Research Problems for Scalable Internet Resource Discovery

C. Mic Bowman
Peter B. Danzig
Michael F. Schwartz

CU-CS-643-93 March 1993

University of Colorado at Boulder

Technical Report CU-CS-643-93
Department of Computer Science
University of Colorado
Boulder, Colorado 80309-430

Research Problems for Scalable Internet Resource Discovery

C. Mic Bowman
Computer Science Department
Pennsylvania State University
University Park, Pennsylvania 16802-6103
+1 814 865 9505
bowman@cs.psu.edu

Peter B. Danzig
Computer Science Department
University of Southern California
941 W. 37th Place
Los Angeles, California 90089-0781
+1 213 740 4780
danzig@usc.edu

Michael F. Schwartz
Computer Science Department
University of Colorado
Boulder, Colorado 80309-0430
+1 303 492 3902
schwartz@cs.colorado.edu

March 1993

Abstract

Over the past several years, a number of information discovery and access tools have been introduced in the Internet, including Archie, Gopher, Netfind, and WAIS. These tools have become quite popular, and are helping to redefine how people think about wide area network applications. Yet, they are not well suited to supporting the future information infrastructure, which will be characterized by enormous data volume, rapid growth in the user base, and burgeoning data diversity. In this paper we indicate trends in these three dimensions, and survey problems these trends will create for current approaches. We then suggest several promising directions of future resource discovery research, along with some initial results from projects carried out by members of the Internet Research Task Force Research Group on Resource Discovery and Directory Service.

1 Introduction

In its roots during the ARPANET days, the Internet was conceived primarily as a means of remote login and experimentation with data communication protocols. However, the predominate usage mode quickly became electronic mail, in support of collaboration. This trend has continued into the present incarnation of the Internet, but with increasingly diverse support for collaborative data sharing activities. Electronic mail has been supplemented by a variety of wide area file systems, remote information retrieval systems, and publishing and library access systems. At present, the Internet provides access to hundreds of gigabytes each of software, documents, sounds, images, and other file system data; library catalog and user directory data; weather, geography, telemetry, and other physical science data; and many other types of information.

To make effective use of this wealth of information, users need ways to locate information of interest. In the past few years, a number of such *resource discovery* tools have been created, and have gained wide popular acceptance in the Internet [ED92, KM91, McC92, BLCGP92, ST91, Neu92, Dro90].[1] Our goal in the current paper is to examine the impact of scale on resource discovery tools, and place these problems into a coherent framework. We focus on three scalability dimensions: the burgeoning diversity of information systems, the growing user base, and the increasing volume of data available to users.

Table 1 summarizes these dimensions, suggests a set of corresponding conceptual layers, and indicates problems being explored by the Internet Research Task Force (IRTF) Research Group on Resource Discovery and Directory Service. Users perceive the available information at the *information interface* layer. This layer must support scalable means of organizing, browsing, and searching. The *information dispersion* layer is responsible for replicating, distributing, and caching information. This layer must support access to information by a large, widely distributed user populace. The *information gathering* layer is responsible for collecting and correlating the information from many incomplete, inconsistent, and heterogeneous repositories.

The remainder of this paper covers these layers from the bottom up. Section 2 discusses problems of information system diversity. Section 3 discusses the problems brought about by growth in the user base. Section 4 discusses problems caused by increasing information volume. Finally, in Section 5 we offer a summary.

2 Information System Diversity

An important goal for resource discovery systems is providing a consistent, organized view of information. Since information about a resource exists in many repositories—within the object itself and within other resource discovery systems—resource discovery systems need to identify a resource, collect information about it from several sources, and convert the representation to a format that can be indexed for efficient searching.

As an example, consider the problem of constructing a user directory. In a typical environment, several existing systems contain information about users. The Sun NIS database [Wei85] usually has information about a user's account name, common name, address, group memberships, password, and home directory. The `ruserd` server has information about a user's workstation and its idle time. In addition, users often place information in a ".plan" file that might list the user's travel schedule, home address, office hours, and research interests.

As this example illustrates, information in existing Internet repositories has the following characteristics:

- It is heterogeneous.

  Each repository maintains the information it needs about resources. For example, the primary purpose of the NIS database is to maintain information about user accounts, while a user's ".plan" file often contains more personal information. In addition, the two repositories represent the information differently: records in an NIS database have a fixed format, but a ".plan" file contains unstructured text.

- It is inconsistent.

  Most information contained in Internet repositories is dynamic. Certain properties change frequently, such as which workstations a user person is using. Other properties change more slowly, such as a user's mail address. Because information is maintained by several repositories that perform updates at different times using different algorithms, there will often be conflicts between information in the various repositories. For example,

---

[1] The reader interested in an overview of resource discovery systems and their approaches is referred to [SEKN92].

Scalability Dimension	Conceptual Layer	Problems for RD Systems	IRTF Research Focus
Data Volume	Information Interface	Information Overload	Scalable Content-Based Searching
User Base	Information Dispersion	Insufficient Replication; Manual Distribution Topology	Massive Replication; Data Access Measurements
Data Diversity	Information Gathering	Difficulty of Extracting Data; Low Data Quality	Operation Mapping; Data Mapping

Table 1: Dimensions of Scalability and Associated Research Problems

information about account name, address, and phone number may be maintained by both the NIS database and an X.500 [Int88] server. When a user's address or phone number changes, the X.500 service will probably be updated first. However, if the account changes, the NIS database will usually be the first to reflect the change.

- It is incomplete.

