Pharos: A Scalable Distributed Architecture for Locating Heterogeneous Information Sources

Version 1.0.2

R. Dolin D. Agrawal L. Dillon A. El Abbadi
Department of Computer Science
University of California
Santa Barbara, CA 93106

July 11, 1996

Abstract

This paper presents the design of Pharos: a scalable distributed architecture for locating heterogeneous information sources. The system incorporates a hierarchical metadata structure into a multi-level retrieval system. Queries are resolved through an iterative decision-making process. The first step retrieves coarse-grain metadata, about all sources, stored on local, massively replicated, *high-level* servers. Further steps retrieve more detailed metadata, about a greatly reduced set of sources, stored on remote, sparsely replicated, topic-based *mid-level* servers. We describe the structure, distribution, and retrieval of the metadata in Pharos to enable users to locate desirable information sources over the Internet.

Contents

1 Introduction

Finding appropriate sources of information on the Internet increases in difficulty with the increasing number of Internet users, the number of sources providing electronic documents, the number of systems for indexing these documents, and the volume of available data. An information source is any site in the network that provides electronic documents such as journal articles, books, hypertext pages, images, files, etc. Prior to digital networks, the number of potential information sources was relatively limited, mainly by locality; hence, a user could discover what was available by browsing individual sources. In a wide-area network such as the Internet, information resides in numerous heterogeneous systems, which are geographically distributed; each such distributed system may provide many information sources. Individual systems provide various discovery tools, e.g., Archie, Veronica, Yahoo. However, users are still easily lost in the vast space of documents that are indexed by these systems. No system-independent location facility currently exists. What was once a problem of limited choice has become a problem of too much choice. The existence of thousands of newsgroups, World-Wide Web (WWW) sites, FTP archives, and Digital Libraries (DL's)[1] renders traditional methods of locating good information sources infeasible.

Our work focuses on the selection of *collections* of documents rather than on particular documents themselves. Thus, we are not directly addressing the problem of a user finding a particular document via author, title, etc. However, a particular author or title is generally included in one or more information domain, such as subject area, geographical region, time period, image feature, type of business, etc. By focusing on finding collections which are characterized by such domains, we believe that users are likely to find those authors and titles which are relevant to the concepts underlying specialized queries. The reason for such a change in focus, as will hopefully be made clear later, is that we believe that locating collections by domains is a much more scalable methodology than locating particular authors, titles, etc.

Current WWW indexes, such as AltaVista[1], Lycos [23], and Yahoo [33], which are designed for locating information on the Internet, are limited in several ways. Notably, these systems do not scale to handle increasingly large numbers of search requests due to limited network bandwidth and server power. Furthermore, because they focus on keyword matching, they are usually unable to provide adequate simultaneous *precision* and *recall*[2]. Existing WWW indexing systems utilize either small, hand-made, hierarchical lists, as in Yahoo, or word-matching on huge document-spaces, as in AltaVista. Standard techniques from the field of Automatic Text Retrieval [26], such as relevance feedback and

[1]In our present context, the main requirement for an information source to be called a *Digital Library* is that it contain its own catalog and query system.

[2]*Precision* is the fraction of returned documents that are relevant to a query, while *recall* is the fraction of relevant documents in a collection that are returned.

local/global term weightings, aid in matching query terms to relevant documents; however, these techniques are seldom used for locating sources. Another factor that limits most existing WWW indexes is their inability to share index information. This lack of sharing results in highly duplicated document fetching and indexing. As the number of indexing sites grows, determining which ones contain the most useful information for a given query becomes increasingly difficult. Furthermore, few WWW indexing systems give sources control over how often they are indexed, or provide for automatic updates when information becomes stale. Another problem with many WWW indexes is that they index the actual texts of documents. This approach is not appropriate for indexing catalogs, especially those which list documents which are not available on-line, and it does not extend to use with non-text documents, such as maps, images, and sound and video files. Information sources that make their documents available only as query results have no mechanism of including their contents within such indexing systems. WWW indexing systems that index document texts require their own indexing techniques and cannot use the metadata that a source extracts about itself.

Problems of locating sources of information, or *resource discovery* [4], are of particular interest to the DL community [14]. Emerging DL systems must deal with storing, locating, cataloging, distributing, and retrieving large numbers of diverse documents in a distributed, heterogeneous environment. Several projects in the NSF/NASA/ARPA Digital Library Initiative Program [6, 24] are therefore investigating these problems. The Alexandria Digital Library (ADL) Project [2] is focusing on indexing spatial information, as well as on the storage, distribution, and retrieval of large (spatial) images. The content and form of the metadata required to classify geographically referenced data is a central research issue. The Illinois Digital Library Project [27] is building a prototype of the *Interspace*, which presents the Internet as a single space of highly interlinked, distributed information. Key research topics include placement of query terms within existing classification schemes, discovery of relationships between different thesauri, and automatic classification through the analysis of document collections [7, 8, 9]. The University of Michigan Digital Library Project [11] is designing an agent-based DL system; user interface agents, mediation agents, and collection agents cooperate to allow concurrent searching of multiple collections.

The *gGlOSS* resource discovery system [15] represents sources by vectors of term frequencies. It is similar, in this regard, to the SMART automated text retrieval system [26], replacing the two-dimensional term/document matrix by a term/collection matrix. SMART and *gGlOSS* are only two of several automatic text retrieval systems. Other notable systems include ConQuest [10], which uses a static semantic network based on dictionary definitions, and Latent Semantic Indexing (LSI) [3], which uses a dynamic semantic network based on word co-occurrences within documents in a collection. Content Routing [28] is a system in which queries get routed to the available servers based on the expected relevance of the server to the query. This system also propagates descriptions of server contents through a hierarchy of information servers. Harvest [5] automatically indexes documents within a source and distributes these indexes. It can be used to describe WWW sites, newsgroups, repositories of technical reports, etc. and includes subsystems for caching and replicating the indexes.

The above approaches do not provide solutions or information architectures that scale well in terms of the information gathering with the growing diversity of data, the information dispersion with the growing user base, and the information visualization with the growing data volume [4]. In this paper, we describe a design of a distributed architecture for locating desirable information sources entitled Pharos, which scales in all three aspects. We achieve this scalability by using a hierarchical metadata structure and a highly decentralized metadata distribution, storage and retrieval mechanism. Furthermore, our approach provides a significant level of autonomy to the individual information sources to distribute their metadata information. Pharos does not use a simple linear ranking of all sources as is done in standard vector-based text retrieval techniques. Instead, it provides users greater freedom in the visualization and selection criteria of their final set. Finally, by supporting an iterative query methodology, Pharos allows users to benefit from relevance feedback techniques.

The next section motivates an iterative, multi-level query method to locate information sources, and gives an overview of our approach. Section 3 describes the hierarchical metadata structure needed for such an approach. Once that is completed, we give a more detailed description of the architecture of the system in Section 4. Finally, we conclude with a description of the contributions of this work and possible design extensions.

2 Design Overview

2.1 Motivation

We consider a modern Internet environment with a large number of users and information sources, including WWW sites, FTP archives, Digital Libraries, file systems, etc. Attempting to locate a small set of sources that best fit user queries requires detailed knowledge of the holdings at each source. There are two extreme approaches for acquiring this metadata. The first, which we informally call the remote approach, is to query each source dynamically in turn (or possibly in parallel) without storing any local metadata. The second, which we informally call the local approach, is for each user to store locally detailed information about the holdings of each source and not to request anything remotely (until a final set of sources has been selected for direct access). Both of these extremes are impractical. The remote approach would take too long even for a single user query, let alone a large number of simultaneous ones. The local approach needs to have sufficient information for any given query; we believe that the amount of metadata required for accurate source comparison is too large to be practical due to both the storage and network requirements.

