Efficient Content Location Using Interest-Based Locality in Peer-to-Peer Systems

IEEE INFOCOM 2003

謝志峰

2003/3/20

Outline

- Introduction
- Interest-Based Locality
- Performance Evaluation
- Experimental Results
- Potential and limitations of shortcuts
- Conclusions
- Related Work

Introduction

- There are two classes of solutions currently proposed for decentralized peer-to-peer content location:

  - **Gnutella** : It is relies on **flooding** queries to all peers.Peers organize into an overlay. To find content, a peer sends a query to its neighbors on the overlay.The neighbors forward the query on to all of their neighbors until the query has traveled a certain radius.

  - **Distributed Hash Table(DHT)**:Peers organize into a well-defined structure that is used for routing queries.

Interest-Based Locality(1/5)

- They propose a technique called **shortcuts** to create additional links on top of a peer-to-peer system's overlay, taking advantage of locality to improve performance.
- Shortcuts are a powerful primitive that can be used to improve overlay performance.
- Shortcuts based on network latency can **reduce hop-by-hop delays** in overlay networks.

Interest-Based Locality (2/5)

- Figure 1 gives an example to illustrate **interest-based locality**.

- The peer in the middle is looking for files **A**,**B**,and **C**. The two peer in the right who have file A also each have at least one more matching file B or C .The peer on the upper right-hand corner has all three files. It and the peer in the middle share the most interest, where interests represent a group of files,namely {A,B,C}.Our goal is to identify such peers,and use them for downloading files directly.

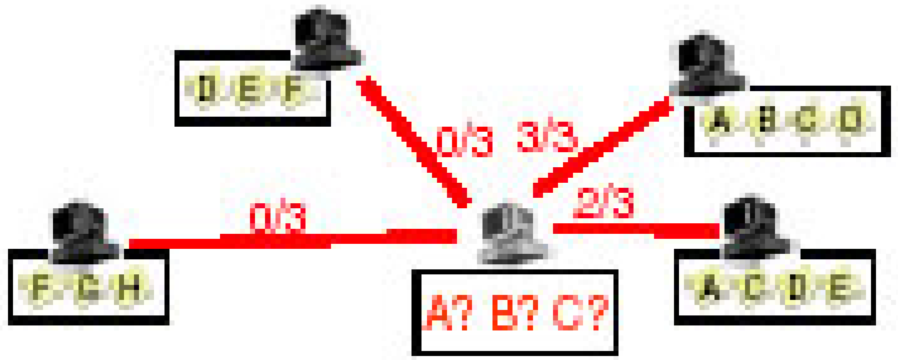
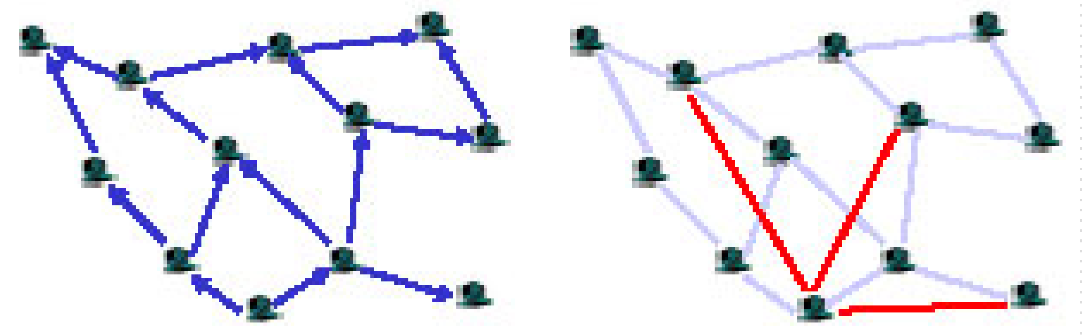


Fig. 1. Peers that share interests.

Interest-Based Locality (3/5)

- Figure 2(a) illustrates how content is located in Gnutella. A query initiated by the peer at the bottom is flooded to all peers in the system.