  Additional attributes of a resource can often be obtained by combining information from several repositories. For example, a bibliographic database does not contain explicit information about a person's research interests. However, keywords might be extracted from the person's research papers, to try to infer research interests for a user directory.

There are two common approaches to these information gathering layer problems. The first approach—*operation mapping*—constructs a "gateway" between existing systems, which maps the functionality of one system into another without actually copying the data. The second approach—*data mapping*—generates an aggregate repository from multiple information sources. The remainder of this section discusses these approaches.

2.1 Operation Mapping: Gateways

A gateway between two resource discovery systems translates operations from one system into operations in another system. Ideally, the systems interoperate seamlessly, without the need for users to learn the details of each system. Sometimes, however, users must learn how to use each system separately.

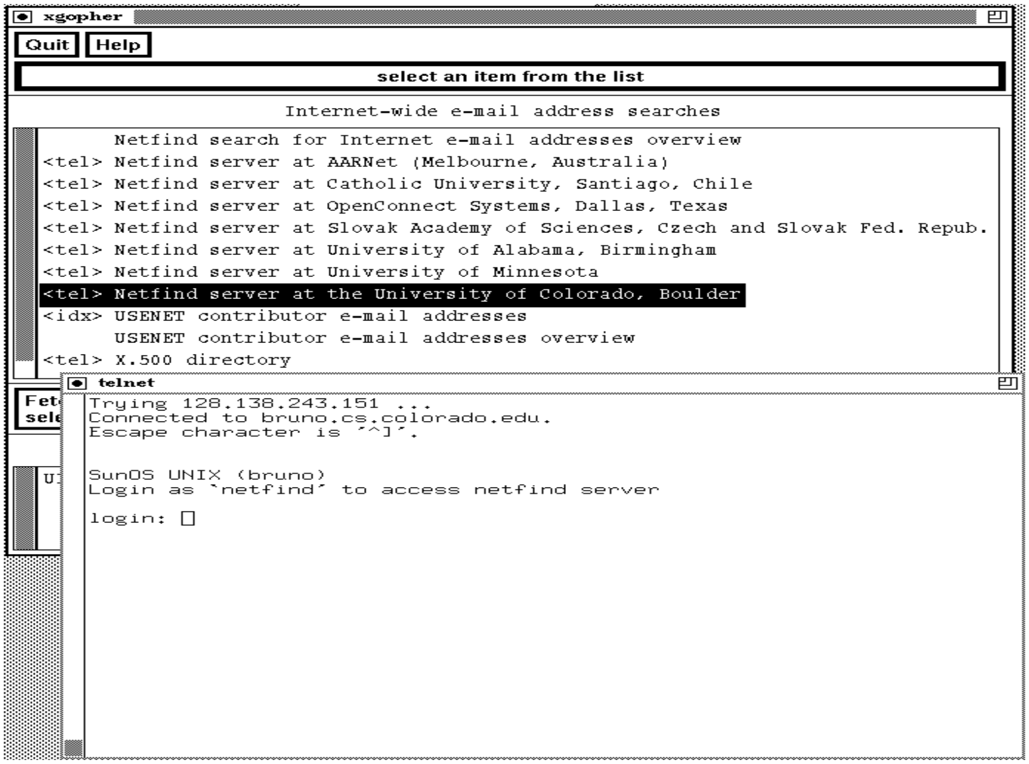
Building seamless gateways can be hindered if one system lacks operations needed by another system's user interface [SEKN92]. For example, if would be difficult to provide a seamless gateway from a system (like WAIS [KM91]) that provides a search interface to users, to a system (like Prospero [Neu92]) that only support browsing. Even if two systems support similar operations, building seamless gateways may be hindered by another problem: providing appropriate mappings between operations in the two systems. To illustrate the problem, consider the current interim gateway from Gopher [McC92] to Netfind [ST91], illustrated in Figure 1.[2] Because the gateway simply opens a telnet window to a UNIX program that provides the Netfind service, users perceive the boundaries between the two systems.

In contrast, we have built a system called Dynamic WAIS [HS93a], which extends the WAIS paradigm to support information from remote search systems (as opposed to the usual collection of static documents). The prototype supports gateways from WAIS to Archie and to Netfind, using the Z39.50 information retrieval protocol [ANS91] to seamlessly integrate the information spaces. The Dynamic WAIS interface to Netfind is shown in Figure 2.

The key behind the Dynamic WAIS gateways is the conceptual work of constructing the mappings between the WAIS search-and-retrieve operations, and the underlying Archie and Netfind operations. In the case of Netfind, for example, when the Dynamic WAIS user requests a search using the "dynamic-netfind.src" WAIS source, the search is translated to a lookup in the Netfind seed database, to determine potential domains to search. The Netfind domain selection request is then mapped into a WAIS database selection request (the highlighted selection in the XWAIS Question window). Once the user selects one of the domains to search, the WAIS retrieval phase is mapped into an actual domain search in Netfind (the uppermost window).

---

[2] Efforts are under way to improve this gateway.



