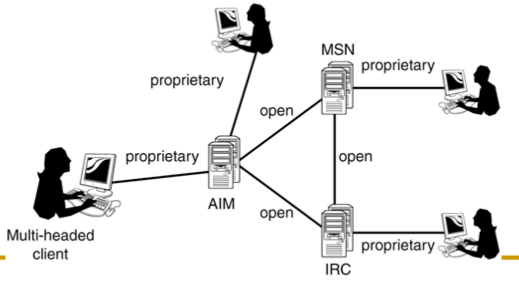
- define a client and a server protocol



- communicate with multiple IM systems



- interoperability at the service level



Examples

- IRC. Internet Relay Chat (www.irc.org),
  - often cited as the original chat medium on the Internet.
  - forefather to many of the IM protocols that have been developed.
  - IRC protocol was designed for use with text-based conferencing.
- Jabber (www.jabber.org)
  - open source IM platform developed by the open source community
  - feature that distinguishes it from existing instant messaging services is its open XML protocol.
  - attempts to build the interoperable protocol that all IM vendors will support
  - you use the Jabber XML protocol (XMPP) from the client to the Jabber server, and the server loads an IM-specific transport module to interoperate with the proprietary IM system.
  - Jabber architecture resembles email.

Managing and Sharing Information

- File sharing
  - e.g. Gnutella, Freenet
  - ad hoc P2P communities
  - share files without requiring centralized coordination or control
- Resource sharing
  - use of the cumulative power of dynamically networked peers to tackle tasks previously possible only on supercomputers
  - E.g. SETI@HOME
- Distributed search engines
  - address the problems inherent in the large size of the information space
  - use a divide-and-conquer strategy to locate information and perform these searches in real time

NextPage

- manage, access, and exchange content across distributed servers on intranets and via the Internet
- platform indexes and connects content across organizational boundaries,
- allowing you to search and locate content from multiple locations without having to know where it physically resides
- extensive array of search functions including keyword, Boolean, phrase, ranked, wildcard, and so on.

Collaboration

- Rendezvous points
  - Shared spaces enable to peers to identify a common network accessible meeting place.
- Identity and presence services
  - Shared spaces become the common point for searching, retrieving, and updating identity and online status.
- Group membership
  - Shared spaces form the basis for defining a group or community of peers connected by a common interest or goal
  - Group membership in a shared space is controlled by the group, rather than by a central administrator.

Groove

- developing and deploying secure enterprise applications
- it offers instant collaboration, shared spaces, Web connectivity, and a host of add-on applications
- developers can integrate Groove into their existing systems.
- access data on traditional corporate networks, but also in nontraditional devices such as PDAs and a range of handheld devices

Distributed Data Services

- move data closer to usage using multiple nodes and sophisticated routing algorithms
- motivation:
  - Edge devices are plentiful and under-used.
  - Users are more mobile, and must access information from multiple locations.
  - Centralization has proven costly and prohibitive beyond a certain level of scalability.
  - Mobile users are demanding faster access to content.
  - Service providers are searching for cheaper solutions.
- require intelligent caching over a widely dispersed cluster of nodes.
- some distributed content networks use predictive seeding to preconfigure the location of data based on usage patterns and known heuristics
- multisourcing permits a content network to map multiple communication paths to a cluster and/or data store (through intelligent routing)
- security (encryption) is required to ensure integrity of data in transit.

Distributed computing (see next week!)

- use the idle processing cycles of the PCs on your network.
- makes commodity devices available to work on computationally intensive problems that would otherwise require a supercomputer or workstation/server cluster to solve.
- three fundamental components to the architecture:
  - The Network Manager manages client resources, and controls which applications are run on the client machines.
  - The Job Manager permits application users to submit work and monitors the progress of this work, and retrieves results.
  - The Client manages the running of applications on a client machine.
- Gain in popularity in genetics and bioinformatics.

Distributed network services

- localize traffic, lowering bandwidth expenditures and improving response times.
- bandwidth management technologies
  - allows the control of the network traffic, ensuring that networks deliver predictable performance for mission-critical applications
  - can prioritize mission-critical traffic, as well as guarantee minimum bandwidth for the most critical, revenue-generating traffic (for example, voice, transaction-based applications).
- Example: webcasting
- scale to use the available bandwidth and computer resources of new participants who request the stream.
- devices that recognize the requester or the data being requested, and prioritize the request or the response accordingly.

JXTA (Juxtapose)

- open source peer-to-peer protocol specification begun by Sun Microsystems in 2001
- Build P2P technology with XML
- use self-describing XML definitions to move messages and manage the environment
- defines three layers.
  - The bottom layer addresses the communication and routing and P2P connection management.
  - The middle layer handles higher-level concepts, such as indexing, searching, and file sharing.
  - The top layer provides protocols that the applications use to manage the middle-layer services and lower-layer "plumbing" to build full-featured P2P applications.
    - The top layer is where typical applications, such as instant messaging, network services, and P2P collaboration environments are defined.
- all protocols are defined as XML messages sent between two peers.

JXTA messages

- messages define the protocols used to discover and connect peers and to access network services offered by peers and peer groups.
- Each JXTA message has a standard format, and may include optional data fields.
- JXTA standardizes the messages exchanged between peers by defining standard XML data streams used to invoke common functions or features of P2P services.
- Messages are sent between logical destinations (endpoints) identified by a URI.
- The transport must be capable of sending and receiving datagram-style messages.
- Endpoints are mapped into physical addresses by the messaging layer at runtime.