We therefore have to use a hybrid solution. We attempt to balance the location of the metadata's storage with network traffic by using an iterative, multi-level query method. A limited amount of metadata about each source is massively replicated among "high-level" servers. These servers are locally situated within an organization such as a university campus or a corporate network. They receive requests from local users about most or all known sources and reply with only enough metadata to allow for a rough comparison of sources. More detailed metadata is stored in specialized, sparsely replicated "mid-level" servers. These servers receive requests from remote users about a small set of sources and supply greater detail than the high-level servers. All high-level and mid-level servers store information about each source. However, while all high-level servers store the same metadata, each mid-level server (up to replication) stores unique metadata within a sub-area of a single, pre-specified, domain-specific classification tree, or *taxonomy*. For example, we expect separate mid-level servers specializing in subject areas such as history, art, physics; in geographical regions such as North America, Africa, Asia; etc. We call any such sub-area of a particular information domain a *sub-domain*. This design solves the problem of the remote approach by providing a fast, scalable method of receiving any needed metadata. It also solves the storage problem of the local approach by distributing the metadata throughout the network and requiring only minimal storage within a user's local environment. Finally, network traffic is reduced in several ways. Smaller local storage reduces the amount of broadcast-type traffic generated by the sources for update purposes, while the mid-level servers reduce the amount of query traffic to the sources generated by users. Furthermore, having users communicate with local servers for initial source filtering greatly reduces network traffic.

2.2 Example Query

As an example query, suppose that a student is studying the following topic: "A Comparison of Political Music in 1967: San Francisco, Shanghai, and Cairo." We performed a search through AltaVista, Lycos, Yahoo, and the University of California Melvyl system's online general catalog, described in Appendix A. AltaVista, Lycos, and Yahoo found little of value for this study. Furthermore, the precision was very low, on the order of 1%, requiring excessive viewing of many irrelevant documents in order to locate the rare relevant ones. Melvyl, on the other hand, returned 44 items of potential interest, such as "When the music's over: the story of political pop," "Female college students in China," "Rhythm and resistance: explorations in the political uses of popular music," "Qira'at munawi'ah," and several recordings. How do we inform the student that, in this case, Melvyl is a potentially good information source?

In our proposed system, the student attempts to find sources as follows. Initially, at the client-side user interface (UI), she specifies the query in terms of relevant information taxonomies and significant sub-domains within them. The taxonomies involved in this search, for example, include subject, geographical region, and time period. Within each of these general taxonomies, the student is prompted to provide more specific sub-domains that match her query. For the subject domain, the UI attempts to locate the query terms or their synonyms within the taxonomy. If they match several sub-domains, the UI prompts her for which ones to include. For example, the phrase *political music* might be covered by political science, history, and music; sources that excel in these areas are potentially useful for finding information about her query. For the geography taxonomy, the student specifies a longitude/latitude bounding region, or a place-name which is checked against a *gazetteer*[3]. In the latter case, the gazetteer would list the possible matches with their bounding coordinates, and the student could select the appropriate locations.

Once the student has specified the appropriate sub-domains within the three taxonomies, the UI then queries the local high-level server and gives highest weights to those sources that contain relatively more information 1) within the subject taxonomy, in, say, politics, history, or music; 2) within the time period taxonomy, about the 1960's; and 3) within the geography taxonomy, in SW North America, SE Asia, or NE Africa. From the weightings, the student selects some initial set of sources, say the best thousand based on the high-level metadata.[4] Next, the UI queries the relevant seven remote mid-level servers for more detailed metadata about the sources included in the initial set: a political subject server, a history subject server, a music subject server, a 1960-1969 time period server, and the three geographic region servers covering SW North America, SE Asia, and NE Africa. These mid-level servers return their much more detailed and specialized metadata about the requested sources. For example, the political subject server provides metadata about the percentage of holdings among the initial set of sources particular to political music; the 1960-1969 time period server provides metadata about the percentage of holdings covering 1967; and the SW North America geographic region server provides metadata about the percentage of holdings dealing with the California area. The UI then merges the information returned from the various mid-level servers and presents them to the student. This information would allow the student to select a small set of sources that are likely to have information about her original query, and to request data directly from these sources.

2.3 Multi-Level Approach

Our multi-level design overview is depicted in Figure 1. This design assumes an iterative query approach where each source's metadata description is in a standardized format

[3]A *gazetteer* is a mapping from symbolic names to geographical coordinates.

[4]Users select sources via a non-textual visualization rather than a list in order to simplify the process.

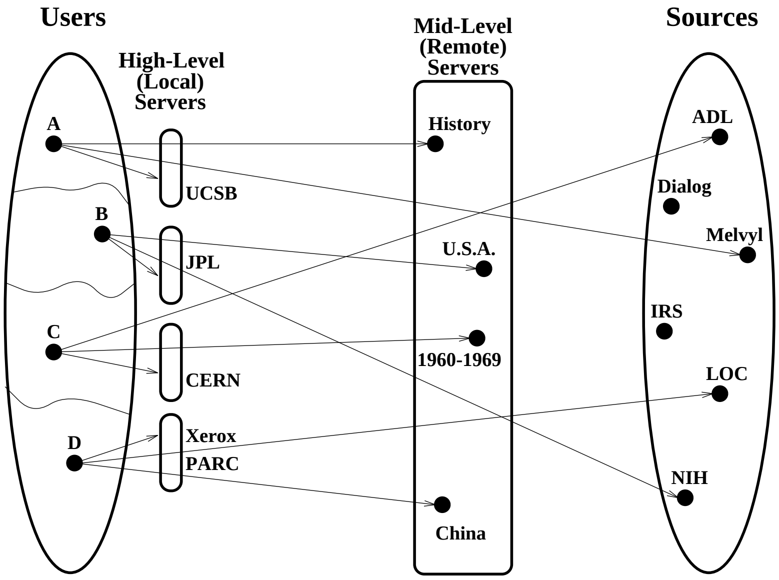

Figure 1: Retrieval Design Overview

and is broken down into components related to the composition of the holdings with respect to various information taxonomies and metadata levels. This metadata is propagated and stored at the different servers, such that coarse-grain information resides on massively replicated *high-level* servers and fine-grain information resides on sparsely replicated *mid-level* servers. Assume that a student at UCSB is performing the search described above. She first queries the high-level server at UCSB. This server supplies enough coarse-grain metadata to provide a rough filtering of the sources. With a greatly reduced list of sources (from $\sim 10^6$ potential sources available on the network to, say, $\sim 10^3$ relevant sources for the specific query), the student then requests more metadata from, for example, the History mid-level server, situated remotely on the network. This server supplies much greater detailed information about "history" from the student's list of sources (thus further reducing the potentially relevant sources, say from $\sim 10^3$ to $\sim 10^1$).

Given such a retrieval system, we need a corresponding metadata distribution system such that the high-level metadata is distributed widely and the mid-level metadata is distributed selectively. The wide distribution mechanism in Pharos is modeled after the distribution of USENET news via NNTP [17, 19]. There are many news servers on the Internet with basically the same replicated data. When someone posts a news article, it gets distributed to all the news servers that serve the groups or news hierarchies within which the article resides. On the receiving side, when someone opens a newsgroup or reads a news article, that information is fetched from the local news server. Similarly, there would be many high-level metadata servers on the network, all with basically identical data. When a source sends out its high-level metadata, that metadata gets distributed to all high-level servers.

Unlike the storage mechanism for high-level metadata, mid-level metadata from different taxonomies in Pharos are not all stored together. Each mid-level server stores all the metadata about each source that is related to a unique sub-domain of a particular information taxonomy. These servers are not generally resident within users' local areas, but are sparsely located at remote sites. When a source sends out its mid-level metadata, it sends the different components to the corresponding servers in a point-to-point manner. The distribution of the mid-level metadata in Pharos is modeled after the distribution of indexes in Harvest [5]. The Harvest system includes efficient point-to-point transfer of metadata, server replication, and caching. It also provides for structured querying of the servers.

3 Metadata Structure

The metadata must be designed to support a multi-level information system and it should match the intended queries as well as the retrieval system. Therefore, metadata should be grouped according to its informational relationship and designed around a hierarchical structure which lends itself to progressively refined queries. Pharos employs information classifications that fit common query methods, such as a subject-based hierarchy, a geographical hierarchy, and a temporal hierarchy. Once these hierarchies have been defined, we include the top portions in the high-level metadata, and lower portions in the mid-level metadata.

