Peer-to-Peer Systems

Michael Welzl michael.welzl@uibk.ac.at

DPS NSG Team http://dps.uibk.ac.at/nsg
Institute of Computer Science
University of Innsbruck, Austria

What is Peer-to-Peer?

- What does the word "peer" mean?
  - Merriam-Webster: one that is of equal standing with another **:** EQUAL; *especially* **:** one belonging to the same societal group especially based on age, grade, or status
- P2P leverages the capabilities of end nodes
- "The 'P' in P2P is People" (Dave Winer, software pioneer (e.g. RSS))
- Not just file sharing!
- Ad hoc networks, where end nodes are routers too, are P2P systems
- The web would be P2P if browsers and servers weren't separated

P2P Principle

- P2P can be seen as an organizational principle
  - System exhibits P2P principle more or less clearly
- P2P principle applicable to many kinds of systems
  - Content distribution, communication, distributed computation, and collaboration
- Core concepts of the P2P principle:
  - Self-organizing, no central management
  - Resource sharing, e.g., files
  - Based on voluntary collaboration, e.g., Wikipedia
  - Peers in P2P are all equal (more or less)
  - Large number of peers in the network
- In contrast: Client-server = clearly defined roles for client and server

Definition of P2P

A P2P system exhibits the following characteristics:

1. High degree of autonomy from central servers
2. Exploits resources at the edge of the network
   - Storage, CPU cycles, human presence
3. Individual nodes have intermittent connectivity

- No strict requirements, instead typical characteristics
- Above characteristics allow us to distinguish P2P systems from other similar systems

What about the Grid?

- History: parallel processing at a growing scale
  - Parallel CPU architectures
  - Multiprocessor machines
  - Clusters
  - ("Massively Distributed") computers on the Internet
- GRID
  - logical consequence of HPC
  - metaphor: power grid
    just plug in, don't care where (processing) power comes from, don't care how it reaches you

- Common definition:
  The real and specific problem that underlies the Grid concept is coordinated resource sharing and problem solving in dynamic, multi institutional virtual organizations
  [Ian Foster, Carl Kesselman and Steven Tuecke, *"The Anatomy of the Grid – Enabling Scalable Virtual Organizations"*, International Journal on Supercomputer Applications, 2001]

Grid scope

- Definition quite broad ("resource sharing")
  - Reasonable - e.g., computers also have harddisks
  - But also led to some confusion - e.g., new research areas / buzzwords: Wireless Grid, Data Grid, Semantic / Knowledge Grid, Pervasive Grid, [*this space reserved for your favorite research area*] Grid
- Example of confusion due to broad Grid interpretation:

  "One of the first applications of Grid technologies will be in remote training and education. Imagine the productivity gains if we had routine access to virtual lecture rooms! (..) What if we were able to walk up to a local 'power wall' and give a lecture fully electronically in a virtual environment with interactive Web materials to an audience gathered from around the country - and then simply walk back to the office instead of going back to a hotel or an airplane?"
  [I. Foster, C. Kesselman (eds): "The Grid: Blueprint for a New Computing Infrastructure", 2nd edition, Elsevier Inc. / MKP, 2004]

⇒ Clear, narrower scope is advisable for thinking/talking about the Grid

- Traditional goal: processing power
  - Grid people = parallel people; thus, main goal has not changed much

Grid = next Web?

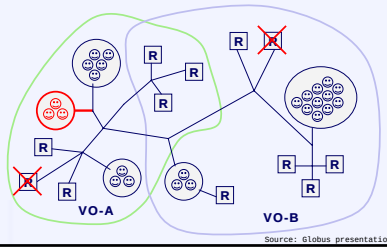
- Ways of looking at the Internet
  - Communication medium (email)
  - Truly large kiosk (web)
- The Grid way of looking at the Internet
  - Infrastructure for Virtual Teams
- Most of the time…
  - the "real and specific goal" is High Performance Computing
  - Virtual Organizations and Virtual Teams are well defined
    i.e. **not** an „open" system, e.g. security is a big issue
- Virtual Teams
  - Geographically distributed
  - Organizationally distributed
  - Yet work on a common problem

But Web 2.0 is already here :-)

It has been called "the next web"

Virtual Organizations and Virtual Teams

- Distributed resources and people
- Linked by networks, crossing admin domains
- Sharing resources, common goals
- Dynamic



