Paper Code: P2P-2A

Interest-Based Content Location in Peer-to-Peer Systems

Kunwadee Sripanidkulchai, Bruce Maggs, Hui Zhang

Carnegie Mellon University

Presenter: Xuebin Xu

Objectives of the paper

- Present an efficient content location algorithm by using interest-based locality in Peer-to-Peer distributed system;
- Compare interest-based locality with other methods and analyze the characteristics of interest-based locality.

What is Peer-to-Peer distributed system[1] ?

- systems in which all nodes have identical responsibilities and all communication is symmetric, i.e. decentralization;
- Advantages over traditional distributed systems: automatic load balancing and self-organization.
- Successful applications of these systems includes: content sharing systems and large-scale storage systems.

Problem:

- Part of the success of these systems comes from their ability to harness idle storage and network resources, offered by everyone who is willing to participate in the system. Unfortunately, such resources are inherently unreliable: we cannot control what code is running in these machines, and they will join and leave the network frequently.
- How to solve the problem, i.e. locate the resource efficiently ?
  Answer: **contents lookup algorithms**

Contents Look-up Algorithms

Contents look-up algorithm: algorithm used to locate content in the peer-to-peer system.

Two classes of **algorithms** are currently used:

- Unstructured content location – relies on flooding queries to all peers;
- Another class of protocols based on the Distributed Hash Table (DHT) abstraction.

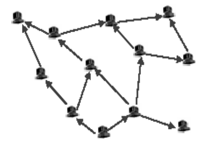
My topics of Today:

A contents look-up algorithm: **interest-based content look-up algorithm**[2];

Flood Search algorithm

- Unstructured content location;
- Peers organize into an overlay;
- To find content, a peer sends a query to its neighbors on the overlay;
- In turn, the neighbors forward the query on to all of their neighbors until the query has traveled a certain radius;
- If a node having already forwarded a query receives the same query again, it will not forward the second time;
- Get a set of peers which contain the interested contents at last.

Con't



- A query initiated by the peer at the bottom is flooded to all peers in the system;
- **Adv**: simple and robust;
- **Dis-adv**: a lot of traffic in the network.

Interest-based shortcuts

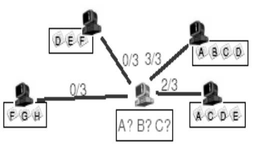


Fig. 1. Peers that share interests.

one powerful principle:

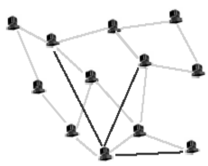
if a peer has a particular piece of content that one is interested in, then it is likely that it will have other piece of content that one is also interested in, i.e. the peers exhibit ***interest-based locality***.

Con't

Heuristic to detect shared interests:

- A new peer first attempt to locate content through flooding;
- The peer select a peer from the return set and add it into its shortcut list;
- Later, its subsequent query go through the shortcut list first. If cannot find the content, the peer issues a flood query to add another peer into its shortcut list;
- Shortcuts are added or removed based on some principle, such as utility. Shortcuts with low utility are removed from the list when the list is full.

Con't



(b) Shortcuts.

- A flood overlay with 3 shortcut links for the bottom-most peer. A query is flooded to the entire system only when none of the shortcuts have the content.

Benefits of shortcuts

Shortcuts are implemented as a separate performance enhancement layer on top of existing content location mechanisms, such as flooding.

Two-fold benefits:

- Shortcuts are modular in that they can work with any underlying content location scheme;
- Shortcuts only serve as performance-enhancement hints.

Thus, if it does not work, source can always located by using underlying overlay.

Performance evaluation

<u>Five Performance indices</u>:

- *Success rate*: How often queries resolve through shortcuts?
- *Load characteristics*: How many query packets do peers process while participating in the system?

Con't

- *Query scope*: For each query, what fraction of peers in the system are involved in query processing?
- *Minimum reply path lengths*: How long does it take for the first reply to come back?
- *Additional state*: How much additional do peers need to maintain in order to implement shortcuts?

Experiment design

- Gnutella file-sharing system:
  - flooding mechanism used to locate content;
  - each query packet has maximum Time-to-Live (TTL) tags;
  - a duplicate query mechanism used to detect duplicate query.

Con't

- **Query workloads**: 5 diverse traces of download requests from real content distribution applications to generate query workloads.
  - Boeing trace = one-day traces from 5 of Boeing's firewall proxies from March 1,99;
  - Microsoft trace = one-day traces from MS firewall proxies from Oct. 22, 01;
  - CMU-Web, CMU-Kazaa and CMU-Gnutella traces = passively monitor the traffic between CMU and the Internet over 24-hour period on Oct. 22, 03.