3.1 Information Hierarchies

An *information hierarchy* is a topic-based classification tree, as shown in Figure 2. In order to describe the collection at a source, we quantify the number of documents with content relevant to each part of an information hierarchy. As mentioned above, the initial design of Pharos employs three information hierarchies: subject, geography, and temporal hierarchies.

The subject hierarchy is modeled after the Library of Congress's *LC Classification* system [21], which has a fixed list of *categories.* For a given query, terms must be deduced that match these categories. This classification system has several advantages: it is very extensive, it is familiar to most of the library classification community, it is open-ended, and it is used by most academic libraries in the U.S.A. While there are problems with using a fixed list for query terms [12], there are several problems with using vocabularies that are based on the terms found in each document collection. For example, it is difficult to compare sources that do not use the same terms, and we cannot expect different

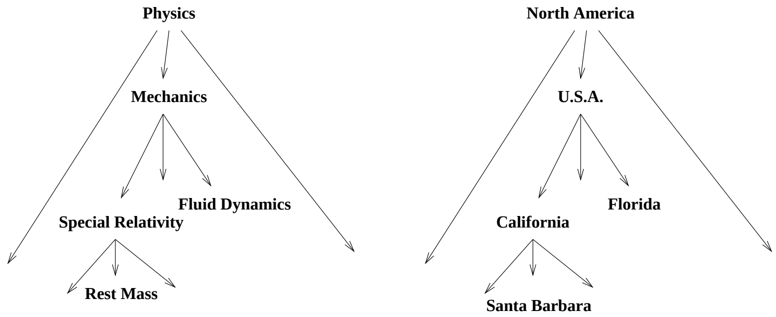

Figure 2: Information Hierarchies

collections to yield the same vocabularies. Moreover, without a static set of terms, it is difficult to construct comparable *hierarchical* structures that can be used to build source metadata within a multi-level approach. The Pharos UI bridges the gap between the words used by the user and the fixed terms in the subject hierarchy. Techniques such as those described in [12, 7] specifically address the problem of finding the relationships between query terms and potentially different document terms. We plan, therefore, to place users' concepts within appropriate sub-domains in the subject hierarchy, while at the same time fixing the categories so that it is easier to accurately compare sources.

A simple geography information hierarchy is composed by tiling the Earth's surface into progressively smaller, hierarchical, longitude/latitude squares. If the highest level of the hierarchy represents the regions of the Earth's surface defined by $45^\circ \times 45^\circ$ grid lines, as in Figure 3, the result is 32 top-level tiles. In this figure, the actual data density of the $\sim 421,000$ documents of the May, 1996 Alexandria Digital Library (ADL) holdings is shown: the lighter the area in a tile, the more documents it contains. The next level of the hierarchy is formed by sub-dividing each top-level tile by $5^\circ \times 5^\circ$ grid lines yielding 2,592 sub-tiles. Each top-level tile has 81 sub-tiles. This sub-tiling process is continued to create a spatial hierarchy of desired levels of details.

The temporal information hierarchy needs to cover all references to time, including future time. The majority of temporal references, however, deal with the last 300 years. For example, Microsoft's Encarta Encyclopedia 96 allows articles to be looked up by "time". Of the 13,065 temporally referenced articles, 3,731 relate to time before the year 1700; 1,762 relate to the 1700's; 5,487 relate to the 1800's; and 5,641 relate to the 1900's. We therefore prefer to have a higher level of granularity for the time periods nearer to 'now' than those further away on a time-line. By using a modified logarithmic scale, which is approximately linear between the years 1700 and 2000, we can achieve a more even distribution of time-referenced documents. This hierarchy is detailed in Appendix B. Because a document's

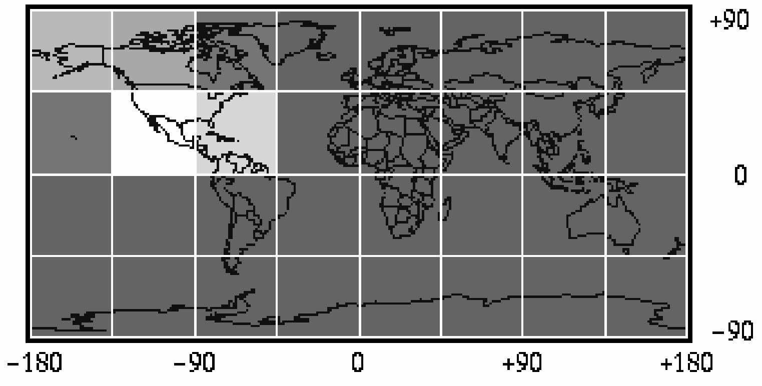

Figure 3: $45^\circ \times 45^\circ$ Grid, 32 Elements

publication date is potentially as important to temporal queries as its time-related content, we use two temporal hierarchies: one for publication dates and another for document content.

3.2 Information Taxonomies

An information *taxonomy* is defined as a static hierarchy within which documents can be classified, usually according to their content. A taxonomy is assumed to be a tree structure such that each node in the tree has two attributes: a label and a numerical value. The label denotes a word, term, or phrase from an information hierarchy. The numerical value of each node of a taxonomy is called the *coverage* of the node. It is used to describe a document collection by quantifying the number of documents with content relevant to the meaning of the label of the node. Since most documents cannot be classified completely within a single leaf of a classification tree, we assign portions of a document to different nodes in the tree. Thus for each node in the taxonomy there is a total number of *document equivalents* associated with it. Intuitively, a node's coverage indicates what fraction of a document collection is classifiable within a node's label. In order to calculate this value, we first allocate fractions of each document to nodes in the tree in such a way that not more than 100% of the document is allocated. We allocate as much of a document as we can to nodes closest to the leaves of the tree. In this way, a document is classified as much as possible within the most specific sub-domains of the hierarchy. Once all the documents in a collection have been fractionally allocated within the tree, the coverage values can be computed. The children of a parent node have labels which represent information sub-domains which are wholly or partly contained within the sub-domain represented by the

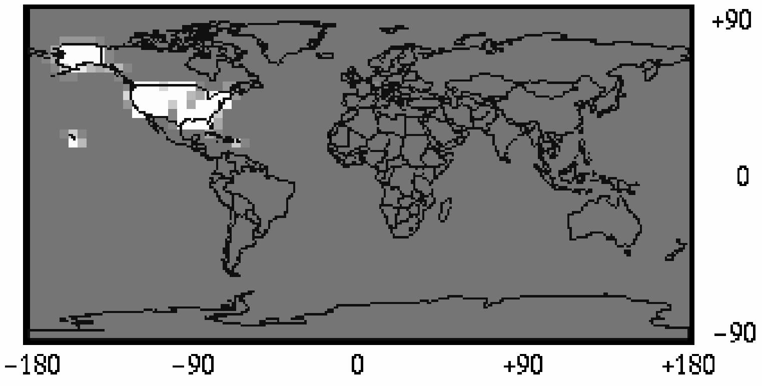

Figure 4: $5^\circ \times 5^\circ$ Grid, 2,592 Elements

label of their parent. By using the notion of coverage, this containment property is captured numerically by defining the coverage at a node to be the sum of the coverages of its children plus the fraction of the collection assigned to it but not to its children. For a leaf node, this is just the fraction of the collection assigned to it. The number of document equivalents associated with a node is its coverage times the total number of documents in the collection.

As an example, consider the geography information hierarchy discussed previously. We create a taxonomy by assigning the $45^\circ \times 45^\circ$ tiles to the depth-1 nodes, the $5^\circ \times 5^\circ$ tiles as depth-2 nodes, etc. Within the ADL collection of $\sim 421,000$ documents shown in Figure 3, the region over Hawaii, defined by the (longitude, latitude) coordinate pair [upper-left=$(-180^\circ, 45^\circ)$, bottom-right=$(-135^\circ, 0^\circ)$], contains $\sim 5,700$ document equivalents. Thus, within the geography taxonomy, this node is assigned a coverage of $\sim$0.014. To calculate the coverages of the depth-2 nodes, we break down the depth-1 nodes into $5^\circ \times 5^\circ$ tiles, as shown in Figure 4. This figure shows another view of the same ADL holdings as in Figure 3. The three depth-2 nodes which contain the Hawaiian Islands contain approximately 1500, 2400, and 1100 document equivalents, and so are assigned coverages of $\sim$0.0038, $\sim$0.0060, and $\sim$0.0028, respectively.

