P2P Systems
Client/Server Architecture

GET /index.html HTTP/1.0
HTTP/1.1 200 OK ...

Clients

Server
Peer-to-Peer Architecture

[Diagram showing a network of peer-to-peer connections with a gateway server.]

Peers

Gateway Server
The architectures

- Server-based architecture
  - Client-Server / Server-Cluster
  - Problems:
    - Limited resources
    - All loads are centered on the server
  - Server-based architecture has low scalability.
  - The setup and maintenance cost is high.

- Peer-to-Peer (P2P) architecture
  - Advantages:
    - Distributing loads to all users
    - Users consume and provide resources
  - P2P architecture has high scalability.
  - The setup and maintenance cost is low.
The Client Side

- Today’s clients can perform more roles than just forwarding users requests
- Today’s clients have:
  - more computing power
  - more storage space
- Thin client $\rightarrow$ Fat client
Evolution at the Client Side

- DEC’s VT100
  - No storage
  - ‘70

- IBM PC
  - @ 4.77MHz
  - 360k diskettes
  - ‘80

- PC @ 4-core 4GHz
  - 300GB HD
  - 2008
What Else Has Changed?

- The number of home PCs is increasing rapidly
- Most of the PCs are “fat clients”
- As the Internet usage grow, more and more PCs are connecting to the global net
- Most of the time PCs are idle

- How can we use all this?

*Peer-to-Peer (P2P)*
What is peer-to-peer (P2P)?

“Peer-to-peer is a way of structuring distributed applications such that the individual nodes have symmetric roles. Rather than being divided into clients and servers each with quite distinct roles, in P2P applications a node may act as both a client and a server.”

-- Charter of Peer-to-peer Research Group, IETF/IRTF, June 24, 2004
(http://www.irtf.org/charters/p2prg.html)
Resources Sharing

- What can we share?
  - Computer resources

- Shareable computer resources:
  - CPU cycles - seti@home, GIMPS
  - Bandwidth - PPLive, PPStream
  - Storage Space - OceanStore, Murex
  - Data - Napster, Gnutella
  - People - Buddy Finder
  - Camera, Microphone, Sensor, Service???
SETI@Home

- SETI – Search for Extra-Terrestrial Intelligence
- @Home – On your own computer
- A radio telescope in Puerto Rico scans the sky for radio signals
- Fills a DAT tape of 35GB in 15 hours
- That data have to be analyzed
The problem – analyzing the data requires a huge amount of computation

Even a supercomputer cannot finish the task on its own

Accessing a supercomputer is expensive

What can be done?
SETI@Home (cont.)

- Can we use distributed computing?
  - YEAH

- Fortunately, the problem can be solved in parallel - examples:
  - Analyzing different parts of the sky
  - Analyzing different frequencies
  - Analyzing different time slices
SETI@Home (cont.)

- The data can be divided into small segments
- A PC is capable of analyzing a segment in a reasonable amount of time
- An enthusiastic UFO searcher will lend his spare CPU cycles for the computation
  - When? Screensavers
SETI@Home - Example
SETI@Home - Summary

- SETI reverses the C/S model
  - Clients can also provide services
  - Servers can be weaker, used mainly for storage
- Distributed peers serving the center
  - Not yet P2P but we’re close
- Outcome - great results:
  - Thousands of unused CPU hours tamed for the mission
  - 3+ millions of users
GIMPS Home Page

September 2006: New Mersenne Prime!

44th Known Mersenne Prime Found!!

Lightning strikes twice. On September 4, 2006, in the same room just a few feet away from their last find, Dr. Curtis Cooper and Dr. Steven Boone's CMSU team broke their own world record, discovering the 44th known Mersenne prime, $2^{32,582,657} - 1$. The new prime at 9,808,358 digits is 650,000 digits larger than their previous record prime found last December. However, the new prime falls short of the 10 million digits required for GIMPS to claim the Electronic Frontier Foundation $100,000 award.

With five record primes found in less than 3 years, GIMPS has been on an incredible lucky streak. Never before have Mersenne primes been bunched so closely together. When looking at the exponents, we expect only 1.78 Mersenne primes between powers of two, and prior to 2003, a maximum of 3 Mersenne primes were found between powers of two. The last 5 Mersenne prime exponents all fell between $2^{24}$ and $2^{25}$ -- and we haven't finished testing all the exponents in that range!

The new prime was independently verified in 6 days by Tony Reix of Bull S.A. in Grenoble, France using 16 Itanium2 1.5 GHz CPUs of a Bull NovaScale 6160 HPC at Bull Grenoble Research Center, running the Glucas program by Guillermo Ballester Valor of Granada, Spain.