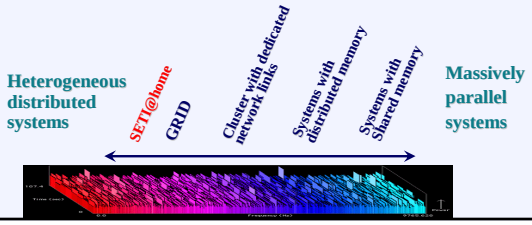
Source: Globus presentation by Ian Foster

The Grid and P2P systems

- Look quite similar
  - Goal in both cases: resource sharing
- Major difference: clearly defined VOs / VTs
  - No incentive considerations
  - Availability not such a big problem as in P2P case
    - It *is* an issue, but at larger time scales
      - (e.g. computers in student labs should be available after 22:00, but are sometimes shut down by tutors)
  - Scalability not such a big issue as in P2P case
    - …so far! ⇒ convergence as Grids grow
- *coordinated* resource sharing and *problem solving* in dynamic, multi institutional *virtual organizations* (*Grid*, P2P)

Case study: SETI@home

- Screen saver which donates CPU cycles to analyzing signals from space to find extraterrestrial intelligence
  - Not really P2P: calculation managed by central server
  - Not really a Grid: lack of coordination
  - … but successful; both communities claim that it's a P2P or Grid system



Properties of P2P Systems

P2P systems typically have the following properties:

1. Unreliable, uncoordinated, unmanaged
   - No central authority, peers are completely independent
   - Increases flexibility of individual peers, but makes the overall system (possibly) unreliable
2. Resilient to attacks, heterogeneous
   - Large number of peers in the system, hard to bring it down?
   - Heterogeneous peers make viruses and worms harder to write?
3. Large collection of resources
   - Voluntary participation, global reach
   - Millions of simultaneous users

P2P Vision

- P2P vision for the future:

  No More Dedicated Servers,
  Everything in Internet Served by Peers

- No mail servers, no file servers, no web servers
- Individual peers, operating independently from one another offer all the basic services
- Is this a realistic vision?

History

- We should learn from it
  - There are ancient P2P systems, people have been reinventing the wheel
  - You shouldn't :-) that's why we look at history
- Examples of historical P2P systems
  - Originally, every host on the Internet (FTP, Telnet: client/server application, but all hosts were clients+servers)
  - Usenet: grandfather of P2P
  - DNS: common example of a P2P service (sure, there are servers – and in Kazaa, there are supernodes…)
- Common Internet theme: virtualization
  - Decoupling entities – examples:
    - DNS decouples names from physical systems
    - URLs let users retrieve documents without knowing names of hosts
    - Virtual hosting, replicated servers relax one-to-one relationship of names to systems

Historical P2P examples: Usenet

- Origin: Unix-to-Unix-Copy Protocol (UUCP)
  - Now replaced with Network News Transport Protocol (NNCP)
- Distributed storage of selected subsets of all data (newsgroups)
  - Totally decentralized, no central authority in control
    selection of new newsgroups: based on democratic voting (news.admin group), but not for alt.* groups (anarchy)
  - Hierarchy: company newsserver: subset of content of ISP newsserver
  - Content restriction: major difference between Usenet and some P2P systems
- NNTP messages contain "Path" header; ensures that traversed newsservers do not get the same message again
  - Not included in all P2P systems, e.g. not in Gnutella
- Another lesson: Usenet lacks accountability of users
  - Spam, spam, spam, spam!

Historical P2P examples: DNS

- Origin: hosts.txt
  - Mapping of name to IP address
  - Everyone supposed to have the same file; add host = change all files
  - More efficient way to handle these data than sending this file around…
- DNS has a natural hierarchy
  - Domains, with per-domain authorities
  - Delegation of searches, with caching of answers for speedup, made this an efficient and scalable query system
    - Hosts can operate as clients and servers, propagating requests as needed
    - Any DNS server can be asked (if allowed), plus hierarchy has default path
- Very common example for a P2P system
  - Old distributed database which still works well

Napster, and what came after it

- Term P2P was coined by Shawn "Napster" Fanning in 1999
- Napster was a huge hit, brought P2P to general attention
- Illegal sharing of copyrighted material by users was the main driver behind Napster's success and the reason for its downfall
  - Ironically, lack of P2P structure (central server) made this possible
- Other systems followed Napster quickly
  - Gnutella addressed the Napster problem (no more server)
- Research community followed quickly
  - Many deployed systems proprietary, hard to examine well…

