Prototype of the National High-Performance Software Exchange

Shirley Browne, Jack Dongarra, Stan Green,
Keith Moore, Tom Rowan, and Reed Wade
Computer Science Department
University of Tennessee*

Geoffrey Fox and Ken Hawick
Northeast Parallel Architectures Center
Syracuse University

December 2, 1994

Abstract

This report describes a short-term effort to construct a prototype for the National High-Performance Software Exchange (NHSE). The prototype demonstrates how the evolving National Information Infrastructure (NII) can be used to facilitate sharing of software and information among members of the High Performance Computing and Communications (HPCC) community. Shortcomings of current information searching and retrieval tools are pointed out, and recommendations are given for areas in need of further development. The hypertext home page for the NHSE is accessible at **http://www.netlib.org/nse/home.html**.

1 Introduction

Over the course of a two-month period, the NHSE developers team, consisting of researchers at the member institutions of the Center for Research on Parallel Computation, have undertaken the task of producing a prototype of the National High-Performance Software Exchange (NHSE). The NHSE is intended as an Internet-accessible resource which will facilitate the exchange of software and of information among research and computational scientists involved with

*Work on the National Software Exchange is supported by NASA under Grant No. NAG 5-2736.

High Performance Computing and Communications (HPCC). The target audiences for the NHSE include HPCC application and computer scientists, users of government supercomputer centers, and potential industrial users. The goals of the prototype are the following:

- Show the feasibility of distributing software and information via the National Information Infrastructure (NII).
- Demonstrate capabilities of existing software repositories.
- Demonstrate browsing and searching mechanisms.
- Identify limitations of current information services and areas needing further development.

2 Access via the National Information Infrastructure (NII)

To put together a prototype in the short time frame of two months, existing information search and retrieval tools were used. The prototype was designed to be accessed by the user from a HTML+ forms-capable World Wide Web (WWW) client, such as recent versions of NCSA Mosaic. The information and software were either already available on or were made available on a variety of network servers, including FTP, Gopher, WAIS, and HTTP.

Mosaic attempts to provide an interface to all the commonly used information retrieval services from a single application program. Mosaic retrieves an object from a network server and either displays it in a window, or if display is not possible or the user prefers, saves it to a disk file. A retrievable object may be stored in a file on a local or remote file system or may be constructed at retrieval time (for example, in response to a search query). An object is named by a string called a Uniform Resource Locator (URL). To retrieve an object, a user clicks on a highlighted anchor (which contains a URL) in a displayed document, specifies a URL to open, or fills in a form (causing Mosaic to generate a search URL). The URL contains the protocol to be used to contact the server, the server's DNS hostname, and any other information to be used by the server to locate or generate the object. The server typically returns either the object or an error message.

Mosaic provides native support for the FTP, Gopher, HTTP, and WAIS protocols. With WAIS, however, an alternative technique is to provide a WAIS-HTTP gateway which converts an HTTP search URL to a WAIS query, processes the query either locally or by contacting a remote WAIS server, and converts the search results to an HTML document.

Software may be made available through the NII by placing it on a file system served by FTP, Gopher, and/or HTTP servers. A software collection may be made searchable by WAIS-indexing descriptions of the software and running a WAIS server and/or by running an HTTP server and providing a WAIS-HTTP gateway. HTTP gateways may also be provided to other search engines, such as relational databases. With Mosaic, software the user retrieves may either be displayed in a window and then saved to disk, or saved directly to a disk file. Only one file may be retrieved at a time, however.

3 User Interface for Browsing, Searching, and Contributing

A user-friendly interface for the NHSE has been constructed in the form of a distributed HTML document. The root of this document is the NHSE home page (http://www.netlib.org/nse/home.html), which contains an outline from which users can reach all the collected HPCC information with a few clicks of the mouse. Although portions of the NHSE interface reside at different sites, most of the HTML pages making up the current prototype reside at the University of Tennessee. These pages have been indexed using WAIS, and a facility for doing keyword searching on the WAIS index has been provided. In the near future, indexing of the NHSE HTML pages will be done by a Harvest Broker which will use either Glimpse or freeWAIS as the indexing and search engine [5]. Because Harvest provides automatic support for gathering, summarizing, and indexing remote HTML pages and for keeping the resulting index up-to-date, a Broker running at the University of Tennessee will be able to index HTML pages from all the NHSE sites, either by having these sites run Gatherers locally or by doing remote gathering.

From the NHSE home page, the user can jump to a page listing HPCC programs and activities. These programs include US federal agencies, DOE, NSF, government, and university supercomputing centers, and various consortiums. The user may click on the name of any listed site to connect directly to a WWW server for that site.