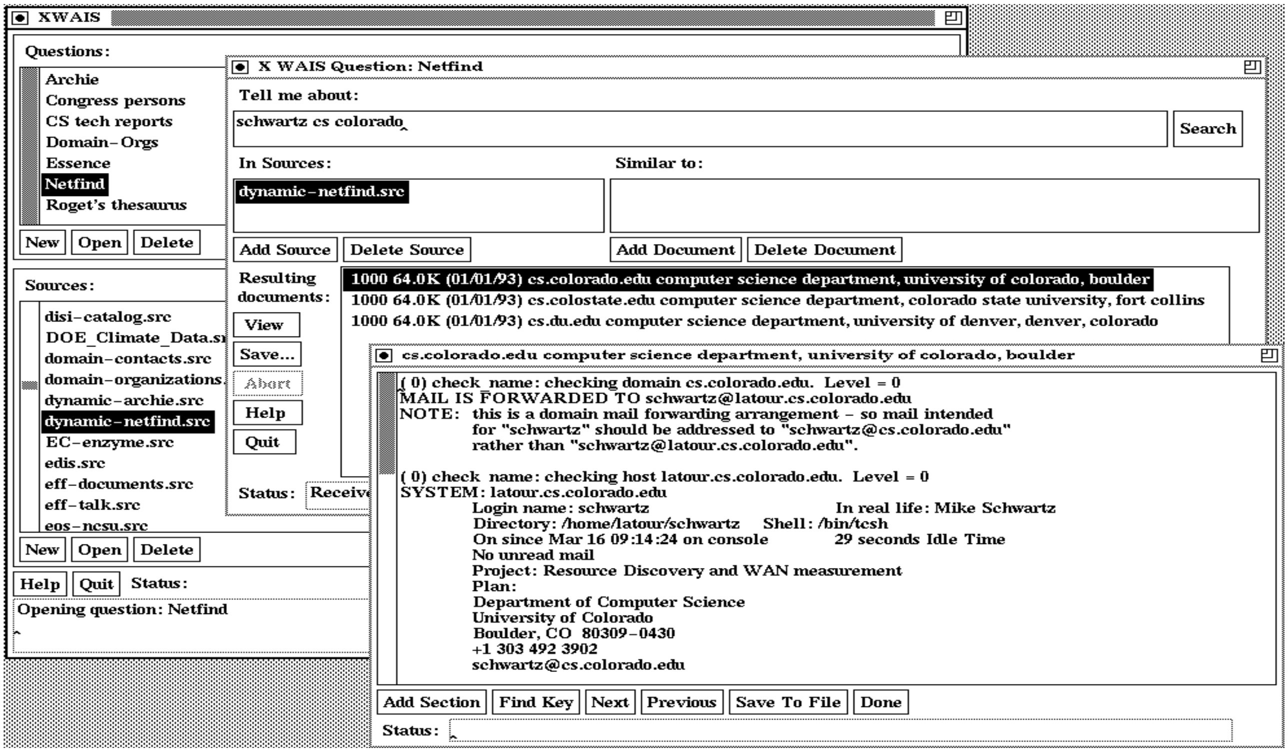
```
Internet-wide e-mail address searches
      Netfind search for Internet e-mail addresses overview
<tel> Netfind server at AARNet (Melbourne, Australia)
<tel> Netfind server at Catholic University, Santiago, Chile
<tel> Netfind server at OpenConnect Systems, Dallas, Texas
<tel> Netfind server at Slovak Academy of Sciences, Czech and Slovak Fed. Repub.
<tel> Netfind server at University of Alabama, Birmingham
<tel> Netfind server at University of Minnesota
<tel> Netfind server at the University of Colorado, Boulder
<idx> USENET contributor e-mail addresses
      USENET contributor e-mail addresses overview
<tel> X.500 directory
```

```
Trying 128.138.243.151 ...
Connected to bruno.cs.colorado.edu.
Escape character is '^]'.


SunOS UNIX (bruno)
Login as `netfind' to access netfind server

login: 
```

Figure 1: Gopher Menu-Level Gateway to Netfind



Status:
Opening question: Netfind

Tell me about: schwartz cs colorado

In Sources: dynamic-netfind.src

Similar to:

Resulting documents:

```
1000 64.0K (01/01/93) cs.colorado.edu computer science department, university of colorado, boulder
1000 64.0K (01/01/93) cs.colostate.edu computer science department, colorado state university, fort collins
1000 64.0K (01/01/93) cs.du.edu computer science department, university of denver, denver, colorado
```

Status: Receiv

cs.colorado.edu computer science department, university of colorado, boulder

```
( 0) check_name: checking domain cs.colorado.edu.  Level = 0
MAIL IS FORWARDED TO schwartz@latour.cs.colorado.edu
NOTE:  this is a domain mail forwarding arrangement - so mail intended
       for "schwartz" should be addressed to "schwartz@cs.colorado.edu"
       rather than "schwartz@latour.cs.colorado.edu".

( 0) check_name: checking host latour.cs.colorado.edu.  Level = 0
SYSTEM: latour.cs.colorado.edu
       Login name: schwartz                    In real life: Mike Schwartz
       Directory: /home/latour/schwartz    Shell: /bin/tcsh
       On since Mar 16 09:14:24 on console          29 seconds Idle Time
       No unread mail
       Project: Resource Discovery and WAN measurement
       Plan:
       Department of Computer Science
       University of Colorado
       Boulder, CO  80309-0430
       +1 303 492 3902
       schwartz@cs.colorado.edu
```

Status:

Figure 2: Dynamic WAIS Information-Level Gateway to Netfind

2.2 Data Mapping: Collecting Data and Resolving Conflicts

Rather than invoking a resource discovery system through a gateway, an alternate means of accommodating diversity is to collect the data directly from the underlying repositories. Doing so involves two parts: mapping protocols for collecting information, and agreement protocols for correlating information [BPY90].

