Architectures for QoS based Retrieval in Digital Libraries

J. Sairamesh, C. Nikolaou and S. Kapidakis
ICS-FORTH
ERCIM institute
Heraklion, Crete
Greece, GR 71110
ramesh, nikolau, sarantos @ics.forth.gr

1 Introduction

Digital Libraries will have a major influence on the design of future information systems. They will form the *cradle* from which future complex information technologies will emerge to provide "transparent" services to a variety of users. In Digital Libraries[1], many real-world *agents* will participate for various economic reasons. For example, publishers of books will place their collections in a digital format to reduce production costs and increase profit margins. Many other types of economic entities will flourish for reasons of lower costs and higher revenue. With the rapid advances in computer and networking technology, thousands of heterogeneous computers can be interconnected to provide a large collection of computing and communication resources. Such a collection will be used by Digital Libraries to house various information objects such as text, audio, video and image, and provide various information access services (qualities of service) to a wide variety of users.

We at ICS-FORTH (Institute for Computer Science, Greece) are involved in the architecture (system design and implementation), and economics of Digital Libraries[1]. We are a part of the ERCIM[2] Digital Libraries initiative called **SAMOS**, which is **G7** attested project. The SAMOS project aims at the development of a networked computer science technical report library in Europe. A digital library architecture will provide Internet access to a distributed, decentralised multi-format collection of documents and a multilingual interface. The SAMOS system will support a distributed digital library by storing and providing access to technical reports from the ERCIM labs.

The SAMOS system will be an open system. This means that other European institutions (university computer science departments, research organizations, industrial R&D departments) could be connected to SAMOS and have access to a large number of European technical report collections. Furthermore, a study concerning the connection of the SAMOS with the DIENST system[2], developed in US, is being conducted. This connection will make possible[3] the quering from Europe any US collection of technical reports connected with the Dienst

[1] Economics from a point of view of interactions between publishers, consumers and providers

[2] ERCIM stands for European Research Initiative on Informatics and Mathematics. URL: http://www-ercim.inria.fr

[3] Some of European institutes have functioning DIENST system running

system, and vice versa[4]. The future system will be developed on open standards such as OMG CORBA[3].

The main objectives of the SAMOS project are: to design, implement, and test a system prototype for networked access to a distributed multi-format (Text, Postscript, SGML, etc.) collection of technical reports contained in the research libraries of the ERCIM consortium, which includes the major IT research organizations in Europe. The collection will be managed by a set of interoperating servers distributed over the network.

These servers will manage three basic library services: (a) repositories of multi-format technical reports; (b) indexes of the technical reports collection and search engines for these indexes; and (c) multi-lingual interfaces to provide front-end services for browsing, searching, and accessing the collection.

From the user standpoint a report collection will consist of a unified space of uniquely identified reports, where each of them may be available in a variety of formats. Using publicly available WWW clients, users can search the collection, browse, read, download and print individual reports in any of the available formats.

What are the issues and considerations

We at ICS-FORTH are working on various issues concerning the design and development of large decentralized and autonomous Digital Library systems. We also view such large systems in an economic sense (commercial digital libraries only) so as to understand the issues of storage, server and network costs, and over-all costs of retrieval and presentation of multimedia objects. This also promotes accounting and billing for the usage of resources.

Our primary goal is to provide resource allocation mechanisms and distributed searching and retrieval based on QoS (quality of service), as requested by the user. The resources we consider are server processing, I/O and memory, and network resources. Along with the indexing and searching mechanisms, we are investigating architectures for managing resources in a large scale digital library for *optimized* retrieval and presentation in a user-preferred fashion. For example, when the search results come back, users have a choice to decide on the objects and their types (or formats) based on their preferences.

We are conducting our experiments and testing our ideas over a testbed of DIENST[2] servers. DIENST uses HTTP and WWW framework for searching and presentation. We are enhancing the DIENST system services in several ways. We first outline some of the issues in desigining digital library systems of scale, and then mention our work with DIENST[5].

In creating and maintaining a large distributed Digital Library system with many nodes on the Internet several problems arise:

- Limited Bandwidth and Reliability: Once the information objects have been identified (indexing and searching), then retrieval becomes a task of utilizing the network bandwidth carefully. This implies information

[4] An agreement, in this sense, has already been reached between SAMOS and the US consortium (NCSTRL) which has developed the DIENST system

[5] Our work is currently experimental with DIENST in order to test our ideas in an Internet based environment.

about the network state can utilized to decide whether all the objects or parts of the objects are worth retrieving in order to improve response time. This calls for *structured* document access, only the relevant parts of the document should be retrieved for better response time (SGML type documents). Like wise for unreliable networks the issues are: what do we know about the state of the network before we send the requests? Can we use system management information before routing requests, and then retrieve information.