Another page reachable from the NHSE home page is the Grand Challenges page, which includes subheadings for NSF Grand Challenge Application Groups, Oak Ridge National Lab Grand Challenge Projects, Argonne National Lab Grand Challenge Projects, University of Colorado Grand Challenge Projects, CRPC Grand Challenge Applications, and Grand Challenge conferences and meetings. The user can browse these pages and read descriptive material about the different projects, some containing contacts for further information.

Other headings on the NHSE home page include Software/Enabling Technologies, Information Databases, Commercial Vendor Catalog, Publications, and Related Network Resources. Software/Enabling Technologies and Information Databases are discussed below. The Commercial Vendor Catalog page contains an on-going survey of high performance computing systems, as well as pointers to home pages for some commercial hardware and software vendors, and to home pages for user groups for various products. A form is provided which vendors can fill out and submit if they wish to be listed. The Publications page contains pointers to journal tables of contents and electronic journals, as well as pointers to WWW and gopher servers for professional societies. The Related Resources page has subheadings for other HPCC-related WWW pages, electronic newsgroups and discussion lists, and resources for NHSE developers.

The NHSE invites contributions from users. The NHSE home page, as well as a number of other pages, contain an anchor which the user may click to obtain a fill-out form for making a contribution. The contributor fills in an email address, a URL for the contribution, and a title. Submitting the form causes an email message to be sent to the NHSE maintainers, who review the contribution before adding it to the appropriate NHSE page, which may be suggested by the contributor. The contributed material is maintained at the contributor's site – only a pointer in the form of a URL is included on the relevant NHSE page. The email address of the contributor is kept confidential, although of course the contributor may choose to make such contact information available on the contribution itself.

4 Software/Enabling Technologies

For the short-term prototype, the NHSE developers have provided a hypertext list of available repositories of mathematical and HPCC software. The user may browse this list, reading the descriptive material, and with a click of the mouse, connect directly to a repository site. Another list describes individual software packages and programs relevant to HPCC, some residing in a repository and some not.

4.1 The Netlib Repository

One of the software repositories is the Netlib Repository, which has been in existence since 1985 and which typically receives several thousand requests a day [6]. Netlib is a moderated collection which contains several packages widely recognized as being of high quality, such as EISPACK, LINPACK, FFTPACK, and LAPACK, as well as some newer, lesser known packages. Netlib is a replicated and distributed repository, with sites in Tennessee, New Jersey, Norway, England, Germany, Taiwan, and Australia. Each software package in Netlib has a master site, and all the other sites having a copy of that package mirror the master copy.

The original user interface for Netlib was an email interface. Although many users still access Netlib via email, some of the Netlib sites have begun offering a variety of other interfaces as well, including anonymous FTP, Gopher, and

HTTP. For example, through HTTP, the Netlib Repository can be browsed and searched by WWW clients such as Mosaic.

Because Netlib is intended to be a library, rather than just a warehouse, information about the software is recorded in template-type files called "index" files. These index files may be searched, or they may be used as input to index-building tools. To further aid searching, Netlib uses the GAMS classification scheme, discussed further below. The Netlib email interface provides the *find* command, which uses Unix grep to perform a keyword search of all the index files. The Netlib WWW server, *www.netlib.org*, provides a field-specific search interface to forms-capable WWW clients such as Mosaic. With this interface, the user can, for example, search for software by a certain author, containing one or more keywords in its abstract, and/or classified under one or more GAMS classifications.

The Netlib Repository contains a number of software packages specifically designed for high performance computing. LAPACK (Linear Algebra PACKage) provides routines for solutions of systems of simultaneous linear equations, for least squares solutions of linear systems of equations, and for solutions of eigenvalue and singular value problems. LAPACK has been designed to run efficiently on shared-memory vector and parallel processors. To minimize data movement between different levels in hierarchical memory, LAPACK reorganizes algorithms to use block matrix operations. The block matrix operations are implemented in the Level 3 BLAS (Basic Linear Algebra Subroutines), which can be optimized for each architecture. ScaLAPACK, which is currently under development, will extend LAPACK to run scalably on MIMD distributed memory computers. Like LAPACK, the ScaLAPACK routines are based on block-partitioned algorithms. The fundamental building block for ScaLAPACK are distributed memory versions of the BLAS, plus a set of Basic Linear Algebra Communication Subprograms (BLACS) for communication tasks that arise frequently in parallel linear algebra computations.

A number of parallel programming environments are available from Netlib, including P4, PICL, PRESTO, and PVM. These packages allow the programmer to write portable parallel programs.

4.2 The GAMS Virtual Repository

The GAMS (Guide to Available Mathematical Software) repository at NIST (the National Institute of Standards and Technology) provides centralized access to the software modules that it catalogs [4]. Rather than operating a physical repository of its own, GAMS provides transparent access to multiple repositories operated by others. GAMS currently indexes four software repositories. Three of these repositories are maintained for use by NIST staff and contain proprietary software. The fourth is Netlib, discussed above. Source code is not available for the proprietary software, but documentation and example programs often are.

