peer-to-peer and agent-based computing

Scalability in P2P Networks

UNIVERSITY OF ABERDEEN

Plan of lecture

- Optimising P2P Networks:
  - Social networks and lessons learned
  - Are P2P networks social?
  - Organising P2P networks
- Peer Topologies
  - Centralised, Ring, Hierarchical & Decentralized
  - Hybrid:
    - Centralised/Ring
    - Centralised/Centralised
    - Centralised/Decentralised
  - Reflector Nodes
- Gnutella Case Studies

Social networks

Stanley Milgram (Harvard professor) – 1967 social networking experiment:
How many 'hops' would it take for messages to traverse US pop?

- Posted 160 letters to randomly chosen people in Omaha, Nebraska
- Asked them to try to pass these letters to a stockbroker working in Boston, Massachusetts

Rules:
1. use intermediaries they know on a first name basis, chosen intelligently!
2. make a note at each hop

- 42 letters made it !!
- Average of 5.5 hops
- Demonstrated the 'small world effect'

Proved that the social network of the United States is indeed connected with a path-length (number of hops) of around 6 – the 6 degrees of separation!

Does this mean that it takes 6 hops to traverse 200 million people?

Lessons learned from experiment

- Social circles are highly clustered
- Few members have wide-ranging connections
  - Form a bridge between far-flung social clusters
  - Bridging plays critical role in bringing the network closer together
  - For example:
    - 25% of all letters passed through a local shopkeeper
    - 50% mediated by just 3 people
- Lessons learned
  - These people acted as gateways or hubs between the source and the wider world
  - A small number of bridges dramatically reduces the number of hops

From social to computer networks

- Similarities between social/computer networks
  - People = peers
  - Intermediaries = hubs, gateways
  - Number of intermediaries = number of hops
- Are P2P networks special then?
  - P2P networks more like social networks than other types of computer networks because:
    - Self-organising
    - Ad-Hoc
    - Employ clustering techniques based on prior interactions (like we form relationships)
- Decentralised discovery & communication (similar to neighbourhoods, villages, etc)

P2P: what's the problem?

- How to organise peers in ad-hoc, multi-hop pervasive P2P networks?
  - Network of self-organising peers organised in a decentralised fashion
  - Networks may expand rapidly from few hundred to several thousand (or even millions) of peers

P2P: what's the problem? (Cont'd)

- P2P Environments:
  - Unreliable
  - Peers connect/disconnect – network failures
  - Random failures (e.g. power outages, DSL failure)
  - Personal machines more vulnerable than servers
  - Algorithms must cope with continuous restructuring of the network core
- P2P systems
  - Must treat failures as normal occurrences not freak exceptions
  - Must be designed to promote redundancy
    - Tradeoff with degradation of performance

How do we organise P2P networks?

- Organisation aimed at optimum performance
- NOT abstract numerical benchmarks!!
  - How many milliseconds will it take to compute this many millions of calculations?
- Rather, it means asking questions like:
  - How long will it take to retrieve this particular file?
  - How much bandwidth will this query consume?
  - How many hops will it take for my package to get to a peer on the far side of the network?
  - If I add/remove a peer to the network will the network still be fault tolerant?
  - Does the network scale as we add more peers?

Performance issues in P2P networks

3 main features that make P2P networks more sensitive to performance issues

1. Communication
   - Fundamental necessity
   - Users connected via different connections speeds
   - Multi-hop
2. Searching
   - No central control so more effort needed
   - Each hop adds to total bandwidth (w/ time outs!)
3. Equal Peers
   - "Freeriders" – unbalance in harmony of network
   - Degrades performance for others
   - Need to get this right to adjust accordingly