Clearly Figure 4 gives a much more precise display of the ADL holdings. However, the 2,592 data elements prohibit that level of detail from being included in the high-level metadata, as discussed in more detail in the next section. On the other hand, Figure 3 does not necessarily include enough detail to be useful. As a case in point, consider a user looking for information about Hawaii. The large-scale region over Hawaii does not appear to contain a sufficient amount of data to make the ADL an attractive source; the average coverage of the 81 depth-2 children is only $\sim$0.00017, implying that there are only approximately 220 documents for Hawaii. On closer examination, however, the small-scale map shows that the

ADL is potentially a very good source for such a user; there are actually over 5000 documents for that region. With only 5% of the depth-1 node's children containing 97% of its data, a high-level comparison involving the ADL might exclude it and the depth-2 metadata would never be requested. This problem of statistically losing lower level information, and hence potentially valuable sources, can occur within any information taxonomy whenever a small fraction of a node's children contain a vast majority of the node's coverage. This case can be considered a *special collection* within a source's taxonomy. We handle special collections by providing a special metadata attribute.

The geography taxonomy is very different from the subject taxonomy in that it is not a word-based hierarchy, but a spatial hierarchy. To put this in perspective, Microsoft's Encarta Encyclopedia 96 allows articles to be looked up by "place". However, while a query on 'North America' yields 296 articles, a query on 'United States' yields 3,958 articles. These results do not occur under a geography taxonomy which would place 'United States' within 'North America'; our definition of a taxonomy requires that the spatial nodes covering North America include at least all those articles assigned to the nodes covering the United States.

3.3 Metadata Levels

We now describe the relevant metadata records that need to be included in the high- and low-level servers. The main constraint in regard to the high-level server is size. The total size T_H of information distributed to the high-level servers is the size S_H of each high-level metadata record describing a given source times the total number N_S of sources described. The overriding constraint is that T_H not grow beyond some maximum value. Using the newsgroup distribution as an example, we assume that each high-level server stores under 10 GB of data. Assuming that we want to be able to handle a large number of sources, say $N_S \leq 10^6$, we derive a value of S_H of roughly 10 KB. A high-level metadata record needs to summarize the coverage information related to various nodes in an information taxonomy. Given these values for N_S and S_H, and the width of the upper levels of the taxonomy trees, we can determine which nodes of the taxonomies should be included in the high-level metadata.[5]

The size of a high-level metadata record is dependent on the number of taxonomies used and the number of nodes that are included from each taxonomy. Although the number of taxonomies a source uses might vary for a given taxonomy, we must specify which nodes to include in the high-level metadata. High-level metadata refers to 1) any taxonomy-independent source description, and 2) the high-level portions of all taxonomies within which a source's collection has been classified. We determine the number of nodes to include based

[5]In the future, we may allow N_S to grow larger by using a single metadata record to represent a hierarchical collection of related sources (e.g., UC source representing the nine UC campus libraries). In order to allow this, the structure of the metadata must be independent of the number of sources described.

Attribute Name	**Type**	**Range**	**Description:** *Example*
HL_Desc_Ver	String	-	Metadata descriptor version number: *0.0.7*
Src_ID	String	-	Name of source: *Alexandria Digital Library*
Src_Type	String	-	Type of source: *DLIB*
Src_Loc	String	-	Geographical location of source (if relevant): *Santa Barbara, California*
Src_Sch	String	-	Availability schedule of source (if relevant): *Closed Mondays*
Siz_Col	Real[3]	0.0+	Collection size in MB, pages, and items: *1.843E6,,8.91E5*
Siz_Cat	Real[2]	0.0+	Catalog size in MB and items: *2.172E6,3.892E6*
Count_Tax	Integer	0+	Number of taxonomies used: *3*
Spec	String[M]	-	List of M Special Collections: *Geography:BE16*
Net_Band	Real	0+	Network bandwidth (Kbps): *1.5E3*
Net_Avg_Util	Real	0.0 - 1.0	Avg. network utilization: *0.47*
Net_Avg_Delays	Integer[N]	0+	Avg. network delay (ms) to N beacons[16]: *,200,,521,,,147,*
Net_Avg_Thru	Integer[N]	0+	Avg. network throughput (Kbps) to N beacons[16]: *,117,,,,,246,*
Count_Avg_Hits	Real	0+	Avg. number of accesses (hits/day): *1.237E5*
Pol_Charg	String	-	Charging Policy: *See http://www.alex.ucsb.edu/doc/charging.html*
Pol_Lend	String	-	Lending Policy: *Must be UC affiliate*
Count_Src	Integer	1+	Number of sources: *1*

Table 1: Taxonomy-Independent High-Level Metadata

on our estimated size, S_H, of a high-level record, the number of taxonomies included, N_T, and the size, S_I, of the taxonomy-independent metadata. For simplicity, we assume that each taxonomy's node metadata is the same size, S_N, and that the same number of nodes, $N_H(T)$, will be included from each taxonomy T. Thus we have $S_H = S_I + S_N * N_H(T) * N_T$, or $N_H(T) = (S_H - S_I)/(S_N * N_T)$. Assuming $S_H \approx$ 10KB; $S_I \approx$ 1KB; $S_N \approx$ 100B; and $N_T \leq 4$, we conclude that $N_H(T) \approx 22$. That is, we assume that we can include on the order of 20 to 30 nodes from each taxonomy in our high-level metadata records.

Table 1 shows the portion of the high-level metadata that is independent of the taxonomies (of size S_I). In the `Description` column, we illustrate the corresponding attribute with an example. Other than the source identification information, these attributes can be viewed as quality and cost factors that are presented to the user through the UI. The user can then consider a range of factors in determining specific sources relevant to current queries. The first attribute, `HL_Desc_Ver`, is the version of the metadata schema and is used to allow for backward compatibility. The `Src_*` attributes describe basic source identification information. `Siz_Col` describes the size of the collection in MB, pages, and items. These units are used to calculate the fractional assignment of documents in three units. The reason that we use multiple units is that they may yield very different ratios. For example, an image library whose images may range from over 100MB/image to less than 10KB/image will show very different geographic coverages in MB than in items. `Siz_Cat` describes the size of the catalog in MB and items. `Count_Tax` is set to the number of taxonomies used for classification by the source. `Spec` is the list of character strings used to describe special collections, described earlier. The set of parameters `Net_*` are included for connectivity estimation between the users and the sources [16]. `Count_Avg_Hits` describes the average number of accesses the source receives per day. Additional attributes are available for various library policies relevant to charging, lending, etc. Finally, `Count_Src` denotes the number of sources included in a particular metadata record to allow condensed multi-source records. If this value is greater than one, some of the other values might not be included, such as the network parameters.

Tables 2 and 3 show the taxonomy-dependent part of the high-level metadata. Within the full metadata record for a source, the attribute-values in Table 2 are repeated once for each taxonomy. `Tax_ID` and `Tax_Desc_Ver` identify the taxonomy name and version, respectively. The next attribute, `Cov_Root`, gives the fractional amount of the collection which has been classified within the taxonomy, with respect to MB, pages, and items. The next four attributes give statistical information about the root's children (the depth-1 nodes in the taxonomy). Furthermore, within each taxonomy, the attribute-values of Table 3 are repeated once for each high-level node in the tree.