A mapping protocol is implemented as a function that collects information from a repository and reformats it. There may be several implementations of mapping protocols, each customized for an existing repository. The most common mapping protocols are implemented as clients of an existing service. For example, Netfind extracts user information from several common Internet services, including the Domain Naming System (DNS) [Moc87], the finger service [Zim90] and the Simple Mail Transfer Protocol [Pos82].

The agreement protocol defines a method for handling conflicts between different repositories. For example, Figure 3 illustrates data for the Enterprise [BD93] user directory system, which is built on top of Univers. This figure shows three mapping protocols that gather information from the NIS database, the ruserd server, and the user's electronic mail, respectively. Several attributes can be generated by more than one mapping protocol. For example, address information potentially exists in both the NIS database and the information supplied by the user. The agreement protocol considers information gathered directly from the user as the most reliable. Depending on the attribute, the agreement protocol may permit some properties to have several values, such as the two address attributes that describe the user in Figure 3.

Data mining is a special case of agreement protocols, which works by cross-correlating information available from multiple repositories. This can have two beneficial effects. First, data mining can flag inconsistencies that indicate potential problems. For example, the Enterprise agreement function could mail a message to the user if it detected a conflict between the electronic mail addresses listed for the user in different repositories.

Second, data mining can support mapping protocols where information not explicitly present in any of the repositories can be derived by cross-correlating existing information. For example, Netfind continuously collects data from a number of sources, and cross-correlates the information to form a more complete seed database. One source might discover a new host called "astro.columbia.edu" from recursive Domain Naming System (DNS) traversals [Lot92], and cross-correlate this information with the existing seed database record for columbia.edu ("columbia university, new york, new york"), its understanding of the nesting relationships of the Domain name space, and a database of departmental abbreviations, to derive a new record for the Astronomy Department at Columbia University.

3 User Base Scale

Already interconnecting millions of users in thousands of sites, the Internet will grow significantly with the addition of new constituencies, including commercial, K-12 schools, and digital library users. This growth will place a great deal more load on Internet links and servers.

To support a much larger user populace, we believe two problems must be addressed. First, discovery services must support significantly higher, even massive levels of caching and replication, using algorithms specifically designed to function in the Internet's dynamic patchwork of autonomous systems. Second, we believe that continual study and instrumentation of access patterns and data movement will be necessary to tune the placement of caches and replicas. We consider these issues below.

3.1 Massive Caching and Replication

Name servers scale well because their data is typically partitioned hierarchically. Because resource discovery tools search flat, rather than hierarchical or otherwise partitionable views of data, the only way to make these tools scale is by replicating them. To gain an appreciation for the degree of replication required, consider the Archie file location service.

The global collection of Archie servers process approximately 50,000 queries per day, generated by a few thousand users worldwide. Every month or two of Internet growth requires yet another replica of Archie. A dozen Archie servers now replicate a continuously evolving 150 MB database of 2.1 million records. While it responds in seconds on a Saturday night, it can take five minutes to several hours to answer simple queries during a weekday afternoon. Even with no new Internet growth, for

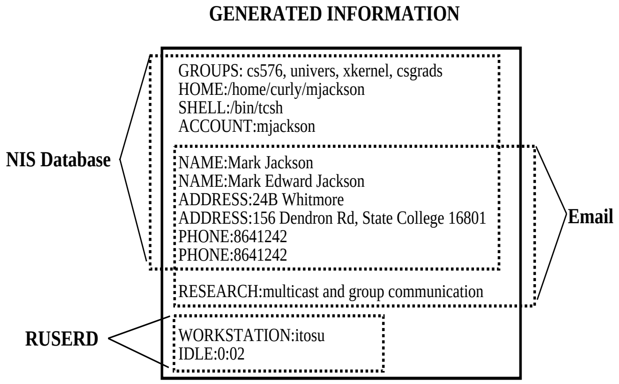


Figure 3: Example Mapping Protocols for User Directory Information

Archie to yield five second response times during peak hours we would need at least 60 times more Archie servers than the current dozen. Because of its success, Archie needs to be replicated thousands of times, just like network news is [Qua90]. Other successful tools that cannot easily partition their data will also require massive replication.

Because replication levels greater than a dozen servers have not been demonstrated on the Internet, we believe this problem requires additional research. On the one hand, without doubt we know how to replicate and cache data that partitions easily, as in the case of name servers [Moc87, BLNS82]. Primary copy replication works well because name servers do not perform associative access, and organizational boundaries limit the size of a domain, allowing a handful of servers to meet performance demands.[3] We have learned many lessons to arrive at this understanding [DOK92, MD88, SBN84]. On the other hand, we have little experience deploying replication and caching algorithms to support massively replicated, flat, yet autonomously managed databases.

What do existing replication schemes for wide-area services lack? First, existing distributed replication systems ignore network topology; they do not route their updates in a manner that makes efficient use of the Internet. We believe it is necessary to calculate the topology over which updates traverse, and to manage replication groups that exploit the Internet's partitioning into autonomous domains. Existing schemes do not guarantee timely and efficient updates in the face of frequent changes in physical topology, network partition, and temporary or permanent node failure. In essence, they treat all physical links as having equal bandwidth, delay, and reliability, and do not recognize administrative domains.

We believe that flooding-based replication algorithms can be extended to work well in Internet environment. In fact, both network news and the dozen or so Archie servers replicate with flooding algorithms. However, for lack of good tools, administrators of both Archie and netnews manually configure the flooding topology over which updates travel, and manually reconfigure this topology when the physical network changes. This is not an easy task because Internet topology changes quickly, and hand-composed maps are never current. While we are developing tools to map the Internet [WCS93], even full network maps will not automate update topology calculation.