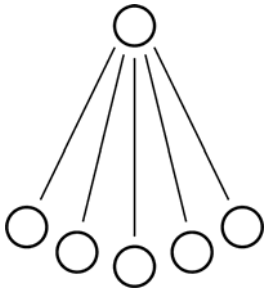
P2P topologies

Core
- Centralised
- Ring
- Hierarchical
- Decentralised

Hybrid
- Centralised-Ring
- Centralised-Centralised
- Centralised-Decentralised

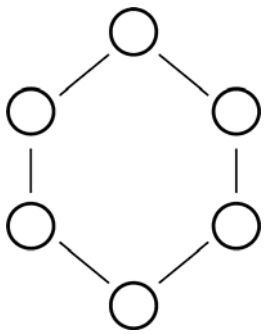
Centralised

- Client/server
- Web servers
- Databases
- Napster search
- Instant Messaging



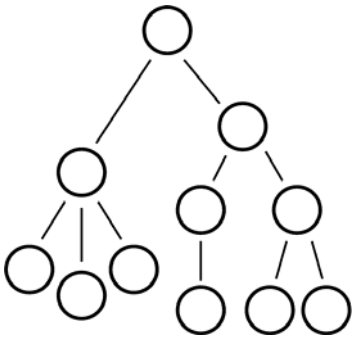
Ring

- Fail-over clusters
- Simple load balancing
- Assumption
  - Single owner



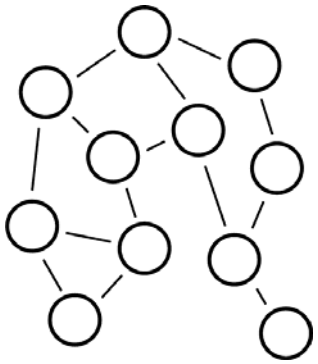
Hierarchical

- Tree structure
- DNS
- Usenet (sort of)



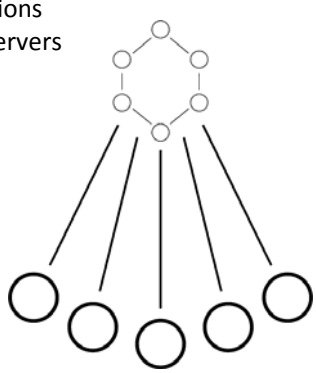
Decentralised

- Gnutella
- Freenet
- Internet routing



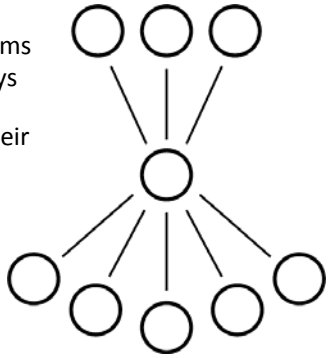
Centralised + Ring

- Robust web applications
- High availability of servers



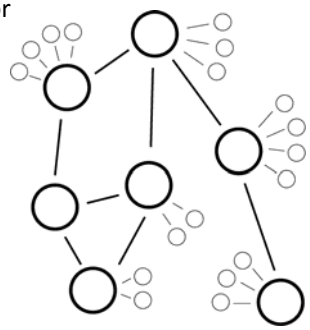
Centralised + Centralised

- N-tier apps
- Database heavy systems
- Web services gateways
- Google.com uses this topology to deliver their services



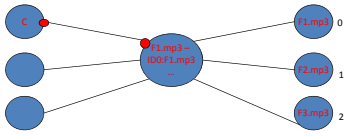
Centralised + Decentralised

- New generation of P2P
- Clip2 Gnutella Reflector
- FastTrack
  - KaZaA
  - Morpheus
- Email
- Possibly like social networks?

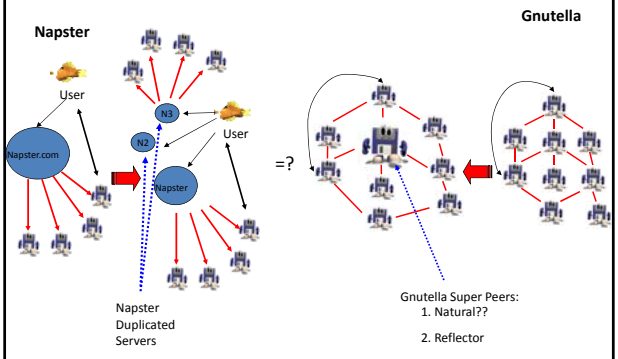


Reflector nodes

- Known as 'super peers'
  - AKA "rendezvous peers" (JXTA)
- Cache file list of connected users
  - Maintain an index
- When a query is issued, reflector node
  - Does NOT retransmit it
  - Answers the query using its own information

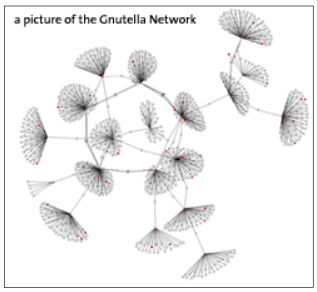


Napster = Gnutella?