The geography taxonomy is treated as a special case in order to include factors such as image or map resolution, which are not relevant to most other taxonomies. This taxonomy

Attribute Name	**Type**	**Range**	**Description**
Tax_ID	String	-	Name of taxonomy
Tax_Desc_Ver	String	-	Taxonomy descriptor version number
Cov_Root	Real[3]	0.0 - 1.0	Root node's coverage value (MB, pages, items)
Cov_Avg	Real[3]	0.0 - 1.0	Avg. coverage of depth-1 nodes
Cov_SD	Real[3]	0.0 - 1.0	Std. Dev. of coverage of depth-1 nodes
Cov_Min	Real[3]	0.0 - 1.0	Minimum coverage of depth-1 nodes
Cov_Max	Real[3]	0.0 - 1.0	Maximum coverage of depth-1 nodes

Table 2: High-Level Metadata for Each Taxonomy T_i

Attribute Name	**Type**	**Range**	**Description**
Node_ID	String	-	Node's label
Cov_Node	Real[3]	0.0 - 1.0	Node's coverage value (MB, pages, items)
Cov_Avg	Real[3]	0.0 - 1.0	Avg. coverage of children
Cov_SD	Real[3]	0.0 - 1.0	Std. Dev. of coverage of children
Cov_Min	Real[3]	0.0 - 1.0	Minimum coverage of children
Cov_Max	Real[3]	0.0 - 1.0	Maximum coverage of children

Table 3: Metadata for Each High-Level Node in T_i

is a superset of that in Tables 2 and 3. Additional attributes are included to describe image resolution statistics which summarize all the documents which are associated with the geographical region covered by each node.

Mid-level metadata is not subject to the same space constraints as the high-level metadata for several reasons. First, because mid-level metadata is only sparsely replicated, there are not as many network constraints in distributing it. Second, each mid-level server handles only a specific sub-domain within a single taxonomy, so it needs to store only a very limited subset of all of the mid-level metadata records for any source. Third, because users request this metadata only after they have already gone through a primary filtering using the high-level metadata, mid-level query results contain data for only a relatively small number of sources. The structure of mid-level metadata is similar to the taxonomy-dependent components of the high-level metadata, but covers nodes of greater depth in the information taxonomies.

4 Implementing Pharos

We now describe the necessary components for the implementation of our system, Pharos. First, the UI is designed to prompt initially for information about the user which is stored between sessions. Next, the UI must prompt for query information. During the query process, the UI will request source metadata from the various servers as needed. Finally, it needs to present the metadata in a concise and understandable manner during each phase of the search. On the other side of the query is the process of extracting the metadata from each information source. While this could in principle be done by hand, we imagine that one of several automated processes are likely to be used. The document collection must be analyzed for clusters and those clusters must then be classified within the associated taxonomies. Between the retrieval and extraction of the metadata, we have the massively replicated, localized high-level servers and the sparsely replicated, remote mid-level servers, each receiving and sending appropriate metadata with very different network characteristics and hence different underlying network protocols.

4.1 User Interface and Metadata Retrieval

The User Interface (UI) is designed to assist the user in an iterative decision-making process [13] in which a final small set of sources is chosen through a series of refinements on an initial large set. This process uses information about the user, called the *User Profile*; information about the query, called the *Task Profile* [29]; and the high- and mid-level metadata about the sources. Although the client acquires source metadata through the network from the high- and mid-level servers, it acquires user and query information through the UI. Besides acquiring this information, the UI also presents metadata about many sources to the user

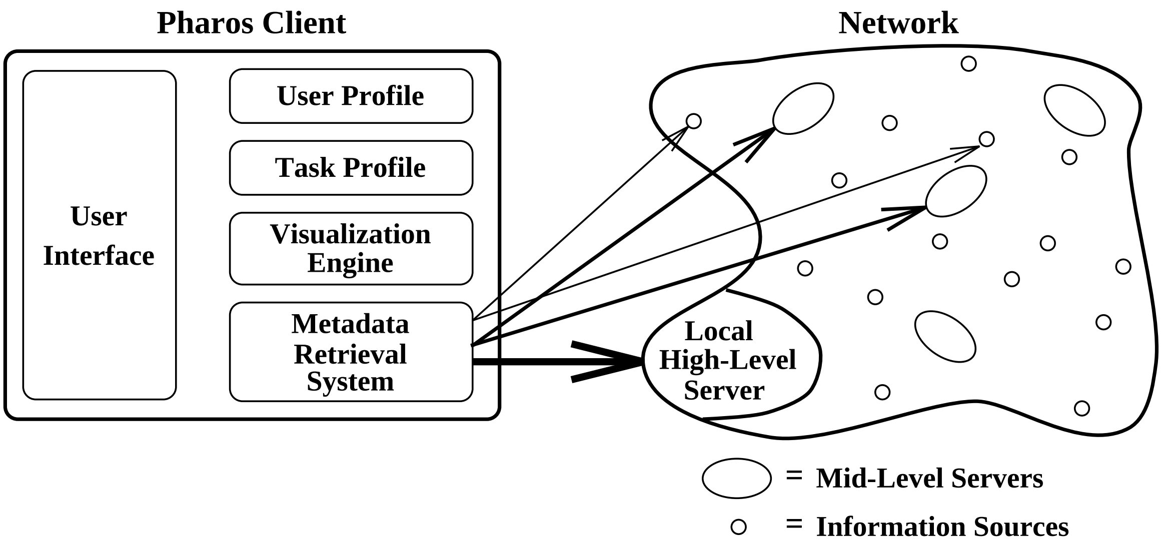

Figure 5: Pharos Client Components

in a concise and understandable manner. This allows the user to interact with that data and to select source subsets for further metadata retrieval. Figure 5 illustrates the key components of the UI subsystem in Pharos. User and Task Profiles are initialized by the users. When a query is issued, these profiles are used to send an initial metadata request to a local high-level server. The metadata returned includes information about the likely candidate sources that best match the query, subject to the User and Task Profile. The metadata is then visualized to the user within the UI. The visualization includes default but tailorable quality and cost factors. Given this information, the user then selects a set of sources, perhaps those with the highest 'value'. From this set, a new metadata request is issued to the specific mid-level servers that store information relevant to the user's query. The returned metadata is again visualized, but with greater accuracy and detail. Finally, the query results are narrowed down to a reasonably small set of sources, which are then directly queried.

The User Profile information in Pharos is used to determine which metadata parameters to retrieve, how to compare sources, and how to display the results. The user can assign relative weights to any parameter to adjust the comparison results. Other profile information include interface parameters such as which taxonomies to use by default. As the set of sources becomes small enough, factors can be considered which require direct communication with each source, such as the number of network hops. The Task Profile includes all parameters needed to specify the source search criteria. The user must de-

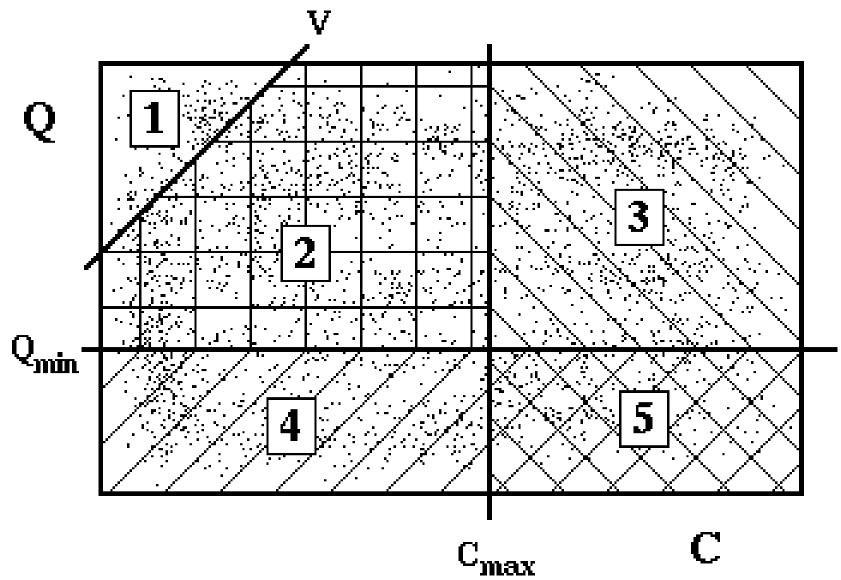

Figure 6: Simple Metadata Visualization

cide which taxonomies to use. For each taxonomy included, the user selects one or more sub-domains either by entering keywords, or by traversing the tree directly and selecting particular sub-domains. The UI must deal with keyword entries differently for each taxonomy. For the subject hierarchy, techniques used in text retrieval, such as by LSI [3], can aid in matching query terms with terms found in the hierarchy. The geography taxonomy requires a gazetteer, and the time-period taxonomy requires the equivalent of a gazetteer for time-names. Finally, as the set of sources becomes small enough, the UI attempts to match keywords with any special collections included in the set.