Avoiding topology knowledge by using multicast protocols [DC90, AFM92] for data distribution has different problems. First, they are limited to single routing domains. Second, Internet multicast attempts to minimize message delay, which is the wrong metric for bulk transport. At the very least, we see the need for different routing metrics. Even a reliable, bulk transport multicast is probably not the right answer, because replication algorithms require mechanisms to remain highly available in the presence of permanent site failures and changes in the replication group.

We are currently exploring an approach to providing massively replicated, loosely consistent services [DLO92, ODDA93]. This approach extends ideas presented in Lampson's Global name service [Lam86] to operate in the Internet. Briefly, our approach organizes the replicas of a service into groups, imitating the idea behind the Internet's autonomous routing domains. Group replicas estimate the physical topology, and then create an update topology between the group members. The left hand side of Figure 4 shows three replication domains. Inside each replication domain, a logical update topology is established that best uses the physical topology connecting the replicas. Whenever the algorithm detects a meaningful change in physical topology between members of a replication domain, it modifies the logical update topology accordingly. Because it does not require a separate recovery algorithm, it is simpler than solutions based on Internet multicast.

Our studies of data movement in the Internet (Section 3.2) lead us to believe that services should help manage their own replication. We believe that services should monitor their request streams and cooperate to make configuration

---

[3] Because it is both a name service and a discovery tool, X.500 could benefit from massive replication.

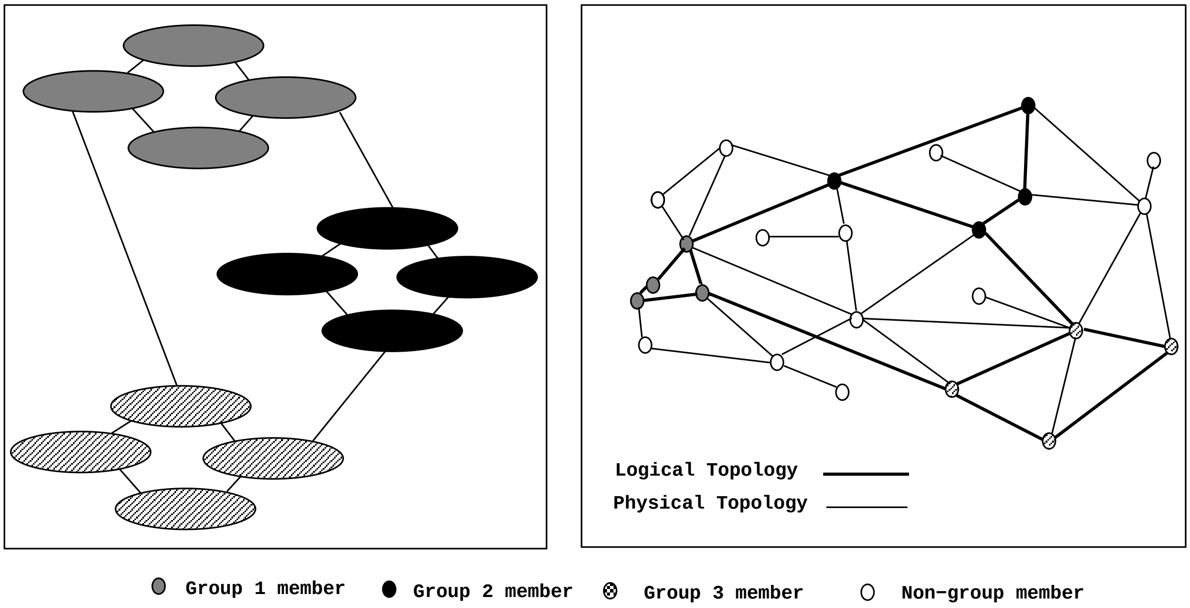


Figure 4: Replication domains and physical versus logical update topology

management suggestions. Suppose, for example, that some X.500 server in Europe fields a large stream of queries from clients in the United States. Should it not suggest that a strategically located replica be created? Alternatively, if some replica finds that it receives few or no queries, should it not suggest its own destruction?

3.2 Instrumentation and Study

Choosing the appropriate degree and placement of replicas requires measuring how data is accessed. Below we briefly discuss two studies that indicate a set of problems with existing Internet mechanisms.

After tracing NSFNET FTP traffic, we recently showed that judiciously placed file caches could reduce the volume of backbone traffic by 17-27% [DHS93]. In addition to reducing network traffic, a hierarchical architecture of whole file caches could reduce disk usage and consistency problems caused by the uncoordinated style of replication currently in use in Internet FTP archives.

We also recently evaluated access patterns to the Archie database. We found, somewhat contrary to what we said about partitioning its database, that Archie data in fact partions into a hot set and a cold set. Only 16% of Archie's database is hot. We suspect that many services, like Archie, would benefit from automatically partitioning itself into hot and cold sets, and only massively replicating the hot set.

Without instrumentation, many services would waste network and secondary store resources on replicating rarely accessed data. Our studies of FTP and Archie underscore the importance of measuring and exploiting access patterns when making caching and replication choices in distributed information systems.

4 Data Volume

The amount of information currently available on the Internet may conservatively be estimated at one terabyte. As the user base increases by two to three orders of magnitude, the amount of available information can be expected to increase proportionately. Moreover, new usage modes will contribute additional scale. For example, as multimedia applications begin to proliferate, users will create and share highly voluminous audio, image, and video data, adding easily one to two more orders of magnitude of data.

