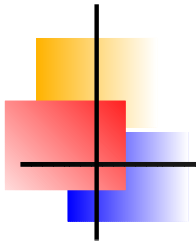
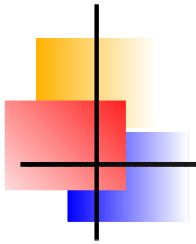


Deploying and Optimizing Squid in a Filesharing Application

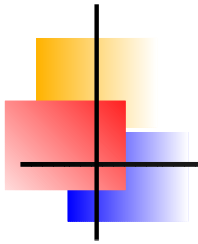
Dan Halperin

DIMACS REU 2004
with Dr. Manish Parashar
and Cristina Schmidt



Outline

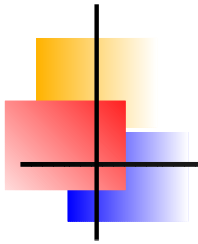
- Peer-to-Peer (P2P) Background
- Differing P2P Models
  - Napster
  - Gnutella
  - Squid
- My project



P2P Background

Main Idea: Peers (i.e. networked machines - computers, cell phones, palms, etc) on a network connect directly to each other instead of through a centralized server.

Why? Scalability, speed, fault tolerance, etc.

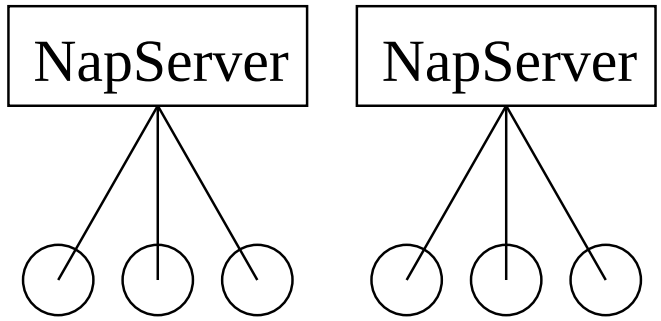


P2P Background - Examples

Hardware sharing: the GRID computing project, "technologies that provide seamless and scalable access to wide-area distributed resources" (1) (e.g. supercomputers, clusters, storage, data, instruments).

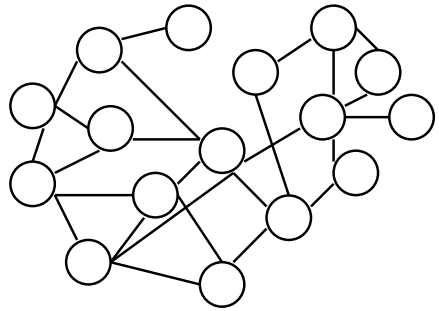
Information Sharing: share network information, services, and resources without centralized servers (e.g. filesharing such as Gnutella).

Napster P2P Model



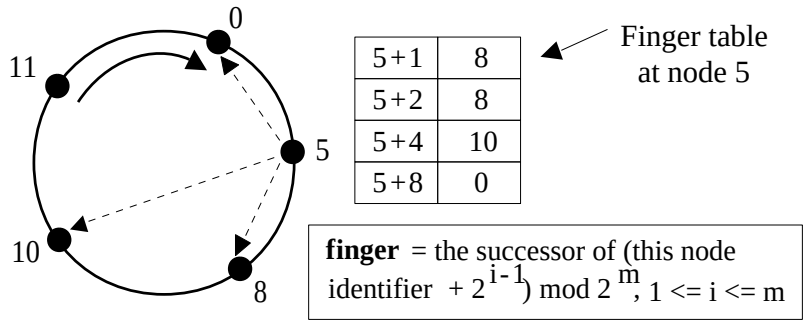
- Servers index all clients' data.
- Clients search to servers, download P2P.
- **Pro**: complex queries
- **Con**: not scalable, central point of failure, fraction of the network is searchable

Gnutella P2P Model

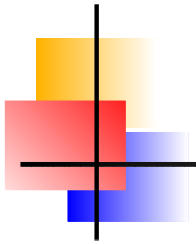


- Peers index own data.
- Search propagated via flooding.
- **Pro:** fully scalable, supports complex queries.
- **Con:** no search guarantees, scarcity compounded, repeated queries.

Squid P2P Model



- Structured overlay - efficient peer lookup.
- Index in Distributed Hash Table (DHT).
- Search is hashed, then DHT is queried.
- **Pro:** *guaranteed results*, complex queries (keywords, wildcards, ranges).
- **Con:** high overlay maintenance costs.