Since there are a large number of sources, it is impractical to list the results of a query in a text-based, tabular fashion. The level of detail of the display of each source is adjusted depending on the total number of sources being displayed. Initially, we display each source as a point in a two-dimensional scatter plot, as depicted in Figure 6. There are several ways of assigning positions to sources. One possibility is to use a quality versus cost approach to visualize the economic value of the sources [31]. The quality and cost dimension of each source is computed as a weighted sum of the relevant attributes and each source is depicted as a point in the Quality/Cost plane. The user can then graphically select sources which are above a minimum quality (regions 1, 2, and 3), below a maximum cost (regions 1, 2, and 4), and/or have the highest value (region 1). Many other forms of weightings and attribute combination are possible. For example, the ranges of values that each attribute may attain vary widely both between different attributes and also, for a given attribute, between different sources. We therefore use normalized attribute values by dividing each one by the maximum value of that attribute among all the sources. Furthermore, for values such as network bandwidth and collection size, we compare the normalized logarithm of the values so that large values do not overly dominate. Similarly, many other visualizations are possible, especially as the number of sources is reduced after filtering.

4.2 Metadata Extraction

The process of extracting the metadata from a source begins with a document collection, as portrayed in Figure 7. This collection is processed by hand or automatically. For text-based documents, it can be processed by one of several possible automatic text analysis tools, such as LSI, SMART [26], or others that are used, for example, in the NIST Text Retrieval Conferences (TREC).[6] Many of these systems read through a collection and build a matrix of weighted frequencies of the number of times a given term occurs in each document in the collection. From this matrix, documents can be placed in a multi-dimensional term-vector space. Nodes in a taxonomy can also be placed in this space and then be taken as centroids of document clusters. Depending on the relative closeness of the centroids to the various documents, we assign the fractional allocation of each document among the nodes in the taxonomy needed to calculate each node's coverage value. High-level automatic subject classification has been shown to be fairly successful, with over a 90% accuracy rate [22]. LSI has been shown to be an effective tool for automated classification [18], and experiments on automatically classifying large document sets within the LC Classification system have yielded correct classifications, under some conditions, of over 80% of newly entered documents [20]. These results indicate that subject-based automatic classification is currently or soon to be sufficiently accurate to characterize sources for comparison purposes, where even order of magnitude estimates can greatly aid in filtering out most irrelevant sources.

While the TREC and related work focuses on subject-based text retrieval, one could use a gazetteer or time-name table to identify geographic or temporal references within text-based documents, or use more sophisticated techniques such as those used in GIPSY [32]. For maps, aerial photographs, and satellite images, documents are often cataloged with spatial extents, and the geographic clustering and classification process is much simpler. Automatically extracting subject information from maps and images, such as identifying vegetation in a map or a particular object in an image, would allow such documents to be automatically classified within, for example, the LC Classification system; such capabilities are ongoing research issues [25].

[6]http://potomac.ncsl.nist.gov/TREC/

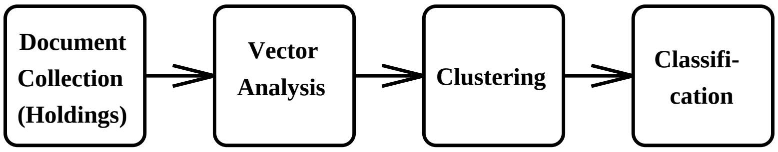

Figure 7: Metadata Extraction

4.3 High-Level Metadata Servers

Once the metadata at each site has been compiled, it needs to be distributed over the network according to the intended storage and retrieval architecture. As previously stated, the high-level metadata needs to be widely distributed and replicated, while the mid-level metadata is very selectively distributed. Because of these storage differences, different distribution protocols are more efficient for the different metadata levels. All high-level metadata is sent to each high-level server.

We briefly compared the distribution of the high-level metadata to the distribution of USENET news via NNTP in Section 2.3. NNTP uses a flooding protocol which has proved to be very robust and efficient for widely distributed, massively replicated data. In fact, rather than create a completely new protocol and transport system, we plan to distribute the high-level metadata in Pharos by using a new newsgroup hierarchy. This can be accomplished, for example, by having each source send out their high-level metadata as a news article. There are several advantages to this approach. First of all, it fulfills the basic design criteria of massive metadata replication. Second, NNTP provides a fast, reliable, and reasonably efficient mechanism of developing a prototype system. Third, this approach poses little or no degradation to the news system because news servers not wishing to store this information may selectively not receive a 'feed' for it, just as they can turn off reception of any other newsgroup hierarchy. Fourth, USENET provides a hierarchical naming convention among the newsgroups, which could be exploited by the source metadata design. For example, collections could be characterized based on theme, such as dl.medicine for MEDLINE, and dl.physics and dl.computer-science for INSPEC, so that servers within specific organizations can retrieve metadata tailored to their interests. Finally, those sites running a news server within their local area are prime candidates for high-level server locations.

A potential problem with this type of posting is that anyone could post articles with bogus metadata about a given source, such as the Library of Congress or the National Institutes of Health. This problem can be avoided by using moderated newsgroups and a standard electronic signature. When a source registers itself, it provides either a public encryption key or a place on the network where it can be found. Each posting is sent to an automated moderator that checks the signature against the source's key for verification before posting it.

Another consideration, as with the mid-level metadata, is the amount of information that can be handled. NNTP traffic generally generates approximately 400 MB per day or more. By cutting off unwanted newsgroups, news servers can substantially reduce the traffic coming to them. Recall that the total size of all the high-level metadata combined is on the order of several gigabytes. We imagine that most sources' statistical compositions of documents probably do not change dramatically over a short period of time. Therefore weekly or monthly updates are sufficient to keep high-level servers up-to-date to the level needed to assist users in the high-level, rough filtering of sources. We anticipate that this entire news hierarchy would not generate more than a small fraction of the normal NNTP traffic.

4.4 Mid-Level Metadata Servers

Mid-level metadata is not massively replicated, but rather stored on one or perhaps a few mirror sites. As such, a flood protocol is neither necessary nor efficient. Instead, a point-to-point metadata distribution system is more appropriate. Each source registers at each mid-level server for which the source has relevant metadata, telling the server how often to update the source's metadata. The mid-level servers in Pharos are responsible for keeping their metadata up-to-date. This approach minimizes network traffic and guarantees that the mid-level servers are ready to receive the metadata before it is sent.

Harvest [5] provides a suitable transport mechanism for distributing and storing mid-level metadata in Pharos. Harvest is a generalized system for automatically indexing documents within a source and distributing the metadata. In Harvest terminology, sources of information are called *providers*. Index metadata is extracted by *gatherers*. The gatherer is broken down into two components, one that extracts the metadata, and the other, *gatherd* (gather daemon), that handles the gatherer-side communication necessary for distributing it. The structured indexing information that the gatherer collects is represented as a list of attribute-value pairs using the *Summary Object Interchange Format (SOIF)*. This metadata is then served to *brokers*, who collect it via *collectors* (which communicate with a gatherer's gatherd) and provide a query interface to the indexes. Brokers can collect metadata either from gatherers or from other brokers, and thus lend themselves to hierarchical metadata propagation.

Although we must perform our own metadata extraction (or *gathering*), incorporating the results into SOIF records is straightforward. Each source constructs a SOIF record for each taxonomy sub-tree used by a mid-level server. Each source builds a standard Harvest SOIF database and runs gatherd. Each mid-level server runs a standard broker that collects the appropriate records from each source. This collection is efficient in Harvest because of the broker's ability to use structured queries that can specify boolean combinations of attribute/value pairs. For example, a broker would retrieve a record by specifying `'(Tax_ID : Subject) AND (Top_Node : Physics)'`. Finally, the brokers running on the servers are designed not only to collect the metadata, but also to handle sophisticated queries. Therefore, this software already facilitates the communication between the UI and mid-level servers.