Dr. Cooper and Dr. Boone could not have made this discovery alone. In recognition of contributions made by the project coordinators and the tens of
August/September 2008: New Mersenne Primes!

45th and 46th Known Mersenne Primes Found!!!!

GIMPS set to claim $100,000 EFF award!

Download free software

On August 23rd, a UCLA computer discovered the 45th known Mersenne prime, $2^{43,112,609} - 1$, a mammoth 12,978,189 digit number! The prime number qualifies for the Electronic Frontier Foundation's $100,000 award for discovery of the first 10 million digit prime. Congratulations to Edson Smith, who was responsible for installing and maintaining the GIMPS software on the UCLA Mathematics Department's computers.

On September 6th, the 46th known Mersenne prime, $2^{37,156,667} - 1$, a 11,185,272 digit number was found by Hans-Michael Elvenich in Langenfeld near Cologne, Germany! This was the first Mersenne prime to be discovered out of order since Colquitt and Welsh discovered $2^{110,503} - 1$ in 1988.

The nearly decade long quest for the EFF award came down to a close race to the finish - with just two weeks separating the discovery of the two primes.

As promised, GIMPS will give $50,000 of the EFF award to the UCLA Mathematics Department for discovering the first 10 million digit prime. $25,000 will go to charity, and most of the remainder will go to discoverers of the previous six Mersenne primes.
EFF Cooperative Computing Awards

Cooperative Computing Awards

"Rules" « Frequently Asked Questions » Status » Resources

The Electronic Frontier Foundation (EFF), the first civil liberties group dedicated to protecting the health and growth of the Internet, is sponsoring cooperative computing awards, with over half a million dollars in prize money, to encourage ordinary Internet users to contribute to solving huge scientific problems.

Through the EFF Cooperative Computing Awards, EFF will confer prizes of:

- $50,000 to the first individual or group who discovers a prime number with at least 1,000,000 decimal digits (awarded Apr 6, 2000)
- $100,000 to the first individual or group who discovers a prime number with at least 10,000,000 decimal digits
- $150,000 to the first individual or group who discovers a prime number with at least 100,000,000 decimal digits
- $250,000 to the first individual or group who discovers a prime number with at least 1,000,000,000 decimal digits

(Prize money comes from a special donation provided by an individual EFF supporter, earmarked specifically for this project. Prize money does NOT come from EFF membership dues, corporate or foundation grants, or other general EFF funds.)

EFF hopes to spur the technology of cooperative networking and encourage Internet users worldwide to join together in solving scientific problems involving massive computation. EFF is uniquely situated to sponsor these awards, since part of its mission is to encourage the harmonious integration of Internet innovations into the whole of society.

"The EFF awards are about cooperation," said John Gilmore, EFF co-founder and project leader for the awards. "Prime numbers
MUREX: A Mutable Replica Control Scheme for Peer-to-Peer Storage Systems
Murex: Basic Concept
Peer-to-Peer Video Streaming

Video stream

...
Peer-to-Peer Video Streaming
Napster -- Shawn Fanning
UNLIMITED ACCESS TO 1,000,000+ SONGS

Here’s what you get with your Napster Membership:

- Listen to and download an unlimited amount of music
- Enjoy 50+ commercial-free, interactive radio stations
- Discover and share music with other Napster Members
- With Napster To Go, you can fill and refill any compatible MP3 player without paying per song or album
- Fast, safe and legal

All the music you want. Any way you want it.
History of Napster (1/2)

- 5/99: Shawn Fanning (freshman, Northeastern University) founds Napster Online (supported by Groove)
- 12/99: First lawsuit
- 3/00: 25% Univ. of Wisconsin traffic on Napster
History of Napster (2/2)

- 2000: estimated 23M users
- 7/01: simultaneous online users 160K
- 6/02: file bankrupt
- …
- 10/03: Napster 2 (Supported by Roxio) (users should pay $9.99/month)

1984~2000, 23M domain names are counted vs. 16 months, 23M Napster-style names are registered at Napster
Napster Sharing Style: hybrid center+edge