Current State in Research

- Lot of interest in P2P in the research world
  - Common to networking and distributed systems
- Strong focus so far on searching and locating objects in P2P networks
- Some work on replication, robustness, and security
- Higher level work on filesystems, P2P applications
  - See later chapters for examples
- Alas, P2P has become buzzword
  - Confusion about terminology, merging of different research communities
  - No commonly accepted definition of P2P

New P2P Systems

- File sharing was first P2P application
- Other applications are coming to light
- BitTorrent: focus more on content distribution than file sharing
  - Makes use of common research result (DHT) since 2005
- P2P extending beyond file sharing: Skype
  - Skype is a P2P telephone "system"
  - Can call other computers, or normal phones
  - Based on the KaZaA network
- P2P streaming systems
  - PPLive, PPStream

Some milestones: tools

- Napster (1999)
  - * 1999, † 2000 (court decision)
- Gnutella (2000)
  - New version (Gnutella 2) in 2002
- Edonkey (2000)
  - Later: **Overnet** based on Kademlia
- FreeNet (2000)
  - Main goal: anonymization
- JXTA (2001)
  - Open source Peer-to-Peer network platform
- FastTrack (2001)
  - Protocol underlying KaZaa, Morpheus, Grokster
- Bittorrent (2001)
  - Popular download system, no search facility

Some milestones: theory

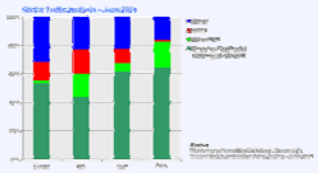
- Distributed Hash Tables (DHT) (1997)
  - Originally used for load balancing among web servers
- CAN (2001)
  - Efficient DHT based P2P data structure
- Chord (2001)
  - Efficient DHT based P2P data structure with O(log n) search time
- Pastry/Tapestry (2001)
  - P2P data structure based on Plaxton routing
- Kademlia (2002)
  - P2P-Lookup based on XOR-metric (similar to Plaxton routing)
- …and many other interesting networks (Viceroy, Distance-Halving, Koorde, Skip-Net, P-Grid, …)

Current State of P2P

- P2P networks going strong, all over the world
  - Many networks highly popular and widely used
  - Different networks in different countries
- P2P networks currently mostly used for illegal sharing of copyrighted material
  - Music, videos, software, …
  - Note: Can be used for legal sharing too (see BitTorrent)
- Other applications starting to emerge (see below)
- Content providers not so happy
  - Sue companies making P2P software (e.g., Napster), sue software developers (Winny), sue users sharing material
  - But also providing alternate means: iTunes & friends

P2P yesterday, today…

- June 2004
  - Source: CacheLogic



- 2005
  - More than 8 M active participants in P2P networks at any time
  - 10 Petabyte of data at any time
  - More than half of the Internet traffic is P2P
  - Several P2P networks shut down due to court decisions
  - People sued because of copyright violation
- 2007: for the first time, P2P traffic is less than web traffic ("YouTube beats BitTorrent")
  - Source: personal email from Prof. Christian Schindelhauer

… and tomorrow?

- Measurements vary depending on location
  - May also be different next year…
- So what is the trend?
- Shift to new (old?) paradigm of Internet usage
  - Every user = content producer; "Web 2.0"
  - Client/Server model doesn't; consider 1000s of users uploading content to single site, site owner decides to quit…
- P2P systems seem to match this communication model well
  - Do they? Consider efficiency vs accountability, reliability…

Technical mismatch: P2P these days

- Internet exploded in '90s
  - WWW took off
  - With it, browser (client) / webserver (server) model
  - If all data flows to clients, and end nodes are always clients…
    - why provide large upload bandwidth?
      ⇒ upload bw restrictions of ADSL, cable modems
      traffic engineering in ISP networks designed for asymmetric usage
    - why allow others to contact client (host) before client contacts them?
      ⇒ firewalls, dynamic IP addresses, NATs…
- Today's P2P systems would have worked better in yesterday's Internet!

Technical mismatch: what can be done?