There are several benefits of using Harvest for distributing and storing mid-level metadata. First, Harvest uses very efficient network protocols and compression for exchanging metadata records. Second, it allows us to place several logical mid-level servers on the same physical server; we need only change the query between the server's broker and the sources so that multiple SOIF records are transferred. Additionally, Harvest includes metadata timestamps and automatic server updates for expired information. Each source is able to set its own timestamps and thus regulate its own update frequency. Further, Harvest's ability to chain brokers allows us to incorporate a more sophisticated mid-level network model than requiring that each source directly communicate with each server. Instead, we can arrange the sources in a hierarchical network structure by placing intermediate brokers between the sources and the servers.[7] Finally, Harvest includes both a replication and caching system. The replication allows us to automate the process of mirroring the mid-level servers. Moreover, the caching allows us to take advantage of possible speed-ups by grouping together similar sources as pre-packaged sets within the UI; such query results might stay in a server's cache.

5 Conclusion

Pharos is a distributed architecture for locating diverse sources of information on the Internet. Such architectures must scale well in terms of information gathering with the increasing diversity of data, the dispersal of information among a growing user base, and the visualization of results with the growing data volume [4]. Pharos scales in all of these aspects. The use of a hierarchical metadata structure greatly enhances scalability. The Pharos distribution scheme also enhances scalability: each user begins with the metadata available at a local high-level server. Furthermore, since the mid-level metadata queries are distributed over a number of topic-based servers, there is little or no network bottleneck. Each source determines which taxonomies most appropriately fit its data domain. Diverse information is more easily classified as the number of taxonomies grow, including, for example, image and sound domains. Furthermore, each source determines when to update its own metadata. By collecting several taxonomy independent factors such as network parameters and the size

[7] In this sense, the mid-level servers can be viewed as the point between physically structuring the metadata based on network topology and physically structuring it based on information topology.

of a collection, Pharos allows users to broaden their search criteria beyond those allowed by term-matching techniques. Moreover, we generalize the linear rankings of sources provided in standard vector-based text retrieval techniques, allowing users greater freedom in visualizing and selecting criteria of their preferred sources. Finally, users can use relevance feedback techniques as a result of the iterative query methodology.

Pharos is extensible in that new taxonomies can be introduced as needed, either as part of an accepted Pharos standard such as the LC Classification, or as part of an independent group which defines its own taxonomy within which to classify documents. As the number of sources and/or taxonomies grow, both the high-level and mid-level metadata may become larger than practical. Therefore, we are looking at several extensions which can accommodate this growth. First, for the high-level metadata, we can allow several sources to be described in the same record. Although the metadata structure is not directly dependent on the number of sources, certain attributes such as network parameters will have to be either adjusted or left out of multi-source records. Another potential high-level space savings can be gained through allowing arbitrary groupings of sources. For example, a source primarily dealing with medical information can distribute its metadata within a 'medical' group, and those servers not interested in such information will simply not load it. For the mid-level metadata, we can split the mid-level into several levels as needed. If done carefully, each split could accommodate several orders of magnitude more sources.

We have begun implementing Pharos by classifying the ADL holdings within a geographical taxonomy. We believe that we will be able to do the same with other geographically based sources. We are also beginning to work with subject and temporal taxonomies. This work will need to be evaluated both by comparison with other systems and with human-based testing. Comparisons with subject taxonomies is more directly related to systems like *gGlOSS* and Content Routing, and we currently know of no equivalent systems for comparison with the other taxonomies discussed. Another important evaluation method is how effective the system is in leading users to what they consider good sources, ones that have relatively more documents they consider relevant than other sources. A human-based evaluation of our system will depend on empirical results based on asking users to complete specific tasks and measuring both their actual success rates and their own perceived success rates.

6 Acknowledgements

The authors would like to thank Dante De Lucia of the University of Southern California for his suggestions about networking; Linda Hill of the University of Maryland for her suggestions regarding classification and text retrieval; and Jason Simpson of the Alexandria Digital Library Project for his assistance in generating the map images.

A Example Query Details

We searched for the topic of political music during 1967 in San Francisco, Shanghai, and Cairo. Each search engine requires a different format. Searches were performed on July 1, 1996. We describe each search separately.

AltaVista regularly re-indexes the full text of the 30 million documents it references, and follows links to discover new pages. A 'simple search' on AltaVista consisting simply of the string 'political music' resulted in "about 500000 matching" documents which contained either the word 'political' or the word 'music'. While higher ranking was given to documents with both words, this approach was not appropriate for AltaVista. The top ten ranked documents yielded three related to political music: one was in Dutch about a Dutch foundation for political music and the other two were about current music. Changing the search to ' "political music" ' (i.e. double quoted) resulted in a match of 73 documents. The top ten documents included the following: an article from the on-line version of Mother Jones magazine, which was almost relevant to the original query, entitled "Rock 'n' Revolution: Dave Marsh on 20 years of political music" (the article consisted of a list of 20 songs very briefly annotated); an order form for CD's or descriptions for current bands or soloists; a magazine review site which listed two magazines partially related to political music; a long legal document on Vietnamese taxation which mentioned 'discs for political music programmes' in a sub-sub-section about turnover tax rates; a university dissertation about a 1977 Swedish musical dramatical production group; and the same Dutch foundation which was referenced in the unquoted query results. Several of the links in the set of 73 documents were no longer available, and some links pointed to the same document or copies thereof. Furthermore, the precision of the document set was well below 50%, especially since many documents contained the string "political music" once, but did not actually deal with the topic, and also because many pages were random people's personal home pages. An interesting document from East Tennessee State University described their archives of Appalachia music and included a 'Suggested Reading List'. There was also a brief description of a book entitled *Nineteenth Century Romanticism in Music, 3/e.* While several links were relevant to trends in, and activities around, current political music, there was little in the way of any comprehensive, let alone historical, discussion. Finally, an 'advanced search' of AltaVista was performed using the following search string: '(political NEAR music) AND (196) AND ("San Francisco" OR Shanghai OR Cairo)'. This query resulted in 3 documents, none of which had any bearing on political music or the 1960's. AltaVista does not support subject-based searching.

Lycos regularly re-indexes the full text of the 50 million documents it references, and follows links to discover new pages. The search on Lycos was less successful than on AltaVista. The simple search for 'political music' (unquoted), gave the highest weight to pages with the greatest number of occurrences of either word; the highest ranking went to a page with the word music in it several times but no occurrence of the word political. There were four documents of the top ten which contained both words, two of which dealt with some type of political music. One was a modern American satire group and the other was a publisher's brief summary of a book about a political musician that did not directly mention the exact years and place of the musician's compositions (though it appeared to be early Twentieth Century in Western Europe). Double quoting the phrase, ' "political music" ', did not affect the search results. A 'customized' search which required both words to appear on the same document resulted in 12 documents. Other than the top 4 documents which were included in the top ten of the previous query, the remaining 8 had little if anything to do with political music. The subject category '`Entertainment & Leisure:Music:Genres`' did not have a particular class for political music. The genres '`Pop/Rock/Alternative`', '`Country & Folk`', and '`Other`' had no mention of the word 'political' except for a single link to one artist. No other genres seemed to be related to political music.

Yahoo is more oriented toward subject-based searching than the other two index sites. It does not follow links to discover new pages, but automatically follows announcement site listings for new web sites. Yahoo does not regularly check links to validate that URL's are up-to-date, and does not state the number of pages it has indexed. One relevant topic, '`Arts:Humanities:History:Music`', had nothing about political music of the 1960's, with the possible exception of a link to someone compiling a list of all songs played at every Grateful Dead concert. A search of this section for the word 'political' yielded no matches. While there was no 'Political' topic under '`Entertainment:Music:Genres`', there was a 'Folk' topic; none of the links from this page, however, were relevant. A search of the entire Music section for 'political' resulted in 11 links, none of which were related to the original query. The topic '`Arts:Humanities:History:American History:20th Century:1960s`' had 4 links. Three were not relevant, but one, entitled "Wild Bohemians – an archive of historical information about Bohemian movements in the US during the 20th century," led, through a series of links, to the WWW site of the official "Museum of the City of San Francisco". This site included a chronology of Rock music and political events in the 1960's, and a complete bibliography about the Haight-Ashbury district, with over 60 highly relevant works, including, for example, "San Francisco nights: the psychedelic music trip, 1965-1968". A global search of all of the Yahoo pages for 'political music' resulted in 38 hits which contained both 'political' and 'music'. Yahoo does not seem to be able to specify 'political music' as a single two-word term. Furthermore, a Yahoo query searches only a document's title and one or two line description; there is no method available for full text searching, as with the other two indexes. None of the 38 hits appeared to be relevant to the original query.