In some cases users already have voluminous data, but cannot use the Internet to share the data because of link bandwidth limits. For example, earth and space scientists collect sensor data at rates as high as gigabytes per day [FJP90, NAS90]. To share data with colleagues, they are forced to send magnetic tapes through the mail. As Internet link capacities increase, scientists will naturally begin using the Internet to share their data, adding several more orders of magnitude to the global collection.

Putting all of these trends together, we can expect today's one terabyte of data to grow by a factor of 10^5 to 10^{10} over the next few years. Clearly, at this scale the amount of data will far outstrip the abilities of current systems to support fruitful information discovery. Imagine attempting to browse through a Gopher menu system with one million times as many entries as are currently present, or using Archie to locate a file from among one million times as many files as are presently tracked! Users would be completely overwhelmed by irrelevant data.

4.1 Searching as a Primary Information Interface

It is not obvious how one might build a system that could support discovery in such a large environment. However, we can observe some trends and characteristics of current systems, which we believe indicate fruitful avenues of exploration for future discovery systems.

There are two resource discovery paradigms in common use in the Internet: organizing/browsing, and searching. *Organizing* refers to the human-guided process of deciding how to interrelate information, usually by placing it into some sort of directed graph (e.g., the hierarchy of directories in an FTP file system). *Browsing* refers to the corresponding human-guided activity of exploring the organization and contents of a resource space. *Searching* is an automated process, where the user provides some description of the resources being sought, and a discovery system locates some matches.

Searching becomes the most feasible approach in large information spaces, because the effectiveness of browsing depends heavily on how well organized the information is. As users generate and collect increasing amounts of data, keeping it all well organized becomes increasingly difficult. In fact, the notion of "well organized" is highly subjective and personal. What one user finds clear and easy to browse may be difficult for users who have different needs or backgrounds. To some extent this problem can be alleviated by systems that support multiple views of information [CM86, Neu92]. Yet, doing so really pushes the problem "up" a level—users must locate appropriate views, which in itself is another discovery problem. Moreover, because there are few barriers to "publishing" information on the Internet, there is a great deal of information that is useful to only very few users, and often for only a short period of time. To other users, this information clutters the "information highway", making browsing difficult.

Beyond the scalability problems of the organizing/ browsing paradigm, the importance of searching can be seen in its increasing popular appeal. A number of file system search tools have been introduced in recent years [HS93b, GJSJ91, MW93, Sun92], and recently a search facility [Fos92] was introduced for Gopher - which is, without doubt, the world's largest browse-based system. Moreover, it is interesting to note that while the Prospero model [Neu90] focuses on organizing and browsing, the Prospero system is primarily used in its capacity as an interface to the Archie search system.

The above discussion is not meant to imply that organizing information is unimportant. Clearly, a well organized information space is easier to use than a poorly organized one. Rather, the point is that once an information space gets large enough, searching is the most feasible option. Ideally, resource discovery systems can combine these approaches into a search-then-browse paradigm, in which users submit search queries and then use the results as starting points to begin browsing. This model is supported, for example, by the Alex file system's "archia" tool [Cat92], which performs an Archie search request and returns the results in the form of Alex path names. The user can then change directories through Alex and browse the relevant parts of the FTP information space.

4.2 Efficient Yet Representative Searching

Given its importance, what can be done to improve the effectiveness of searching? Consider the current range of approaches to indexing information. Figure 5 plots space efficiency against index representativeness. Representativeness is important as an information space becomes increasingly large, because the probability of false hits and false drops increases with the size of the information space [Sal86]. Space efficiency is important, because the amount of data that can fit in a single index determines the scope of efficiently searchable data. For example, the fact that Archie fits over 1,000 anonymous FTP sites' directories into a single index means that it is feasible to search that space. If the index were so large that it had to be placed on 100 different machines and each machine searched for every query, searches would

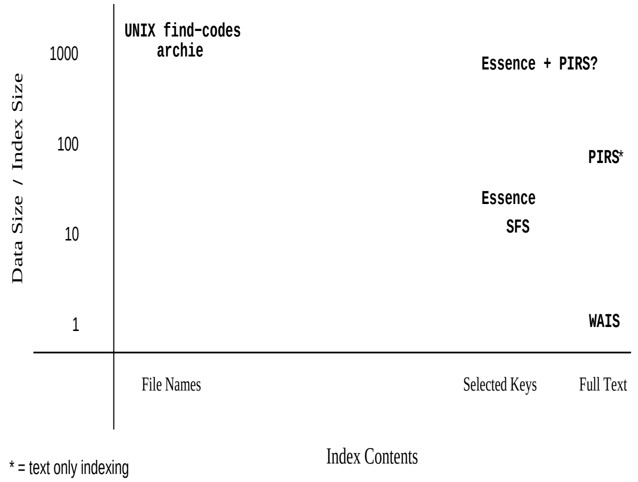


Figure 5: Indexing Space Efficiency vs. Representativeness

be significantly more expensive. The latter case is closer to how WAIS works, and is the reason why WAIS limits users to searching one database at a time. If it were possible to collapse 1,000 WAIS databases into a single index, users could do much more sophisticated searches through Archie data. Imagine, for example, if it were possible to search all of the anonymous FTP sites indexed by Archie, using WAIS full-text searches.