All software modules cataloged in GAMS have been assigned one or more problem classifications from the GAMS classification scheme, which is a tree-structured taxonomy of mathematical and statistical problems [3]. The user can use the taxonomy as a decision tree, or can enter keywords which are mapped to problem classes. In addition, users may search by package or module name.

4.3 ASSET, CARDS, DSRS, ELSA, and AdaIC

ASSET stands for Asset Source for Software Engineering Technology. ASSET maintains the Worldwide Software Resources Discovery (WSRD) Library, which is a software reuse library and reuse information exchange available to software developers in government, industry, and education. Most of the reusable assets in the library are available for public distribution, although a few assets have not yet been authorized for distribution beyond government agencies and their contractors. An approved account is required to download assets, which are however free of charge. ASSET interoperates with CARDS (Comprehensive Approach for Reusable Defense Software) and DSRS (Defense Software Repository System), AdaIC (Ada Information Clearinghouse), and ELSA (Electronic Library Services and Applications) to allow users to retrieve components from these repositories as well. ASSET has also begun including references to commercial software and documents at no cost to vendors wishing to include references to their products and services.

4.4 COSMIC

NASA's COmputer Software Management and Information Center (COSMIC) has been located at the University of Georgia since its beginning in 1966. COSMIC contains over 1,200 computer programs that were originally developed by NASA and its contractors for the U.S. space program. Software is available for a number of areas of interest including: artificial intelligence, computational fluid dynamics, finite element structural analysis, scientific visualization, thermal and fluid flow analysis, and many more. Programs are priced on a cost-recovery basis and usually include source code. U.S. educational institutions are eligible for a substantial discount.

4.5 Other Repositories

Other software collections listed include Parallel Software and Tools at Argonne National Lab, Cornell Theory Center Parallel Programming Tools, the Software Development Group collection at NCSA, Paralib at CalTech, Softlib at Rice University, and StatLib at Carnegie-Mellon University. These collections are not as comprehensive as those described above, and they are not searchable, but they provide lists of software packages that can be browsed.

4.6 Software Packages

Software packages relevant to HPCC have been categorized into application-level software, parallel programming environments, parallel programming languages, and visualization software. There seems to have been little work done to compare and evaluate what software is available. There is much to choose from, but the user has little guidance in choosing and may need to resort to trying out the various packages. Some kind of peer review system would be helpful here.

5 Information Databases

Members of the HPCC community need access to other information in addition to software. Some kinds of information needed include conference and meeting announcements, performance data, whitepages directories, and computing literature.

5.1 Conferences Database

NHSE users need to know about conferences and meetings on high performance computing and in their specialty areas. Conference and meeting announcements are printed in journals and sent out by email to mailing lists, but a searchable on-line database of conferences would be very useful for immediate reference. The NHSE prototype has a forms interface to a conference database that is part of the Netlib Repository at UT/ORNL. The interface allows both searching and submission. A user may search by keyword, dates, and/or location.

5.2 Glossaries and Cross Referencing

In addition to the traditional databases of information, we have developed the new concept of a hypertextual glossary or dictionary as a navigation and cross referencing aid to the information and software available under the NHSE. The primary deliverable glossary is a series of hypertextual definitions of HPCC keywords and acronymns [11]. The HTML system allows a much richer context to be added to definitions than is possible in a traditional flat text representation. For example, one might apply a seach on the glossary for the term "HPCC", which would yield a definition, as well as cross references and links to sites elsewhere on the World Wide Web pertaining to HPCC. The HPCC National Coordination Office in the USA is such an example.

The glossary mechanism allows technology developers and practitioners to provide starting points and help information to those just entering the field and to essentially transfer the most difficult aspect of many technologies - the jargon. In addition to the prototype glossary on HPCC, available as a downloadable package from the NHSE, we are also developing glossaries on subtopics of special interest such as the High Performance Fortran programming language. Other specialised information glossaries are being developed for commercial customers through the InfoMall technology transfer program.

The HPCC glossary was first initiated as an HTML document with hand written entries containing embedded HTML tags. This approach is prone to error in human data entry, and it is cumbersome to insert new entries. To partially solve these difficulties we developed a scripting tool which parses entries, flags incorrectly formatted entries, and makes checklists of internal, external and duplicated entries. However, as the glossary has grown to in excess of 600 definitions, this approach is no longer scalable, since it becomes increasingly difficult for a human being to remember all the possible cross references when writing a new entry. We are therefore currently developing textual analysis tools to aid the entry writer in cross referencing a new entry. We are using hypertext technology for this in the form of client data entry forms that invoke word stemming and other analysis programs on the server side. The word stemming process is required so that duplicate cross references are not generated for an entry which might contain "cache", "caches", "cached", "caching", for example. At present our textual analysis system requires storage and comparison with a growing dictionary of English word and endings. This tool may in fact justify the use of a parallel search engine itself.