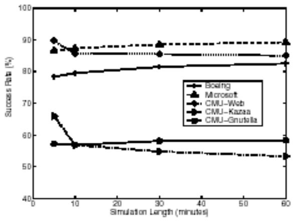
Con't

- Monitoring software, based on "tcpdump", installs a kernel filter to match packets containing an HTTP request or response header, regardless of port numbers.
- For web query workload, only static contents are used in the evaluation. Content containing "cgi", "asp", "pl" and "?" are removed from the log file.

Con't

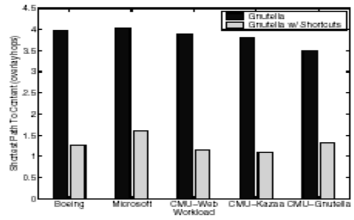
- **Simulation experiment**: the performance of Gnutella and that of Gnutella with shortcuts for each query workload are compared.
  - Eight one-hour segments from each query workload selected randomly;
  - Assume that peers that send any queries join the system at the beginning of the segment and stay until the end.

Results (success rate of shortcuts)



(a) Success rates of shortcuts.

Results (shortest path to content)



(c) Shortest path to content.

TABLE II

LOAD AT EACH PEER IN QUERY PACKETS/SECOND.

Trace	Protocol	5	6	7	8
Boeing	Gnutella Flooding	355.4	462.6	493.5	670.9
	Gnutella w/ Shortcuts	66.0	86.5	98.7	132.0
Microsoft	Gnutella Flooding	478.7	832.1	1,163.8	1,650.1
	Gnutella w/ Shortcuts	70.5	115.5	162.1	230.4

Potential and limitations of shortcuts

Questions:

- what is the best possible performance when peers learn about shortcuts through past queries;
- Are there practical changes to basic algorithm to bring the performance to the best performance?
- Can we improve shortcut performance if we discover shortcuts through our existing shortcuts, in addition to learning from past queries?

Con't

Three changes to the basic algorithm:

- peers add all peers returned from Gnutella's flooding as shortcuts;
- remove the 10-entry limit on the shortcut list size and allow the list grow without bound.

Con't

- discover shortcuts through existing shortcuts.

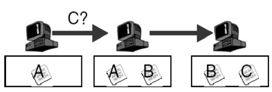
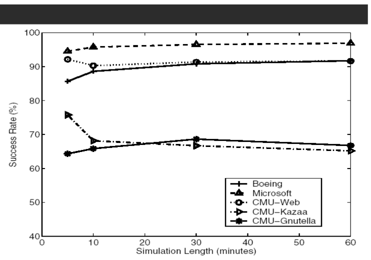


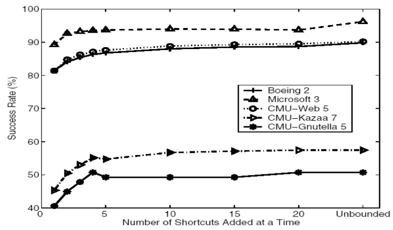
Fig. 5. Discovering new shortcuts through existing shortcuts.

Results
(add as many shortcuts as possible)



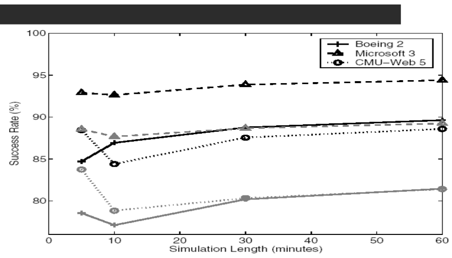
(a) Add as many shortcuts as possible.

Results
(success rate vs. # of shortcuts added)



(b) Success rate and the number of shortcuts added.

Results
(success rate of asking shortcut's shortcuts)



(c) Success rate for asking shortcuts' shortcuts.

Conclusions

- Propose a technique to create shortcuts in content location overlay;
- It is promising approach to introducing performance enhancements to overlay construction algorithms;
- Shortcuts significantly improve performance without degrading the scalability or correctness of the underlying overlay construction algorithms because they are designed as modular building blocks on top of generic large-scale overlay.

Critical on the paper

- The logic of the paper is good. The conclusions can be inferred from the experimental result very well.
- However, the experimental set-up is not extremely clear to me. How the authors extracted the results from original data is unknown to me.
- What's more, the authors did not interpret how to define the interest in detail! Just said to compare the file name/URL.

References:

[1] Rodrigo Rodrigues, Barbara Liskov, Liuba Shrira, "The Design of a Robust Peer-to-Peer System", Tenth ACM SIGOPS European Workshop. Saint Emilion, France, September 2002.

[2] Kunwadee Sripanidkulchai, Bruce Maggs, Hui zhang, "Efficient Content Location Using Interest-based Locality in Peer-to-Peer Systems".

Quiz questions

1. What is peer-to-peer system? What is the advantages of peer-to-peer system over traditional distributed system?
2. Explain flood content look-up algorithm.
3. Explain interest-based content look-up algorithm.
4. Please name several performance indices for content-location algorithm.
5. How to further improve the performance of interest-based content location algorithm?