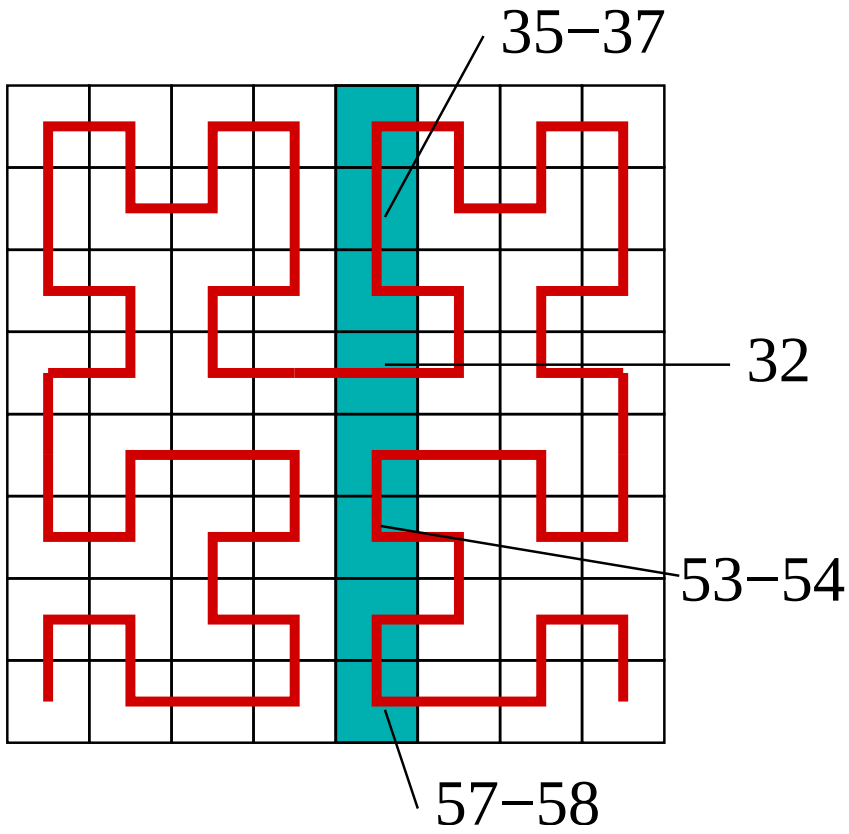
Squid

How Squid uses the DHT

- d keywords form a d-D keyword space $\mathcal{K}$.
- Peers given IDs in $[0, R]$ (1-D Peer ID space $\mathcal{P}$). (R = |maximum keyword| ^{d})
- Hash function $H : \mathcal{K} \rightarrow \mathcal{P}$ yields location to search for results for any query.
- Overlay provides efficient routing.