5.3 Performance Database

To make informed decisions about what computers to purchase and/or use for their applications, computational scientist need access to benchmark performance data. The process of gathering, archiving, and distributing computer benchmark data has typically been carried out by various users and vendors with little coordination. The NHSE prototype provides access to a publicly-available centralized database of performance data for all ranges of machines from supercomputers to personal computers [2]. The Linpack benchmark, Perfect benchmark, and Genesis benchmarks are included. A number of benchmark reports and a benchmarking bibliography are also available.

5.4 Whitepages Database

NHSE users need to be able to locate and contact other members of the HPCC community. A user may remember a piece of information about someone, such as that person's name, company, or address, and need to retrieve more complete information or an email address. Another possibility is that the user wishes to locate other people with similar research interests and having expertise in a particular area.

The NHSE prototype has an on-line search facility for the NANET (Numerical Analysis NET) Whitepages, which is a database of over 9000 members of the numerical analysis community. This database consists of entries from the SIAM membership list and those contributed by users. Both searching and joining are possible via the NANET WWW interface.

5.5 Literature Database

Users need easy access to searchable databases of recent literature such as newsletters, technical reports, and papers. As a simple first step, the NHSE developers have provided full-text WAIS indices of the University of Tennessee Technical Reports, the Numerical Analysis Digest (an electronic newsletter), and the LAPACK Working Notes. The user may perform a keyword search on any of these databases and have a list of titles returned. This search interface is primitive, however, as it does not allow field-specific searching or choice of result set format. More result set information, such as title, author, and a short abstract, would aid the user in deciding whether an item is relevant without viewing it in its entirety.

Netlib is participating in the BibNet Bibliography Network Project by mirroring the BibNet files from their home site at the University of Utah. BibNet is a public-domain bibliography database in BibTeX format. The BibNet project is intended to allow scientists to provide complete and updated information on their own work, find information about ongoing research of others, and maintain pointers to electronically accessible material. Contributions are currently solicited from the numerical analysis and scientific computing communities. A searchable index of titles and abstracts that includes the BibNet database is also available on the NHSE Information Databases page.

6 Roadmaps to HPCC Technology and Applications

Navigation aids to HPCC information are a valuable tool both for the experienced HPCC practitioner and the newcomer to HPCC technologies. We have developed a number of roadmap packages to provide contextual guides to selected HPCC base technology areas as well as application areas. For example, a hypertextual package describing application activities within the CRPC is available (http://www.npac.syr.edu/crpc/). This package describes and links to the Grand Challenge applications activities as well as selected other applications level activities ongoing by CRPC and its collaborators. This package is also accesable from the "Grand Challenges" section of the NHSE.

We have developed a roadmap integrated information package for applications of the High Performance Fortran (HPF) language (http://www.npac.syr.edu:80/hpfa/). HPF is an important HPCC software technology that originated within CRPC from collaborative efforts between Rice and Syracuse Universities. To encourage the uptake and widespread use of HPF, both by commercial compiler and system vendors as well as users, our

HPF Applications package focuses on HPF education and training material as well as HPF code examples and kernels. We have also provided a great deal of contextual linkage between the contributed codes and documents, locally and elsewhere on the Web, as well as many custom written for the package. This package has over forty short application code kernals illustrating how features of the HPF language may be used effectively.

We have broadened our efforts so as to build a prototype information integration package, or "roadmap", to all HPCC technologies and applications. Our prototype is accessible from the NHSE Home Page and also as http://www.npac.syr.edu/roadmap/. The roadmap attempts to link industrial market areas, application areas typical to industry and commerce, and academic areas to the basic HPCC software technologies.

We have also developed a prototype survey of HPCC hardware systems and vendors past and present. This survey, which draws on the experiences of HPCC practitioners from both sides of the Atlantic, contains entries on over 60 vendor or developer organisations who are associated with over 100 different HPCC systems. Our system is cross referenced to the Performance Database information, described elsewhere in this document. In addition to providing a catalog of current vendor systems, the survey serves as a textbook for some of the technological ideas developed during the history of parallel and high performance computing Links are also integrated in with the entries that point to other relevant information on the Web about HPC systems, and in particular to those currrent vendors who have web servers of their own. The prototype survey is available under the "Commercial Vendor Catalog" section of the NHSE.

At present most of our information integration packages are constructed manually drawing on the editing skills and domain expertise of a human writer. Manual construction is prone to typing errors in the HTML structure and is a very labor intensive way of integrating information and building hypertext links. We are investigating the use of online databases of documents on the Web, as well as databases of cross references, so that easy-to-use tools can be built to aid the editor of such packages. Our investigation draws on other HPCC technologies, such as large interactive databases and natural language processing, and has great promise for aiding and enabling a human editor to work more effectively and accurately, if not actually automating this process.