- Figure 2(b) depicts a Gnutella overlay with 3 shortcut links for the bottom-most peer. To avoid flooding, content is located first through shortcuts.



(a) Gnutella. (b) Shortcuts.

Fig. 2. Content location paths.

Interest-Based Locality (4/5)

- **Shortcut Discovery**
- They use the following heuristic to detect shared interests:
  - Peers that have content that we are looking for share similar interests.
  - When a peer joins the system, it may not have any information about other peers' interests.
  - Its first attempt to locate content is executed through flooding.
  - The lookup returns a set of peers that store the content.
  - There peers are potential candidates to be added to a "shortcut list".
  - If a peer cannot find content through the list, it issues a lookup through Gnutella, and repeats the process for adding new shortcuts.

Interest-Based Locality (5/5)

- Shortcut Selection:
  - They rank shortcuts based on their perceived utility.If shortcuts are useful,they are ranked at the top of the list.A peer locates content by sequentially asking all of the shortcuts on its list, starting from the top, until content is found.
  - Rankings can be based on many metrics, such as **probability of providing content**, **latency** of the path to the shortcut, available **bandwidth** of the path , amount of content at the shortcut, and load at the shortcut.
  - Each peer continuously keeps track of each shortcut's performance and updates its ranking when new information is learned.

Performance Evaluation (1/2)

- The metrics they use to express the benefits and overhead of interest-based shortcuts are:
  - **Success rate**:How often are queries resolved through shortcuts?If success rates are high, then interest-based locality techniques have the potential to improve performance.
  - **Load characteristics**:How many query packets do peers process while participating in the system?Reducing the load at individual peers is desirable for scalability.
  - **Query scope**:For each query, what fraction of peers in the system are involved in query processing?A smaller query scope increases system scalability.
  - **Minimum reply path lengths**:How long does it take for the first reply to come back?
  - **Additional state**:How much additional state do peers need to maintain in order to implement shortcuts? The amount of state measures the cost of shortcuts and should be kept to a minimum.

Performance Evaluation (2/2)

- They use five diverse traces of downloads request from real content distribution application to generate query workloads.
- First three traces(labeled Boeing, Microsoft and CMU-Web in) capture Web request workloads,which we envision to be similar to requests in Web content file-sharing applications.
- Last two traces (labeled CMU-Kazaa and CMU-Gnutella) capture requests from two popular file-sharing applications.

TABLE I

TRACE CHARACTERISTICS.

Trace	Characteristics	1	2	3	4	5	6	7	8
Boeing	Requests	95,504	95,429	166,741	201,862	1,176,153	1,541,062	1,617,608	2,039,347
	Documents	42,800	44,153	75,833	79,306	305,092	391,229	434,766	513,264
	Clients	868	1,052	1,443	2,278	18,059	21,690	22,344	25,293
Microsoft	Requests	764,177	917,325	960,119	1,588,045	2,083,911	3,818,368	4,515,815	6,671,774
	Documents	102,548	164,505	198,559	285,711	416,784	662,986	718,444	956,617
	Clients	11,636	11,929	13,013	15,387	19,419	23,492	28,741	32,361
CMU-Web	Requests	125,138	104,781	132,405	155,847	338,656	358,778	432,843	495,119
	Documents	61,569	43,616	61,981	72,513	162,951	153,405	190,372	211,570
	Clients	6,322	6,426	7,054	7,602	11,176	12,274	13,892	15,408
CMU-Kazaa	Distinct Requests	7,757	7,779	8,086	9,075	9,243	13,307	13,760	15,188
	Documents	3,720	3,625	3,806	4,338	4,771	6,619	7,172	6,312
	Clients/Peers	6,482/6,985	6,514/6,968	6,732/7,217	7,468/8,064	7,601/8,542	10,977/11,983	11,362/12,660	12,558/13,590
CMU-Gnutella	Distinct Requests	392	389	395	415	480	502	581	884
	Documents	260	247	239	254	318	339	393	609
	Clients/Peers	256/464	270/383	271/373	296/405	320/543	341/477	383/590	542/735

Experimental Results (1/3)

- The average success rate at the end of 1 hour is as high as 82%-90% for the Web workloads,and 53%-58% for the CMU-Gnutella and CMU-Kazaa workloads.
- Success Rate:Success rate is defined as the number of lookups that were successfully resolved through interest-based shortcuts over the total number of lookups.
- The individual success rate observed at each peer increases with longer simulation times as peers learn more about other peers and have more time to refine their shortcut list.