- Server Load: In the near future, we expect Digital Libraries to contain mostly multimedia documents (or objects). These objects, while being accessed by users, will use up many local system resources (bandwidth and buffer) and a measure of the load is very useful to re-route requests to other Digital Library servers that contain the relevant objects. We expect that servers will *replicate* information objects to save on network costs and provide better QoS.

- Dynamic load (cost) and QoS based searching. We are interested in using the server load conditions, and if available network load conditions, in order to search, retrieve and present objects to the user, in a manner *prefered* by the user. The problem is to optimize resources for searching and retrieval. For example, based on the current server load or network load, a document can be retrieved at a high or a low quality (postscript file with all figures, HTML file with no figures or a simple Text file).

  Similarly, information about the server and network load can be used for retrieving video or image object types such as JPEG or MPEG or MPEG-II. With object *replication*, the problem becomes: Which copy of the objects and their formats that minimize the retrieval response time for a fixed QoS or maximize the QoS for a given response time limit.

- Distributed Searching Alternatives

  - *Selective broadcast*

    In the current DIENST system, each DIENST server keeps indexes related to its collection. A request from a user is broadcast to all the DIENST servers (known to the request site), and replies to the request are collected and presented in an HTML format. This is not *scalable* as the number of servers increases, and all the severs may not contain the documents related to the request.

    The problem is to create a better distributed indexing system, which sends requests only to the servers that contain the required documents or objects. For this, we are investigating various distributed indexing architectures. In essence we perform a **Selective Broadcast** when a search request is placed.

  - Better broadcasting with freshness of indexes. When new documents or new collections (publishers) come into existence, the collections and corresponding indexes have to be known dynamically so as keep the indexes fresh for future requests, and improve upon the quality of search.

What is novel and our Testbed

We have a testbed of 7 DIENST servers running on our local FDDI based network. We are using this mainly for experiments to implement and test various indexing and searching mechanisms, performance tools, and QoS-based (load and cost for various object formats) retrieval of information objects. We have extended DIENST to support a framework whereby costs of access to the server based on load can be added.

In Figure 1, there are several digital library sites. Each site has digital library server (such as DIENST or other servers in the future) which represent a domain (university, organization, publisher, etc). Within each site a name-server stores information such as current load , status and so on about each digital library within the site. Each DIENST server posts load information in the form of costs (or access prices) to the nameserver. Figure 1[6].

The novelty lies in the use of name services for various information retrieval options such as:

- QoS based information retrieval: Nameservices such as DNS (Internet Domain Naming System) for the storage of QoS information from DIENST servers. DNS keeps records about hosts, and we exploit this to store QoS information about them. Such information can be used by the DIENST servers when they send out search requests or by agents (Java scripts or CGI scripts) for optimized searching and retrieval. In essence we have provided *hooks* which can be exploited for better search and retrieval.
- Customized searching and agents. The use of agent-technology such as Java scripts and CGI scripts for user customized queries. Users can present their search profiles using the Web based forms. In our implementation, DIENST servers update the corresponding name-servers belonging to their site (as shown in figure 1). One can use agents on behalf of the digital library servers to update information at the nameservers. The updates carry information on the current load and services being offered, and the cost of access. We plan to extend this in the future to CORBA based agents, traders and brokers.

Mathematical modeling and log analysis: We are investigating various (including economic models[4]) to analyze searching and retrieval of multimedia documents of the DIENST system. The modeling and analysis helps in accounting for resource usage, QoS provisioning, and better searching mechanisms.

We show in figure 2, an example of the access costs at the servers (of our testbed of 7 DIENST servers). The access costs reflect the over-all load at each server. Costs increase as load increases. The costs (or prices) are computed using simple performance models (example queueing models). The x-axis of the figure is the time of day, and the y-axis is the cost as a function of load at the server. The plot shows the cost fluctuations over a period of one day at each of the seven servers.