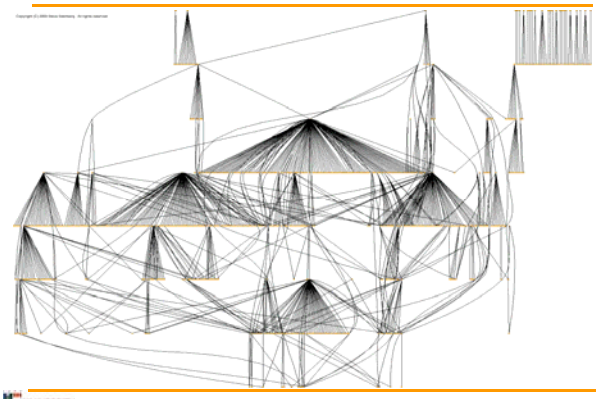


Gnutella network today

- Topology of a Gnutella network
  - From LimeWire site (popular Gnutella client)
- Notice power-law or centralised-decentralised structure



Another view of Gnutella



Gnutella studies 1: "free riding"

Two types of free-riding:

1. Download files but never provide any files for others to download
2. Users that have undesirable content

Gnutella studies 1: "free riding" (Cont'd)

- Article: "Free Riding on Gnutella", E. Adar and B.A. Huberman (2000):
  - 22,084 of the 33,335 peers in the network (66%) of the peers share no files
  - 24,347 or 73% share ten or fewer files
  - Top 1% (333 hosts) represent 37% of total files shared
  - 20% (6,667 hosts) sharing 98% of files
- Findings:
  - Even without Gnutella reflector nodes, the Gnutella network naturally converges into a centralised + decentralised topology with the top 20% of nodes acting as super peers

Gnutella studies 2: equal peers

- Studied Gnutella for one month
  - Noted an apparent scalability barrier when query rates went above 10 per second
- Why?
  - Gnutella query: 560 bits long
    - Queries make up approximately ¼ of traffic
  - Each peer is connect to three peers:
    - 560 × 10 × 3 = 16,800 bytes/sec
    - ¼ of traffic, so total traffic 67,200 bytes/sec
  - 56-K link cannot keep up with amount of traffic
    - One node connected in the incorrect place can grind the whole network to a halt
  - This is why P2P networks place slower nodes at the edges

Gnutella studies 3: communication

- Article: *Peer-to-Peer Architecture Case Study: Gnutella Network*, Matei Ripeanu
  - Studied topology of Gnutella over several months
- Two important findings
- Two suggestions

Gnutella studies 3: communication

Findings:

1. Gnutella network shares the benefits and drawbacks of a power-law structure
   - Networks that organise themselves so that most nodes have few links & small number of nodes have many links
   - Unexpected degree of robustness when facing random node failures.
   - Vulnerable to attacks: removing a few super nodes can have massive effect on function of network as a whole.
2. Gnutella network topology does not match well with underlying Internet topology
   - Leads to inefficient use of network bandwidth

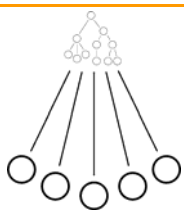
Gnutella studies 3: communication

Suggestions:

1. Use a software agent to monitor network and intervene by asking *servents* to drop/add links to keep the topology optimal.
2. Replace Gnutella flooding mechanism with a smarter routing and group communication mechanism

What about other topologies?

- Centralised + Hierarchical?
  - Back end tree of information
  - Caching architectures
- Decentralised + Ring?
  - P2P network of fail-over clusters
- What else?



Further reading

- Chapter of textbook
- *Free Riding on Gnutella*, E. Adar and B.A. Huberman (2000), First Monday 5(10)
  http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/792
- *Peer-to-Peer Architecture Case Study: Gnutella Network*, Matei Ripeanu
  http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=990433
- Distributed Hash table models
  - Pastry: Microsoft Research
    http://research.microsoft.com/~antr/pastry
  - Chord: chord
    http://en.wikipedia.org/wiki/Chord_(peer-to-peer)