As illustrated by Figure 5, Internet indexing systems tend to fall into two extremes: space efficient but marginally representative (Archie and UNIX find-codes [Ber86]), and space-inefficient but highly representative (WAIS). The middle range of Figure 5 is occupied by three other systems. Essence [HS93b] and the MIT Semantic File System [GJSJ91] do selective indexing of documents, selecting keywords based on knowledge of the structure of the documents being indexed. For example, Essence understands the structure of several common word processing systems (as well as most other common file types in UNIX environments), and uses this understanding to extract authors, titles, and abstracts from text documents. In this way, it is able to select fewer keywords for the index, yet retain many of the keywords that users would likely use to locate documents.

Manber's Personal Information Retrieval System
(PIRS [MW93]) takes a different approach. Rather than selective indexing, PIRS supports full-text indexes, using a hybrid index/search scheme. A PIRS index is more rough-grained than a WAIS index, keeping pointers only to the granularity of blocks small enough to be searched sequentially. Manber has demonstrated this approach on 100 megabyte data sets with indexes of only 1-4% the size of the indexed data, and search times of a few seconds. This technique exploits the fact that a fairly small and fixed set of words account for the content of most natural language documents. Representing each keyword only once for several documents (and searching) rather than once for each occurrence in each document saves a significant amount of index space.

We are currently investigating means of supporting highly representative yet space-efficient indexes, through a hybrid of the PIRS and Essence approaches. Because each scheme achieves indexing space efficiency through different means (selective indexing in Essence, and hybrid index/search in PIRS), combining the schemes may potentially yield a multiplicative space reduction. In the best case, this could mean an index of .05% the size of the indexed data. With this space efficiency, it would be possible to support flat searches through 2,000 one-gigabyte disks, with a top-level index that fits on a single disk. The ability to coalesce the information into a single index is important, as it allows efficient global searches.

Achieving this level of indexing efficiency will require research to overcome two problems. First, PIRS currently requires that the index co-reside with the data. This would not be feasible in a distributed environment. Second, PIRS currently does not provide acceptable search speeds for more than 256 megabytes of indexed data, because the probability that an indexed word occurs in all blocks increases with data size[4]. Essence adds another factor of 10-20 of index size reduction. We are currently considering several approaches to these scaling problems.

5 Summary

The Internet's rapidly growing data volume, user base, and data diversity will create difficult problems for the current set of resource discovery tools. Future tools must scale with the diversity of information systems, number of users, and size of the information space.

With growing information diversity, techniques are need to gather data from heterogeneous sources, and sort through the inherent inconsistency and incompleteness. Internet Research Task Force efforts

---

[4] There is an upper bound on block size, based on I/O costs for sequential searching.

in this realm focus on operation-mapping gateways and data-mapping protocols.

With a growing user base, significantly more load will be placed on Internet links and servers. This load will require much more highly replicated information than currently exists, and will require measurement studies to decide what data should be replicated, and where to place copies. IRTF efforts in this realm focus on massive caching and replication algorithms, and on data access measurements.

As the volume of information continues to grow, organizing and browsing data break down as primary means for supporting resource discovery. At this scale, discovery systems will need to support scalable content-based search mechanisms. Current systems tend to strike a compromise between index representativeness and space efficiency. Future systems will need to support indexes that are both representative and space efficient. IRTF efforts in this realm focus on scalable content-based searching algorithms.

Acknowledgements

We thank Alan Emtage for providing us with the Archie logs that led to some of the results in Section 3.2. Panos Tsirigotis implemented the software needed for analyzing this data.

Bowman was supported for this work in part by the National Science Foundation under grants CDA-8914587 and CDA-8914587AO2, and an equipment grant from Sun Microsystems, Inc. Danzig was supported for this work in part by the Air Force Office of Scientific Research, under Award Number F49620-93-1-0082. Schwartz was supported for this work in part by the National Science Foundation under grant numbers NCR-9105372 and NCR-9204853, and by a grant from Sun Microsystems' Collaborative Research Program.

References

[AFM92] S. Armstrong, A. Freier, and K. Marzullo. Multicast transport protocol. RFC 1301, February 1992.

[ANS91] ANSI. *ANSI Z39.50.* American National Standards Institute, May 1991. Version 2, Third Draft.

[BD93] C. M. Bowman and C. Dharap. The Enterprise distributed white-pages service. *Proceedings of the USENIX Winter Conference*, pages 349–360, January 1993.

[Ber86] U.C. Berkeley. *UNIX User's Manual Reference Guide, 4.3 BSD.* Computer Systems Research Group, Computer Science Division, EECS, University of California, Berkeley, April 1986.

[BLCGP92] T. Berners-Lee, R. Cailliau, J-F. Groff, and B. Pollermann. World-Wide Web: The information universe. *Electronic Networking: Research, Applications and Policy*, 1(2):52–58, Spring 1992.

[BLNS82] A. Birrell, R. Levin, R. M. Needham, and M. D. Schroeder. Grapevine: An exercise in distributed computing. *Communications of the ACM*, 25(4):260–274, April 1982.

[BPY90] M. Bowman, L. L. Peterson, and A. Yeatts. Univers: An attribute-based name server. *Software - Practice & Experience*, 20(4):403–424, April 1990.

[Cat92] V. Cate. Alex - a global filesystem. *Proceedings of the Usenix File Systems Workshop*, pages 1–11, May 1992.

[CM86] D. Comer and T. P. Murtaugh. The Tilde file naming scheme. *Proceedings of the Sixth International Conference on Distributed Computing Systems*, pages 509–514, May 1986.

[DC90] S. Deering and D. Cheriton. Multicast routing in datagram internetworks and extended lans. *ACM Transactions on Computer Systems*, 8(2):85–110, May 1990.