1. Users launch Napster and connect to Napster server

2. Napster creates dynamic directory from users’ personal .mp3 libraries

3. **beastieboy** enters search criteria

4. Napster displays matches to **beastieboy**

5. **beastieboy** makes direct connection to **kingrook** for file transfer

<table>
<thead>
<tr>
<th>Title</th>
<th>User</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>song1.mp3</td>
<td>beastieboy</td>
<td>DSL</td>
</tr>
<tr>
<td>song2.mp3</td>
<td>beastieboy</td>
<td>DSL</td>
</tr>
<tr>
<td>song3.mp3</td>
<td>beastieboy</td>
<td>DSL</td>
</tr>
<tr>
<td>song4.mp3</td>
<td>kingrook</td>
<td>T1</td>
</tr>
<tr>
<td>song5.mp3</td>
<td>kingrook</td>
<td>T1</td>
</tr>
<tr>
<td>song5.mp3</td>
<td>slashdot</td>
<td>28.8</td>
</tr>
<tr>
<td>song6.mp3</td>
<td>kingrook</td>
<td>T1</td>
</tr>
<tr>
<td>song6.mp3</td>
<td>slashdot</td>
<td>28.8</td>
</tr>
<tr>
<td>song7.mp3</td>
<td>slashdot</td>
<td>28.8</td>
</tr>
</tbody>
</table>
Gnutella History

- Gnutella was written by Justin Frankel, the 21-year-old founder of Nullsoft.
- (Nullsoft acquired by AOL, June 1999)
- A day later AOL yanked Gnutella, at the behest of Time Warner.
- Too late: 23k users on Gnutella
- People had already downloaded and shared the program.
- Gnutella continues today, run by independent programmers.
Gnutella -- Justin Frankel and Tom Pepper
Gnutella Protocol

Scenario: Joining Gnutella Network

- The new node connects to a well known ‘Anchor’ node or ‘Bootstrap’ node.
- Then sends a PING message to discover other nodes.
- PONG messages are sent in reply from hosts offering new connections with the new node.
- Direct connections are then made to the newly discovered nodes.
Topology of a Gnutella Network
Gnutella: Issue a Request
Gnutella: Flood the Request
Gnutella: Reply with the File

Fully distributed storage and directory!
Searching in Gnutella involves broadcasting a Query message to all connected peers. Each connected peer will send it to their connected peers (say 3) and so on. Typically, this search will run 7 hops. If the number of connected peers, $c=3$ and the hops i.e. TTL=7 then the total number of peers searched (in a fully connected network) will be:

$$S = c + c^2 + c^3 + \ldots + c^7 = 3 + 9 + 27 + 81 + 243 + 729 + 2187 = 3279 \text{ Nodes}$$
MMOGs
Massively Multiplayer Online Games

- MMOGs are growing quickly

- Multi-billion dollar industry
- 10 million subscribers for *World of Warcraft*
- 600,000 concurrent users
Google Earth
To HsinChu..
and NTHU..
It is going to be 3D
NTHU 3D Student Center...
To the ground
3-Dimensional Virtual Tourism

- Google Earth
- MS Virtual Earth
- NASA World Wind
- Web3D X3D Earth
DARPA SIMNET: screenshot From Bruce Sterling's "War is Virtual Hell," (1993).
CALVIN: a distributed collaborative virtual environments for architectural layout designs (1996)
P2P and MMOGs
Model for MMOGs

- Many *nodes* on a 2D plane
- An avatar needs to know only those within Area of Interest (AOI)

★: self  ▲: neighbors

Area of Interest (AOI)
P2P AOI Neighbor Discovery

We need to solve the neighbor discovery problem in a fully-distributed, message-efficient manner with specific goals:

- **Consistent**
  → Good neighborship consistency

- **Scalable**
  → Limited overhead

- **Responsive**
  → Short latency
Neighborship Consistency (1)

Definition

\[
\frac{\text{no. of current AOI neighbors observed}}{\text{no. of current actual AOI neighbors}}
\]
Neighborship Consistency (2)

- An example

Neighborship Consistency = $\frac{4}{5} = 80\%$
Voronoi-based Overlay Network: VON

- Use Voronoi diagram to solve the neighbor discovery problem
  - Each node constructs a Voronoi diagram of its neighbors
  - Identify *enclosing* and *boundary* neighbors
  - Mutual collaboration in neighbor discovery
Voronoi Diagram

- 2D Plane partitioned into *regions* by *nodes*, each region contains all the points closest to its node.
Voronoï-based Overlay Network: VON

- • node $i$ and the big circle is its AOI
- ■ enclosing neighbors
- ▲ boundary neighbors
- ★ both enclosing and boundary neighbors
- ▼ normal AOI neighbors
- ◆ irrelevant nodes
Procedure (MOVE)

1) Positions sent to all neighbors, mark messages to B.N. 
   B.N. checks for overlaps between mover’s AOI and its E.N.