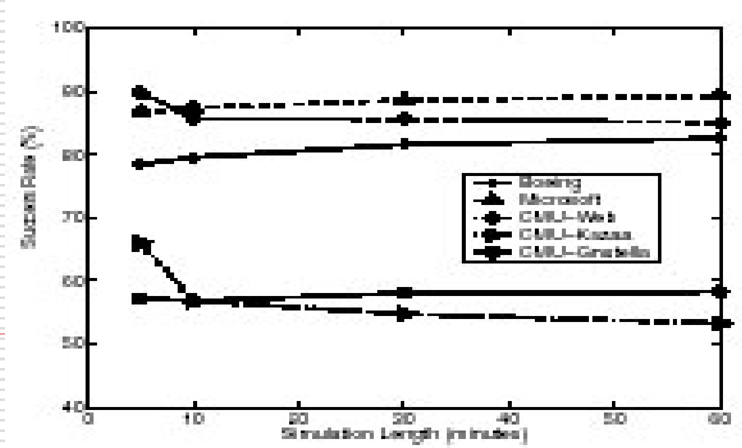


Figure 3(a)

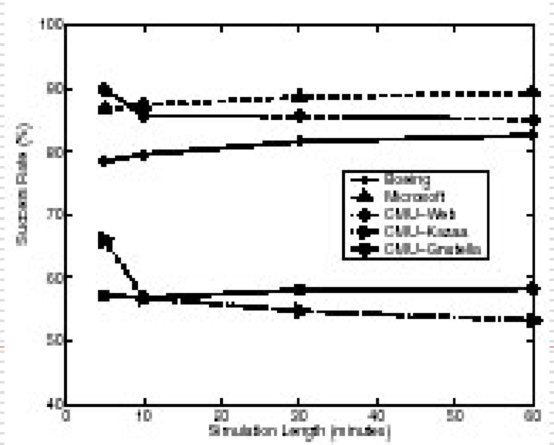
Experimental Results (3/3)

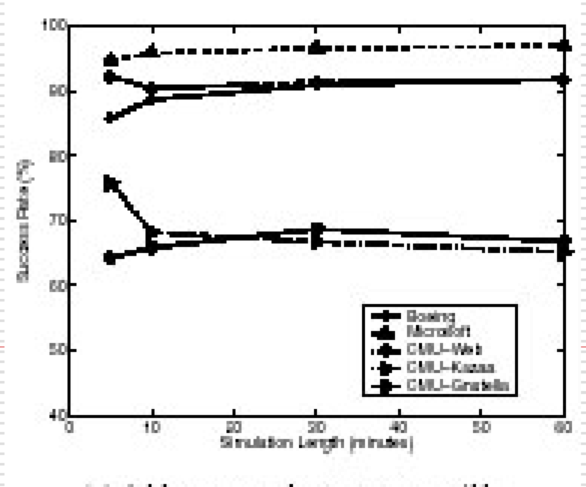
- Load at each peer and query scope.
- **Less load** and **smaller scope** can help improve the scalability of Gnutella.
- Load is measured as the number of query packets seen at each peer.
- Shortcuts are effective at finding both **popular** and **unpopular** content.When using shortcuts**,45%-90%** of content can be found quickly and efficiently.
- Shortcuts have good load distribution properties. The overall load is reduced, and more load is redistributed towards peers that make heavy use of the system. In addition, shortcuts help to limit the scope of queries.
- Shortcuts are scalable, and incur very little overhead.