[DHS93] P. B. Danzig, R. S. Hall, and M. F. Schwartz. A case for caching file objects inside internetworks. March 1993. Technical Report CU-CS-642-93, Department of Computer Science, University of Colorado, Boulder, Colorado.

[DLO92] P. B. Danzig, S.-H. Li, and K. Obraczka. Distributed indexing of autonomous Internet services. *Computing Systems*, 5(4):433–459, Fall 1992.

[DOK92] P. B. Danzig, K. Obraczka, and A. Kumar. An analysis of wide-area name server traffic: A study of the Domain Name System. *ACM SIGCOMM 92 Conference*, pages 281–292, August 1992.

[Dro90] R. E. Droms. Access to heterogeneous directory services. *Proceedings of the IEEE INFOCOM '90*, June 1990.

[ED92] A. Emtage and P. Deutsch. archie: An electronic directory service for the Internet. *Proceedings of the Winter 1992 Usenix Conference*, pages 93–110, January 1992.

[FJP90] J. C. French, A. K. Jones, and J. L. Pfaltz. Scientific database management. Technical report, August 1990. Final Workshop Report.

[Fos92] S. Foster. About the Veronica service, November, 1992. Electronic bulletin board posting on the comp.infosystems.gopher newsgroup.

[GJSJ91] D. K. Gifford, P. Jouvelot, M. A. Sheldon, and J. W. O'Toole Jr. Semantic file systems. In *Proceedings of 13th ACM Symposium on Operating Systems Principles*, pages 16–25, October 1991.

[HS93a] D. Hardy and M. F. Schwartz. Extending the WAIS paradigm to support dynamic information spaces. Technical report, 1993. Department of Computer Science, University of Colorado, Boulder, Colorado. In preparation.

[HS93b] D. Hardy and M. F. Schwartz. Essence: A resource discovery system based on semantic file indexing. *Proceedings of the USENIX Winter Conference*, pages 361–374, January 1993.

[Int88] International Organization for Standardization. Information Processing Systems – Open Systems Interconnection – The Directory – Overview of Concepts, Models, and Service. Technical report, International Organization for Standardization and International Electrotechnical Committee, December 1988. International Standard 9594-1.

[KM91] B. Khale and A. Medlar. An information system for corporate users: Wide Area Information Servers. *ConneXions - The Interoperability Report*, 5(11):2–9, November 1991.

[Lam86] B. Lampson. Designing a global name service. *ACM Principles of Distributed Computing*, pages 1–10, August 1986.

[Lot92] M. Lottor. Internet growth (1981-1991). Technical report, January 1992.

[McC92] M. McCahill. The Internet Gopher: A distributed server information system. *ConneXions - The Interoperability Report*, 6(7):10–14, July 1992.

[MD88] P. Mockapetris and K. Dunlap. Development of the Domain Name System. *Proc. of the ACM SIGCOMM '88, Stanford, California*, pages 11–21, August 1988.

[Moc87] P. Mockapetris. Domain names - concepts and facilities. RFC 1034, November 1987.

[MW93] U. Manber and S. Wu. A two-level approach to information retrieval. Technical report, January 1993. University of Arizona Technical Report.

[NAS90] NASA. *Workshop Record of the Second Annual CESDIS Workshop - The Role of Computer Science in the Mission to Planet Earth.* Center of Excellence in Space Data and Information Sciences, October 1990.

[Neu90] B. Clifford Neuman. The virtual system model: A scalable approach to organizing large systems: A thesis proposal. Technical Report TR-90-05-01, University of Washington, Seattle, May 1990.

[Neu92] B. C. Neuman. Prospero: A tool for organizing internet resources. *Electronic Networking: Research, Applications, and Policy*, 2(1):30–37, Spring 1992.

[ODDA93] K. Obraczka, P. Danzig, D. DeLucia, and N. Alaam. Massive replication of data in autonomous internetworks. *Technical Report, University of Southern California*, March 1993.

[Pos82] J. B. Postel. RFC 821: Simple Mail Transfer Protocol. August 1982.

[Qua90] J. Quarterman. *The Matrix - Computer Networks and Conferencing Systems Worldwide*. Digital Press, 1990.

[Sal86] G. Salton. Another look at automatic text-retrieval systems. *Communications of the ACM*, 29(7):648–656, July 1986.

[SBN84] M. Schroeder, A. Birrell, and R. Needham. Experience with Grapevine: The growth of a distributed system. *ACM Trans. on Computer Systems*, 2(1):3–23, February 1984.

[SEKN92] M. F. Schwartz, A. Emtage, B. Kahle, and B. C. Neuman. A comparison of Internet resource discovery approaches. *Computing Systems*, 5(4):461–493, Fall 1992.

[ST91] M. F. Schwartz and P. G. Tsirigotis. Experience with a semantically cognizant internet white pages directory tool. *Journal of Internetworking: Research and Experience*, 2(1):23–50, March 1991.

[Sun92] SunSoft. *SearchIt 1.0*. SunSoft, Inc., 1992. User's Guide.

[WCS93] D. C. M. Wood, S. S. Coleman, and M. F. Schwartz. Fremont: A system for discovering network characteristics and problems. *Proceedings of the USENIX Winter Conference*, pages 335–348, January 1993.

[Wei85] P. Weiss. *Yellow Pages Protocol Specification*. Sun Microsystems, Inc., 1985.

[Zim90] D. Zimmerman. RFC 1288: The finger user information protocol. November 1990.