7 Technology Transfer to Industry and Government

Many of the ideas and concepts explored during the development of the NHSE prototype are applicable to building Information and Technology Exchanges for other organizations and user communities. We use the term Technology Integration Systems (TIS) to describe a generalized hypertextual information and technology repository with a carefully chosen navigational layout, structured to assist technology transfer between developers and users. Through a contractual arrangement between the CRPC's InfoMall technology transfer program and the US Air Force Material Command, we have built such a prototype TIS for Rome Laboratory, one of the Air Force's "Superlabs" [9]. This system was constructed to demonstrate how the mechanism's embodied in the NHSE could be used to assist technology transfer within an Air Force context. The effort focused on technologies of special interest to Rome Lab, either in its role as a technology developer, or as a technology consumer. As well as the relatively mechanistic process of placing non-classified information on line as hypertext, we undertook the process of integrating this technology information in a world wide context. This was undertaken by seeking advice from a range of acknowledged world class experts, including many senior researchers of the CRPC, in the form of consultant reports on selected technologies. The three specialized topics chosen for the phase one concept demonstration were software engineering, distributed computing, and broadband networking.

The prototype server contains:

- approximately fifty technology data sheets with contextual integration in the form of hypertextual links to related technologies and applications at world wide sites,
- over twenty consultant reports covering the HPCC technology areas described above,
- various information navigation aids to the technologies, applications, and the US Air Force bases in the form of summary pages for selected activities at each site.
- various hypertext links to information assets within the NHSE, including terminology glossaries, product sheets, and acronymn expansion lists.

The Rome Lab prototype can be accessed through the InfoMall server package at **http://www.npac.syr.edu/infomall/** or directly at **http://king.syr.edu:2001/**. The project to build this server and the process behind it, as well as a summary of the results, are described in a separate technical note [9].

Rome Laboratory personnel have uptaken many of the ideas from our prototype and are placing a considerable body of more detailed information on an internal server. In addition we are helping Rome transfer this technology to other US Air Force sites. Furthermore, in view of the favourable response to this prototype, other industrial and business organizations are seeking our assistance in building TIS packages.

8 Areas Needing Development

The current situation with software and information retrieval on the Internet presents a number of difficulties that need to be overcome if the NHSE is to become a scalable, reliable, efficient, and interoperable distributed source of information. These difficulties are discussed below, along with suggestions for overcoming them.

8.1 Universal Naming Scheme

Currently location-dependent filenames are used to name objects. Multiple copies of objects often exist, but outside of the distributed Netlib Repository, different sites may have different filenames for the same object. To complicate matters further, the same filename is often used for different objects or for different versions of the same object.

Current search facilities return the locations of objects, usually in the form of URLs. As a result, multiple hits may be returned for the same object. Another disadvantage is that the searchable indices have to be updated every time a copy of an object is moved, and such relocation is likely to happen much more often than changes to the object or to its description.

Naming objects by their locations presents a problem for caching, because it is difficult to tell whether a desired object is in the cache. Looking up whether the URL is cached does not suffice, because the contents of the location may have changed. Hence, clients or cache servers are forced to try to determine if the contents of the location have changed since the last access. Caching by location also wastes the opportunity to take advantage of multiple copies of objects. The Xnetlib client demonstrates a way of doing local caching by putting local copies of index and data files on the user's local file system, and these cached copies may be shared on a site-wide basis. To do such caching outside of Netlib, however, will require universal names.

Although it uses consistent filenames across different Netlib sites, Netlib still has the problem that an object's name is tied to its location. If an object is moved to a new location on the file system, it gets a new name, and the old name becomes stale. Furthermore, Netlib files may be updated in place, and although such updates are automatically propagated to all mirroring sites, the user cannot tell from the name, or from any other readily accessible information, that the object has changed.

Hence, there is a need for a name that remains constant for the lifetime of an object, regardless of its location. The Uniform Resource Name (URN) has been proposed by the IETF Uniform Resource Identifier (URI) Working Group [13]. The Netlib Development Group has begun implementing a specialized type of universal name called a Location Independent Filename (LIFN) as part of the Bulk File Distribution (BFD) project. A LIFN is assigned to a particular sequence of bytes, and once the assignment has been made, the same LIFN cannot subsequently be used to name any other sequence of bytes. The space of LIFNs is subdivided among several *publishers*, or *naming authorities*, who are responsible for ensuring the uniqueness of LIFNs within their portions of the LIFN-space. Resolving a LIFN involves finding an appropriate LIFN-to-location server and then contacting that server to obtain a list of locations for the LIFN. The prototype BFD implementation includes LIFN-to-location server code, as well as a WWW client library for resolving LIFNs. LIFNs are currently being assigned to Netlib files and are being integrated into the Netlib search interfaces. Use of LIFNs will allow Netlib to automatically mirror files from author-maintained sites, rather than handling updates from these sites manually. Future plans include include integration of authenticity and integrity checking into server-to-server and client-to-server protocols. Such checking will allow a network of trusted LIFN servers and file servers to be set up for a naming authority and will allow a client to authenticate such servers. More information about BFD is available at http://www.netlib.org/nse/bfd.