Period	Range (year)	Period	Range (year)	Period	Range (year)
1	$< -10^9$	2	$-10^9 \longleftrightarrow -10^7$	3	$-10^7 \longleftrightarrow -10^6$
4	$-10^6 \longleftrightarrow -10^5$	5	$-10^5 \longleftrightarrow -10^4$	6	$-10^4 \longleftrightarrow -1001$
7	$-1000 \longleftrightarrow 999$	8	$1000 \longleftrightarrow 1199$	9	$1200 \longleftrightarrow 1399$
10	$1400 \longleftrightarrow 1499$	11	$1500 \longleftrightarrow 1599$	12	$1600 \longleftrightarrow 1699$
13	$1700 \longleftrightarrow 1749$	14	$1750 \longleftrightarrow 1799$	15	$1800 \longleftrightarrow 1849$
16	$1850 \longleftrightarrow 1899$	17	$1900 \longleftrightarrow 1909$	18	$1910 \longleftrightarrow 1919$
19	$1920 \longleftrightarrow 1929$	20	$1930 \longleftrightarrow 1939$	21	$1940 \longleftrightarrow 1949$
22	$1950 \longleftrightarrow 1959$	23	$1960 \longleftrightarrow 1969$	24	$1970 \longleftrightarrow 1979$
25	$1980 \longleftrightarrow 1989$	26	$1990 \longleftrightarrow 1999$	27	$2000 \longleftrightarrow 2009$
28	$2010 \longleftrightarrow 2019$	29	$2020 \longleftrightarrow 2049$	30	$2050 \longleftrightarrow 2099$
31	$2100 \longleftrightarrow 2499$	32	$2500 \longleftrightarrow 2999$	33	≥ 3000

Table 4: Top-Level Temporal Information Hierarchy

B Temporal Information Hierarchy

Table 4 shows the top-level sub-domains of a temporal information hierarchy. This hierarchy has 33 top-level periods, which cover all past, present, and future time. Since the majority of existing documents have been written during or regarding the last few hundred years, these periods are given a much higher level of granularity than, for example, cosmological, geological, or anthropological periods. Such a partitioning of the time-line provides for a more equal distribution of the number of documents likely to be classified within each period.

References

[1] AltaVista, http://altavista.digital.com/, 1995.

[2] Andresen, D., et al., "The WWW Prototype of the Alexandria Digital Library," Proceedings of the International Symposium on Digital Libraries, Tsukuba, Japan, 1995.

[3] Berry, M.W. & Dumais, S., "Using Linear Algebra for Intelligent Information Retrieval," SIAM Review, December 1995; and http://www.cs.utk.edu/ library/TechReports/1994/ut-cs-94-270.ps.Z.

[4] Bowman, C.M., et al., "Scalable Internet Resource Discovery: Research Problems and Approaches," CACM, vol.37, (no.8):98-107, 114, Aug. 1994.

[5] Bowman, C.M., et al., "The Harvest Information Discovery and Access System," Proceedings of the Second International World-Wide Web Conference, pp. 763-771, Chicago, Illinois, October 1994; and http://harvest.cs.colorado.edu/.

[6] CACM, "Digital Libraries," Section on Digital Library Initiatives, CACM, Vol. 38, No. 4, pp. 57-64, April, 1995.

[7] Chen, H. & Lynch, K.J., "Automatic Construction of Networks of Concepts Characterizing Document Databases," IEEE Transactions on Systems, Man, and Cybernetics, Volume 22, Number 5, Pages 885-902, 1992; and http://ai.bpa.arizona.edu/papers/ieee91/ieee91.html.

[8] Chen, H. et al., "Generating, Integrating, and Activating Thesauri for Concept-Based Document Retrieval," IEEE Expert, Special Series on Artificial Intelligence in Text-Based Information Systems, Volume 8, Number 2, Pages 25-34, April, 1993.

[9] Chen, H. & Ng, T., "An Algorithmic Approach to Concept, Exploration in a Large Knowledge Network (Automatic Thesaurus Consultation): Symbolic Branch-and-bound Search vs. Connectionist Hopfield Net Activation," Journal of the American Society for Information Science, Volume 46, Number 5, Pages 348-369, June 1995; and http://ai.bpa.arizona.edu/papers/snnn92/snnn92.html.

[10] ConQuest Software, Inc., "ConQuest Guide for Developers" (Proprietary), Version 4.0, Columbia, MD, 1994.

[11] Crum, L., "University of Michigan Digital Library Project," CACM, Vol. 38, No. 4, April, 1995.

[12] Dumais, S., "Improving the retrieval of information from external sources," Behavior Research Methods, Instruments, & Computers, Vol 23, No. 2, pp. 229-236, 1991.

[13] Ernst, R. & Yovits, M., "Information Science as an Aid to Decision-Making," *Decision-Making: Creativity, Judgment [sic], and Systems*, ed. Brinkers, H., Ohio State University Press, 1972.

[14] Fox, E., et al., "Digital Libraries," CACM, Vol. 38, No. 4, April, 1995.

[15] Gravano, L. & García-Molina, H., "Generalizing *GlOSS* to Vector-Space Databases and Broker Hierarchies", Proceedings of the 21st VLDB Conference, Zurich, Switzerland, 1995.

[16] Guyton, J. & Schwartz, M., "Locating Nearby Copies of Replicated Internet Servers," Computer Communication Review, vol.25, No.4, pp. 288-298, ACM-SIGCOMM '95, Oct. 1995.

[17] Horton, M., "Standard for Interchange of USENET Messages," RFC850, IETF Network Working Group, 1983.

[18] Hull, D., "Improving Text Retrieval for the Routing Problem using Latent Semantic Indexing," *Proc. of the 17th ACM-SIGIR Conference*, pp. 282-291, 1994.

[19] Kantor, B. & Lapsley, P., "Network News Transfer Protocol: A Proposed Standard for the Stream-Based Transmission of News," RFC997, IETF Network Working Group, 1986.

[20] Larson, R., "Experiments in Automatic Library of Congress Classification," JASIS, vol.43, No.2, pp. 130-148, 1992.

[21] Library of Congress, *LC Classification Outline*, Fifth Edition, Library of Congress, Washington, D.C., 1986.

[22] Losee, R. & Haas, S., "Sublanguage Terms: Dictionaries, Usage, and Automatic Classification," JASIS, vol.46, No.7, pp. 519-529, 1995.

[23] Lycos, http://www.lycos.com/, 1995.

[24] NSF, "Research on Digital Libraries," Program Guideline NSF 93-141, NSF, September, 1993, and http://www.nsf.gov/ftp/CISE/program/nsf93141.txt.

[25] Richards, J., *Remote sensing digital image analysis: an introduction*, 2nd rev. and enlarged ed., Springer-Verlag, Berlin, 1993.

[26] Salton, G., *Automatic Text Processing*, Addison-Wesley Publishing Company, Inc., Reading, MA, 1989.

[27] Schatz, B., "Building the Interspace: The Illinois Digital Library Project," CACM, Vol. 38, No. 4, April, 1995.

[28] Sheldon, M., et al., "Content Routing for Distributed Information Servers", Proceedings of the 4th International Conference on Extending Database Technology, 1994.

[29] Shneiderman, B., "Designing the User Interface, Strategies for Effective Human-Computer Interaction", 2nd Ed., Addison-Wesley, 1992.

[30] The Stanford Digital Libraries Group, "The Stanford Digital Library Project," CACM, Vol. 38, No. 4, April, 1995.

[31] White, D., *Fundamentals of Decision Theory*, North-Holland Publishing Company, Amsterdam, 1976.

[32] Woodruff, A. & Plaunt, C., "GIPSY: Automated Geographic Indexing of Text Documents," JASIS, Vol. 45, No. 9, pp. 645-655, 1994.

[33] Yahoo FAQ, http://www.yahoo.com/docs/info/faq.html, 1995.