2) Connect to new nodes upon notification by B.N.
Procedure (LEAVE)

1) Simply disconnect
2) Others then update their Voronoi diagram
   new B.N. is discovered via existing B.N.
Procedure (JOIN)

1) *Joining node* sends coordinates to any *existing node*
   Join request is forwarded to *acceptor*

2) *Acceptor* sends back its own *neighbor list*
   *Joining node* connects with other nodes on the list
AOI-Cast

- A node has to send a message to all nodes within its AOI.
- AOI-Cast is a scoped multicast
VoroCast and FiboCast

- We propose two forwarding AOI-cast schemes to reduce the peak bandwidth consumption
  - VoroCast
    - No redundant message
    - Low latency
  - FiboCast
    - An extension of VoroCast
    - Adjusting the message forwarding frequency according to the hop-distance dynamically
VoroCast

// node x selects a child node to forward r’s messages
SelectChild(x, r)
// find out which child y would select x as its parent
for each y \in (\{(x.N - x.P - r.N) \cap r.AN\})
\[ z = \text{min\_dist}(y.N, r); \]
if (z equals x)
\[ \text{add } y \text{ as a child of } x; \]
Users in NVEs may pay more attention to activities that are more obvious in the vicinity. (u sees the central node more clearly than v.)

We can adaptively adjust the transmission frequency so that neighbors with more hop counts away receive messages less frequently.
FiboCast

Two variables in a message:
- current hop count (chc): increased each hop
- maximal hop count (mhc): the message is dropped when chc = mhc

mhc is set by a segment of a Fibonacci sequence with the last being ∞ and the first being at least 2 in a round-robin manner

E.G.: For a Fibonacci sequence segment <0, 1, 1, 2, 3, 5, 8>, the maximal hop counts would be 2, 3, 3, 4, 5, 7, 10, ∞, 2, 3, 3, 4, 5, 7, 10, ∞, 2, 3, 3, 4, 5, 7, 10, ..., etc.
Communications in MMOGs

- Players often *type* and *read* text instead of *speaking* and *listening*
- Teamspeak, Ventrilo, Skype
  - Set your own voice server
  - Group chat for 5 (Skype) to ~10 (Teamspeak)
- But…
  - Need to set up dedicated server(s)
  - Support only *limited, fixed user group*
  - Explicit group setup and joining
AOI voice chatting

- Each player listens and talks to all users within its **AOI** (area of interest)
  - More natural
  - More realistic
A simple approach

- NimbusCast:
  - MMOG system first provides a list of AOI neighbors
  - individual voice packets for all AOI neighbors
But... bandwidth overload occurs

- Number of supportable users very limited
  - $256 \text{ kbps} / 16 \text{ kbps} = 16 \text{ users (theoretical maximum)}$
**Human conversation model**

- Conversations are made of short bursts of voices
  - *Talkspurts*: single talk & double talk
  - *Pauses*: gaps in speech or mutual silence

- A person only talks 40% of time
- Talking rate of a listener is 6%

---

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Talk-spurt</td>
<td>38.53</td>
</tr>
<tr>
<td>Pause</td>
<td>61.47</td>
</tr>
<tr>
<td>Double talk</td>
<td>6.59</td>
</tr>
<tr>
<td>Mutual silence</td>
<td>22.48</td>
</tr>
</tbody>
</table>

[ITU-T, 2003]
Our proposal

- **Observation**: idle bandwidth can be used to forward voice packets, supporting more users

- **Under two goals**:
  - Reasonable delay [ITU-T P.59]
    - 150~250ms (good)  \quad 400ms (tolerable)
  - No bandwidth overload:
    - Voice bursts should not exceed bandwidth limit
QuadCast

- Quadrant-based Forwarding
  - A list of AOI neighbors is first obtained from the MMOG system
  - Voice packets sent to *forwarding assistants* (FAs)
  - FAs then forwards to remaining AOI neighbors in each quadrant
  - A *recipient list* is attached to the packet, and continuously divided

![Diagram showing quadrant-based forwarding]

- A list of AOI neighbors is first obtained from the MMOG system
- Voice packets sent to *forwarding assistants* (FAs)
- FAs then forwards to remaining AOI neighbors in each quadrant
- A *recipient list* is attached to the packet, and continuously divided
Packet aggregation

- Different packets might be sent to the same target:
  - Header sharing (HS)
  - Voice mixing (VM)
Player clustering in MMOGs

- In MMOGs, players are usually grouped at some hot-spots
- End-to-end latency in some quadrants may be long

(b) Balanced player grouping.
SectorCast

- Similar to QuadCast except in player grouping
- Players are sorted and grouped into four *sectors* with equal sizes according to *polar angles*
Alternatives of the recipient list

- Delivering the recipient list consumes much bandwidth.