8.2 Indexing and Searching

Currently searching on the World-Wide Web is very ad hoc, and searches typically have poor precision and recall. Furthermore, the user has no way to evaluate the quality of his/her search. For example, if no items are found, it is not possible to tell whether no information exists or the search needs to be modified or expanded, and there are no reliable methods for restricting or expanding a search. There are also a multitude of different search services, with a great deal of overlap and using different protocols, and this situation makes searching redundant and expensive.

The NHSE needs a structure and guidelines for describing and classifying objects, especially software. Descriptions should be in the form of attribute/value templates, with standard definitions for the data elements. The IAFA software and document templates are a start, but rigorous data definitions for these templates are lacking. The Netlib index file format is another example of a software template, but rigorous definitions of data elements are again lacking. A candidate for a standard format which is both rigorously defined and extensible is the RIG Proposed Standard RPS-0002 (1994), A Uniform Data Model for Reuse Libraries (UDM), available from the Reuse Library Interoperability Group via AdaNET at 800-444-1458. The IETF URI Working Group has discussed the Uniform Resource Citation (URC) as a way of encapsulating meta-information about objects, but so far there has been no agreement on the format of URCs or on how they will be deployed and used.

Research in information science has shown that the quality of searching is vastly improved by the use of comprehensive, well-defined classification schemes. The Guide to Available Math Software (GAMS) classification scheme has demonstrated the effectiveness of classification for retrieving mathematical software [3]. An extension to GAMS, or the development of an additional classification scheme, is needed, however, for non-mathematical software that does not fit into any of the current GAMS classes. It would also be useful to have mappings between GAMS and existing document classification schemes such as the American Mathematical Society and Computing Reviews categories.

Because it is probably not feasible or even desirable to have a centralized authority for describing and classifying software and documents contributed to the NHSE, guidelines need to be established to assist publishers in accurately describing and classifying contributed materials. This classification activity could be facilitated by a hypertext help system, with pointers to data definition schema and classification examples. A similar hypertext system could assist users in formulating good queries and in using classification schemes.

Current indexers typically build an inverted index that includes every occurrence of every word (possibly excluding some common words) in the indexed material. The main drawback of inverted indices is their space requirements, which typically range from 50 to 300 percent of the size of the original text. Inverted indices also require exact spelling, and a misspelled word can cause a piece of information to be lost. At the cost of somewhat slower searching performance, the Glimpse indexing tool builds a much smaller index (typically 2 to 4 percent of the original text) and supports arbitrary approximate matching [12]. The material indexed by an indexing tool might be the title or filename only (e.g., Archie), the title plus keywords plus abstract (e.g., ALIWEB), or the full text of documents (e.g., WAIS). Indexers and search engines are needed that support attribute-specific indexing and searching. The NHSE prototype uses the freeWAIS code from CNIDR as the basis of most of its searchable indices. The current version of freeWAIS supports only free-format indexing, but an alternative version that supports attributes (freeWAIS-sf) by building a separate inverted index for each attribute is under development. A version of Glimpse that supports attributes is also under development. There also a number of commercial products available that support text searching or combined relational database and text searching.

Z39.50 (Version 2, 1992) is a US ANSI standard protocol for information retrieval [1]. Z39.50 was designed with the intent of supporting bibliographic database applications. The format of the data retrieved is not constrained by the protocol but is agreed to by the client (*origin* in Z39.50 terminology) and server (*target* in Z39.50 terminology). The MARC format and a search attribute set suitable for bibliographic data are registered within the current version of the standard. It is expected that as the protocol begins to be used by other communities and for other types of data, other attribute sets and record syntaxes will be developed. There is a mechanim that allows new record syntaxes to be registered and then referred to by well known identifiers. A number of library automation vendors are developing Z39.50 support for their products. A number of sites on the Internet have put up Z39.50 servers (a list of these sites is at http://ds.internic.net/z3950/dblist.txt). Thus, support for Z39.50 by the NHSE information infrastructure will allow interoperability with the growing number of organizations using this standard. The NSE may want to take the lead in defining and registering an attibute set and record syntax especially suited for searching software repositories.