- Firewalls
  - Tunnel through port 80
  - Stupid (see RFC 3093, the "Firewall Enhancement Protocol"); leads to arms race
  - Better: let end systems and firewalls communicate
- Dynamic addresses and NATs
  - IPv6 would solve the problem, but deployment difficult
  - P2P systems usually set up communication via intermediate host, then communicate
  - Also: create alternate address spaces
- Asymmetric bandwidth
  - Prevent unnecessary (re)transmissions by caching
    (note: distributed caches like Squid have worked out many consistency / load sharing issues that p2p apps face)
  - User should be in control of bandwidth usage
  - P2P creates a demand for uplink bandwidth; in the long run, ISPs may need to adapt

Dealing with today's users

- Usenet, email worked well when Internet users were well behaved
  - Now, Spam is everywhere!
  - Need Accountability: identify individuals, even if "pseudonymously" to preserve privacy (somewhat conflicting goal)
  - Should be able to prevent "freeriding"
  - Reputation tracking mechanisms help
  - Consider news.admin voting: such mechanisms make sense for P2P systems
- Significant effort went into accountability in P2P systems
  - Payment schemes (e.g. "mojo" in MojoNation)
  - Tit-for-tat scheme in BitTorrent

Why Does P2P Work?

Why are P2P file sharing networks so successful?

1. Easy to use
   - P2P software readily available, simple to use
2. Provide something useful (for free)
   - Until recently, only alternative to P2P content was "buy a CD"
   - Online music stores may change this?
3. Anyone can contribute
   - Contributions not tied to geographical location; user in Brazil can provide files for everyone (compare with ad hoc networks!)
   - Enough "altruistic" users to make P2P networks useful

- Some systems (Skype) completely hide the P2P-part
  - Will this become the future trend?

P2P: Traps and Pitfalls

- What could render current P2P networks useless?
  - In particular, file sharing networks

1. Removal of desirable content
   - Stricter enforcement of copyright laws?
2. Alternative ways of getting same content
   - Online music stores?
3. Blocking of P2P traffic by ISPs
   - Or making users pay for bandwidth they use?
4. Viruses or worms on P2P networks
   - Exploit bugs in P2P software

When 2 P2P and when not 2 P2P?

- So, when is P2P the right solution?
  - Or, when is it the wrong solution?
- *Claim:* Our earlier P2P vision is technically feasible
  - In other words, possible to build everything on Internet without any dedicated servers
- Just because it's technically feasible, doesn't make it sensible…
- In other words, just because we can do it P2P, doesn't mean that we should do it P2P
  - True in many areas of life…
- So, when *is* P2P the right solution?!?

Some Criteria

Let's consider the following criteria:

1. Budget
   - How much money do we have?
2. Resource relevance
   - How widely are resources interesting to users?
3. Trust
   - How much trust there is between users?
4. Rate of system change
   - How fast does "something" in the system change
5. Criticality
   - How critical is the service to the users

Analysis

Budget
- If you have enough money, build a centralized system
- Look at Google if you doubt this claim ;-)
  - Any system can be made to scale with enough money
- P2P is therefore useful when budget is not unlimited
  - In other words, most real-world situations…
  - For the rest of this analysis, we assume limited budget

Resource relevance
- If shared resources are highly relevant to a large number of users, P2P makes sense
- Easier to build a distributed solution when interest is widely spread

Analysis /2

Trust
- If other users can be trusted, P2P is a good solution
  - For example, corporate network or any closed network
- Building a fully distributed, trusted network is still very much a research problem (and may remain so…)

Rate of system change
- How are the system dynamics?
  - Rate of peers joining and leaving, rate of information change in system, rate of change in network topology, …
- If the rate of change is too high, a distributed P2P solution might not be able to keep up
- Again, research problem

Analysis /3

Criticality
- How important is the service to the users?
- If you "can live without it", P2P is acceptable
- If "it must work", then consider other solutions…

- Summary: P2P is good when:
  - Budget is limited
  - Resources have wide interest and relevance
  - Trust between participants is high
  - Rate of change is manageable
  - Criticality is low
- Note: Again, no need to fulfill every point!

Conclusion

- Peer-to-peer principle: self-organization and resource sharing

- P2P systems exhibit following characteristics:
  - Autonomy from central servers
  - Use of edge resources
  - Intermittent connectivity

- Hard to clearly define the limits of P2P
  - Compare with distributed systems and grid computing
  - Different people working in different areas have different definitions

References / acknowledgments

- Slides from Jussi Kangasharju and Christian Schindelhauer