[6] Updates will be done securely in the future

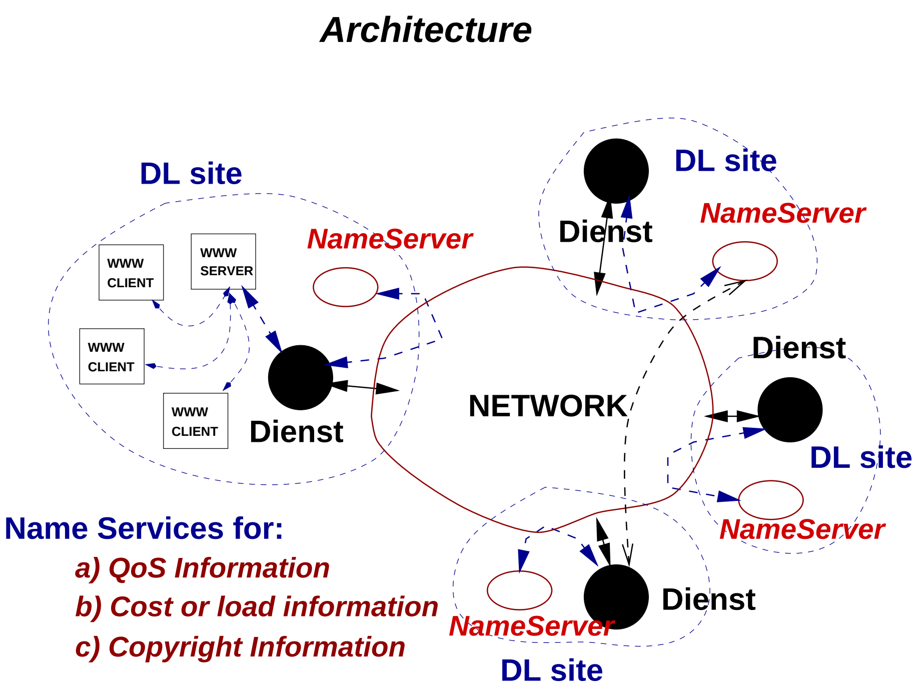

Figure 1: Architecture

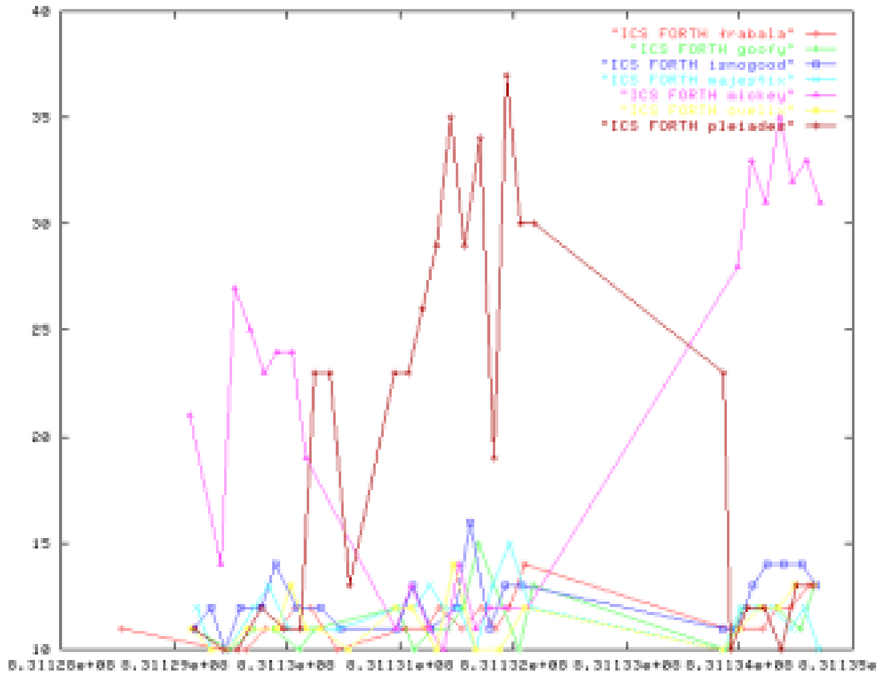

Figure 2: Server access cost, which reflects the server load

Acknowledgements

We like to thank Carl Lagoze and Jim Davis for fruitful discussions on DIENST and the evolving NCSTRL system, and various suggestions in improving the DIENST indexing and searching.

References

[1] Digital Libraries, special issue, *communications of the ACM*, 1995, 38 (4).

[2] C. Lagoze, E. Shaw, J. R. Davis and D. B. Krafft, *Dienst: Implementation Reference Manual*, TR95-1514, Cornell University, May 5th, 1995.

[3] Object Management Group. 1993. "The Common Object Request Broker: Architecture and Specification." http://www. acl.lanl.gov/sunrise/DistComp/Objects/corba.html

[4] D. F. Ferguson, C. Nikolau J. Sairamesh and Y. Yemini, "Economic Models for Allocating Resources in Computer Systems," in *Market based Control: A Paradigm for Distributed Resource Allocation*, ed. Scott Clearwater, World Scientific Publishing Co., 1995.