Squid

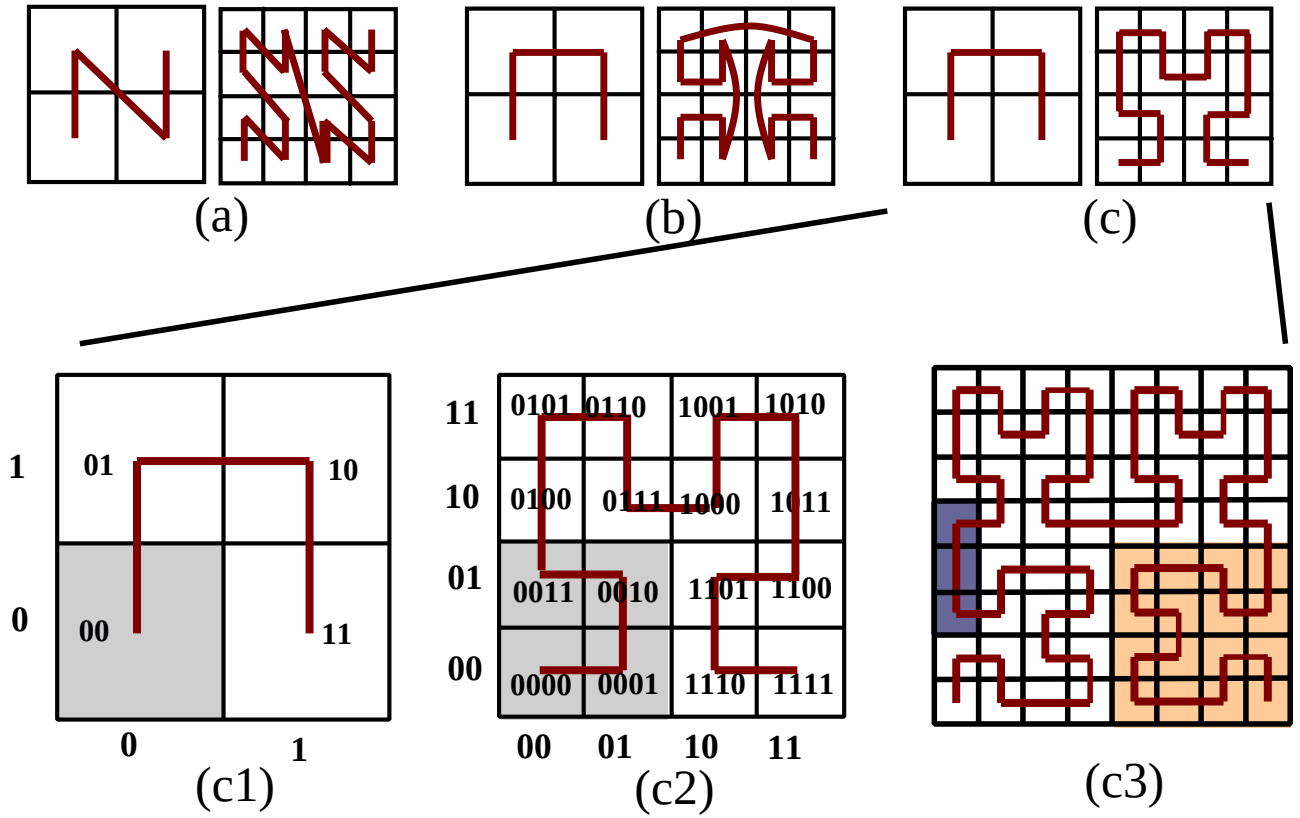
Hash Function: Space-Filling Curves (SFCs)



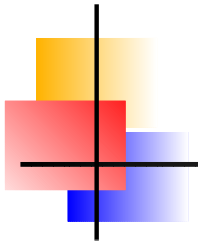
The Hilbert SFC fills $\mathcal{K}$. Query covers some subregion of $\mathcal{K}$, and will contain some parts of the curve. The indices along the curve are the images of the hash function in $\mathcal{P}$.

Squid

Some SFCs and Hilbert Refinement



Curve locality.



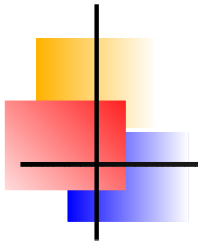
My Project

For a given application,

- What is the best SFC?
- What is the best overlay?
- What other optimizations can we make?

The best way to approach this:

- What application-specific assumptions can we make?
- How can we exploit them?



Filesharing Application

We're creating a filesharing application.