<table>
<thead>
<tr>
<th>RTP header</th>
<th>UDP header</th>
<th>IP header</th>
<th>Voice content</th>
<th>Recipient list</th>
</tr>
</thead>
</table>

Network Bandwidth  ➔  Computation
Recipient list calculation in QuadCast

- The recipient list is replaced by the source player’s position.
- When the voice packet is forwarded, the position of FA is attached to the packet.
Recipient list calculation in SectorCast

- The recipient list is replaced by source’s position and a sector range.
- When a voice packet is forwarded, the FA’s position and the sector range is attached to the packet.
3D Streaming

- Continuous and real-time delivery of 3D contents over network connections to allow user interactions without a full download.

- Contents are fragmented, transmitted, reconstructed, and then displayed.
3D streaming vs. media streaming

- Video / audio media streaming is very matured

- User access patterns are different for 3D content
  - Highly interactive → Latency-sensitive
  - Behaviour-dependent → Non-sequential
4 types of 3D streaming

- Object streaming
- Scene streaming
- Visualization streaming
- Image-based streaming
Object streaming

Hoppe 1996
Progressive Meshes
Scene streaming

- Many objects
- Object selections & transmissions

Teler & Lischinski
2001
P2P-based 3D Streaming

Models & assumptions
- Many 3D objects (position, orientation)
- User navigations with AOI visibility
- Objects are fragmented (base & refinement pieces)
- Data are initially stored at server
Observation

- AOI neighbors share content in memory
- Former AOI neighbors share content in disk
FLoD: Flowing Level-of-Details

- Assume P2P-NVE overlay, such as VON
- Download 3D content from AOI neighbors

Basic design
- Each object has a unique ID and associated progressive mesh data
- Scene description records object ID, orientation, and scale
- World is partitioned into cells
Steps of FLoD

1. Peers exchange incremental formation of cached 3D data periodically
2. A peer sends requests for downloading scene descriptions
3. A peer sends requests for downloading object data
4. A peer asks the server if none of the peers responds
Object Prioritization

- Visual importance

\[ I_o = \left( \frac{D_{o,\text{max}} - D_o}{D_{o,\text{max}}} \right)^2 \exp^{-K_o|\theta_o|} \]
Peer selection

- Multi-level request
Prototype experiment

- Cell-based construction
- Use an actual game scene
- 100 x game scene (514KB -> 51.8MB)
Fragmentation
Related Publication


Related Publication

- **NETWORKING 2007**: Mo-Che Chan, Shun-Yun Hu, Jehn-Ruey Jiang, “An Efficient and Secure Event Signature (EASES) Protocol for Peer-to-peer Massively Multiplayer Online Games”
- **CDS 2007**: Jehn-Ruey Jiang, Jiun-Shiang Chiou and Shun-Yun Hu, “Enhancing Neighborship Consistency for Peer-to-Peer Distributed Virtual Environments”
- **P2P-NVE 2007**: ehn-Ruey Jiang and Hung-Shiang Chen, “Peer-to-Peer AOI Voice Chatting for Massively Multiplayer Online Games”
- **NIME 2008**: Shun-Yun Hu, Shao-Chen Chang, and Jehn-Ruey Jiang, “Voronoi State Management for Peer-to-Peer Massively Multiplayer Online Games”
- **INFOCOM 2008**: Shun-Yun Hu, Ting-Hao Huang, Shao-Chen Chang, Wei-Lun Sung, Jehn-Ruey Jiang, and Bing-Yu Chen, “FLoD: A Framework for Peer-to-Peer 3D Streaming”
- **CDS 2008**: Jehn-Ruey Jiang, Yu-Li Huang and Shun-Yun Hu, “Scalable AOI-Cast for Peer-to-Peer Networked Virtual Environments”
- **MMVE 2008**: Guan-Yu Huang, Shun-Yun Hu, and Jehn-Ruey Jiang, "Scalable Reputation Management for P2P MMOGs”
- **NOSSDAV 2008**: Wei-Lun Sung, Shun-Yun Hu and Jehn-Ruey Jiang, “Selection Strategies for Peer-to-Peer 3D Streaming”
Thank you!
Adaptive Computing and Networking Lab.

http://acnlab.csie.ncu.edu.tw