A central searchable database for the NSE will not scale to many thousands of users, nor will replication of a single monolithic database. The alternative is to partition search records into subject area databases. The search problem then becomes one of locating the appropriate search server. The concept of centroids has been proposed and discussed by the IETF URI Working Group [14]. The centroid idea involves a hierarchical organization of search servers, and a server propagates upwards the set of keywords which it has indexed. Without some analysis, it is by no means certain that the centroid idea will scale, however.

The Harvest system provides a set of customizable tools for gathering information from diverse repositories, building topic-specific content indices, and widely replicating the indices [5]. A *Gatherer* collects indexing information from a service provider, while a *Broker* provides an indexed query interface to the gathered information. Harvest reduces server load and network traffic by using provider-site-resident indexing, content summaries instead of full text, and sharing of information between interconnected Gatherers and Brokers. It is proposed that a number of topic-specific Brokers be constructed. A distinguished Broker, called the Harvest Server Registry (HSR), which may be replicated, maintains information about all Harvest Gatherers and Brokers. It is suggested that the HSR be consulted when looking for an appropriate Broker to search. The Harvest system appears to offer a good framework for developing a search interface for the NSE. The Netlib Development Team is currently experimenting with Harvest, and a Harvest Broker that indexes a large number of NSE pages is accessible from the NSE home page.

8.3 Collection Management

The NSE will grow faster and contain more useful information if application teams have some degree of ownership of the software repositories and are motivated to contribute to them. A development team such as the group that produced this NSE prototype has neither the manpower nor the expertise to develop, catalog, and maintain high-quality software and documents in all the application areas. Although funding could conceivably be found to increase the manpower of the development group, scientists closest to the different application areas will be in the best position to produce useful software and documents. On the other hand, these scientists cannot be expected to document, refine, and thoroughly debug experimental software to the point where it is of sufficiently high quality to be generally useful and shareable, because they do not have software development expertise nor are they funded to do software development. Nor can they be expected to learn all the details of various classification schemes so as to accurately catalog their contributions. Thus, additional funding would be best spent on software development and cataloging experts who can work closely in teams with application scientists, either as regular staff, or in the case of smaller institutions or short-term projects, on a consulting basis.

Quality control will be needed, both for the software and documents themselves, and for the descriptive information and the cataloging and indexing process. Control is needed for software so that the user can have confidence in software obtained from the NSE, and for cataloging so that users can accurately evaluate the results of their searches. In its plan for the NSE, the Center for Research on Parallel Computation (CRPC) has proposed a review process for software similar to that used for articles submitted to a scholarly journal. The CRPC would provide or recruit area editors for the different software domains, such as linear algebra, numerical simulation, parallel compiler technologies, and visualization. A rating scheme for software quality would be defined, based on criteria such as stability, robustness, documentation quality, level of support, and transportability.

Although miminum submission requirements and quality control are essential, the software submission process must be structured in such a way that it does not place so heavy a burden on application developers as to discourage contributions.

Development of a comprehensive collection of Internet resources for the NSE will require the following:

1. A clear statement of the scope, purpose, and goals of the NSE.
2. Division of the overall scope into categories and sub-categories.
3. Assignment of categories and sub-categories to individuals who are aware of current developments in Internet resources in their respective areas.
4. Feedback from users and domain experts on the significance and value of resources.
5. Identification of focused, specialized, and well-maintained collections of information to serve as destinations of links.
6. Recognition for publishing in a moderated collection such as the NSE so that contributors and reviewers will find it worthwhile to put effort into improving the quality of resources.

The current representation of the NSE as an interlinked sub-web of HTML pages has a number of drawbacks, including maintenance difficulty and fixed organization and format. To overcome theses difficulties, the NSE contents should be reorganized as a display-independent database, with each entry in the database assigned one or more values from a faceted classification scheme (e.g., type of resource, subject area). The user should be able to select a default, pre-generated organization and format, or to generate a new organization on the fly. Portions of the database should be distributed and replicated at various sites. Distribution is necessary to permit local maintenance of their portions of the database by domain experts. Replication is necessary to achieve high availability and scalability. With distribution, the page the user sees might be constructed by querying servers at different sites for the current state of portions of the database. Location-independent naming of files (see section 8.1 will simplify the management of the distributed NSE database.

8.4 Software Engineering Model for Producing and Using Shared High Performance Software

Application scientists will often be reluctant or unable to create from scratch large-scale application programs that achieve the best possible performance on different advanced computer architectures. Using code from generic mathematical software libraries of from application-specific libraries may be useful for prototyping, but programs constructed from such codes may not achieve sufficiently high performance for large-scale problems. The reason for insufficient performance of generic library software on parallel architectures is that various operations, such as data mapping and partitioning, must be finely tuned to a particular architecture to achieve maximum performance.