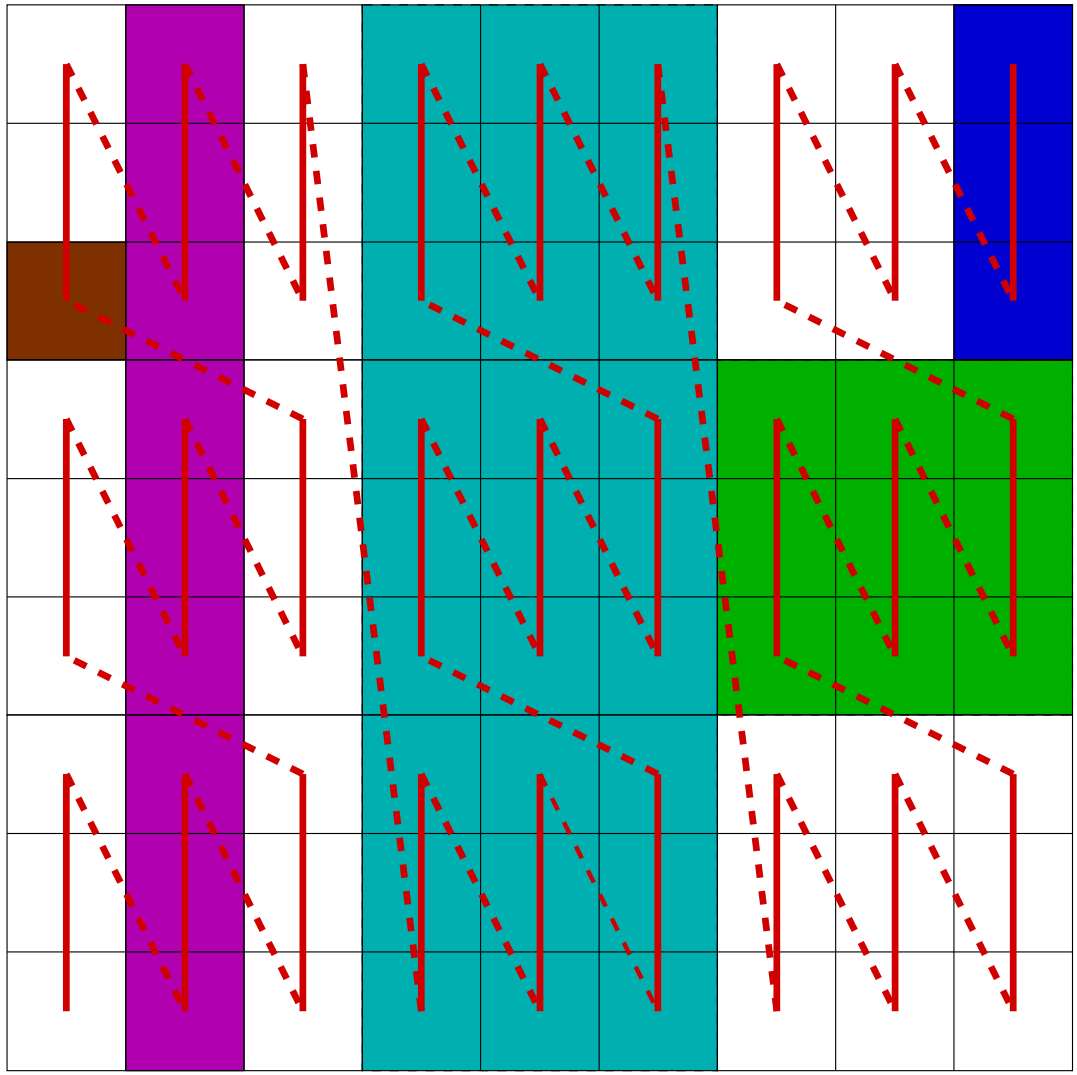
Assumptions we can make:

- Interchangeable axes.
- No ranges, except for wildcards.

Idea: Use a modified Z-curve for the SFC.

Motivation: Do not need to preserve locality in all directions. Use simpler, more customizable Z-curve.

Sample Z SFC searches



(b*,**)

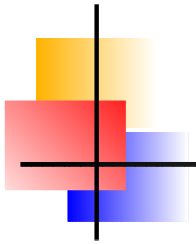
(ab,**)

(c*,b*)

(cc,c*)

(aa,ca)

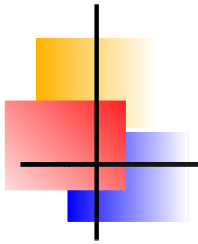
~~(**,ab)~~



Z-Curve

Advantages:

- Curve follows query shapes.
- Simple to generate.
- Easy hash: (“cat”, “dog”, “***”) $\rightarrow$ $cd * ao * tg *_{26}$.
- Customizes to alphabet.
- Count clusters: $|\Sigma|$ ^ (# internal stars).



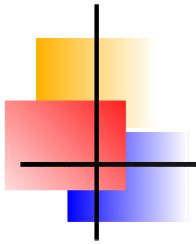
References

(1) GRID Computing Web Page. *http://www.gridforum.org*.

(2) A. Oram, ed. *Peer-To-Peer: Harnessing the Power of Disruptive Technologies*, O'Reilly & Associates, Inc., 2001.

(3) C. Schmidt and M. Parashar, "Enabling Flexible Queries with Guarantees in P2P Systems," *IEEE Internet Computing* (May-June 2004), pp.19-26.

(4) C. Schmidt and M. Parashar, "Peer to Peer Information Storage and Discovery Systems," forthcoming.



Acknowledgments

- DIMACS REU program
- Rutgers University
- Dr. Manish Parashar
- Cristina Schmidt