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 (1/2)

- Figure 4(a) depicts the best possible success rate averaged across all trace segments for all workload.
- The average success rate at the end of 1 hour is as high as 97% and 65% for the Microsoft and CMC-Kazaa workload.
- We observe that success rate for the basic shortcuts algorithm depicted in Figure 3(a) is only **7-12%** less than the best possible.





Potential and limitations of shortcuts (2/2)

- In Figure 5 the peer on the left has previously downloaded file A, and has added the peer in the middle as a shortcut.
- It now wants to find file C.
- Its immediate shortcut does not have file C, but its shortcut's shortcut which is the peer on the right , has file C.
- In this case, the peer on the left can successfully locate content through its shortcut's shortcut.

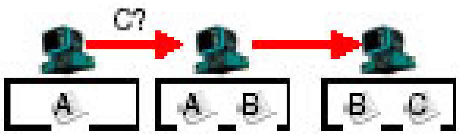


Fig. 5. Discovering new shortcuts through existing shortcuts.

Understanding Interest-Based Locality (1/2)

- They explore the effect of the structure of **Web pages** on interest-based locality.
- Each Web page consists of multiple embedded objects.
- If objects in peer-to-peer systems have the same granularity as Web pages, would interest-based locality still be useful?

Understanding Interest-Based Locality (2/2)

- The black lines in Figure 6 depict the average success rate for the Web page workloads when using the basic algorithm of adding one shortcut at a time.
- The success rates are **85%** and **63%** at the end of the one-hour period for the Microsoft and the CMU-Web workloads.
- The performance are contributed to by structure of Web pages and the interest-based relationship between Web pages.
- Intersrt-based shortcuts are capable of exploiting both properties to further improve performance.

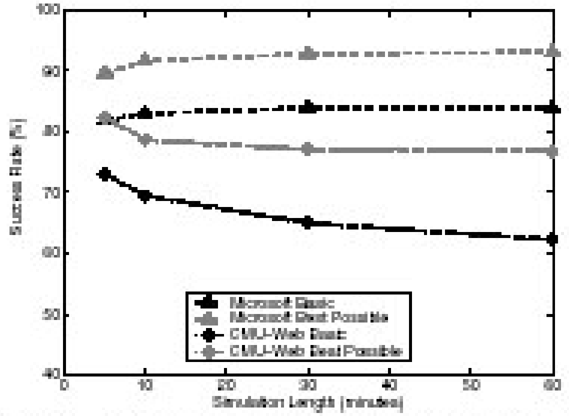


Fig. 6. Success rates for Web page workload.

Sensitivity To Underlying Content Location Mechanism.

- They explore the performance of shortcuts with Chord,a DHT-based protocol.
- Chord provides efficient and scalable distributed lookups that resolves content Ids to locations in 1/2logN overlay hops, where N is the number of participating peers. To facilitate lookups,each node maintains O(logN) state about peers in the system.
- Interest-based shortcuts can improve the performance of Chord, a DHT-based protocol.

LOAD AT EACH PEER IN QUERY PACKETS/SECOND.

Trace	Protocol	5	6	7	8
Boeing	Chord	0.0352	0.0414	0.0473	0.0397
	Chord w/ Shortcuts	0.0113	0.0132	0.0165	0.0145
Microsoft	Chord	0.0677	0.0985	0.1179	0.1462
	Chord w/ Shortcuts	0.0228	0.0334	0.0365	0.0471

Conclusions(1/2)

- Performance-based content retrieval can also be implemented using interest-based shortcuts. There advantage of such a service is that content can be retrieved from the peer with the best performance.

- Using interest-based shortcuts in Gnutella and Distributed Hash Table(DHT) can improve performance

Conclusions(2/2)

- In addition to improving content location performance,keyword or string matching searches for content and performance-based content retrieval are two examples of such services.
- Interest-based short-cuts only allow such information to be used intelligently to improve performance.

Related Work

- Approaches based on expanding ring searches,which are designed to limit the scope of queries.

- Approaches based on query caching searches,which improve performance.