An approach being taken to produce high-performance linear algebra software, while preserving transportability and reusability, is illustrated by the LAPACK and ScaLAPACK packages available from Netlib [7, 8]. The algorithms have been structured so as to carry out key matrix computations by means of calls to the Basic Linear Algebra Subroutines (BLAS). In addition, for ScaLAPACK, common parallel communication tasks are carried out by calls to the BLACS (Basic Linear Algebra Communication Subroutines). The BLAS and BLACS may then be implemented and finely tuned by vendors of advanced architecture machines. The result is a common application-level programming interface provided by the LAPACK and ScaLAPACK routines, but with the resulting application programs being transportable across different architectures that have implemented the underlying support routines. A similar approach of using lower-level parallel implementations of common computational and communication operations may work for other application domains, such as partial differential equations or combinatorial graph algorithms.

Work is needed to develop, evaluate, consolidate, and standardize system software that allows easier and more efficient use of parallel architectures. Such software should include notations for data mapping, parallel compilers, parallel programming enviroments, and visualization and debugging tools. Application-specific languages, such as FIDIL for computational fluid dynamics, will also be useful.

Although most available mathematical software is written in Fortran, some users may be more comfortable and productive with an object-oriented language such as C++. LAPACK++, available from Netlib, is an object-oriented C++ extension to the Fortran LAPACK library for numerical linear algebra. ScaLA-

PACK++, currently under development, is an object-oriented C++ library for implementing linear algebra computations on distributed memory parallel computers. LPARX, developed by researchers at the University of California at San Diego and available from Netlib, is a C++ class library that provides runtime support for dynamic, non-uniform scientific calculations running on MIMD distributed memory architectures.

Software engineering models for developing library software for high performance computing and for writing application programs that use this software should be developed, described, and illustrated with examples. The hypertext medium presents a good way of making this information available. Development of the software engineering models should be an ongoing process, with feedback from users guiding their development.

8.5 Whitepages Directory

A directory of NSE users should be provided with an adequate search interface and with the capability for a user to join, change his/her entry, and delete his/her entry. For security and scalability reasons, this directory might best be maintained in a distributed fashion, using for example the Whois++ protocol. Each site could maintain its own Whois++ server, and a server-to-server protocol could be used to propogate local records to a few large directories maintained by major repository sites such as Netlib. These top-level directories could be deployed on a geographical basis. A user with some idea of where an individual is located could then perform a less expensive search than querying the entire distributed database. There would still be the problem, however, of locating appropriate search servers if the user's query were based on other criteria, such as subject area or research interests.

The NSE may also want to consider participating in an existing research database such as the BEST North America Database, accessible at http://best.gdb.org/best.html. BEST North America collaborates with over 125 major research universities and other R&D organization in the United States and Canada. Its purpose is to provide a comprehensive directory of faculty researchers and scientists, and to connect academic and corporate researchers via the Internet and other electronic platforms. Services provided to participating institutions include technical editing and database management of raw data supplied by the institution, tools for electronic gathering of researcher profile data, and access to an institution's records as a separate database.

8.6 Access to Bibliographic Databases

NSE users need easy-to-use tools for carrying out comprehensive bibliographic searches in their research areas. On possibility would be for the NSE to maintain its own comprehensive bibliographic database. Populating and maintaining such a database would require a great deal of human effort, however, and would duplicate efforts of existing on-line bibliographic databases. An alternative approach would be to provide a hypertext forms interface to existing on-line databases. Most of these databases require an account and a password, but once the user had obtained these, the forms interface could facilitate querying the database and retrieving and manipulating the search results. The hypertext interface could also give information on how to obtain an account. Even with this second approach, newly published and unpublished papers might be kept in a supplemental database maintained by the NSE. This supplemental database would be populated by user contributions and could contain pointers to on-line versions of papers. The BibNet project, led by Stefano Foresti and Nelson H. F. Beebe of the University of Utah, is an on-line bibliography that is mirrored by the Netlib sites and which consists of user contributions of Bibtex files in the field of scientific computing. Contributors to BibNet may include pointers to on-line copies of papers in the form of URLs. BibNet could be used as the supplemental NSE bibliographic database if it were advertised and made accessible through the NSE WWW pages.

8.7 Replicated and Authenticated Information Servers

We have observed the following problems with network information servers:

1. Internet servers are frequently overloaded or unavailable for various reasons.

2. Servers currently provide no assurances of authenticity or integrity.

To deal with the above problems, a universal location-independent naming scheme needs to be coupled with file replication and authentication mechanisms. Transparent replication via a name-to-location lookup service will achieve fault tolerance and load balancing for end-user network services. File servers should be required to authenticate themselves before being allowed to replicate files and update the name-to-location mapping. A description associated with each location-independent filename should provide an MD5 or similar fingerprint for verifying the integrity of a retrieved file.

References

[1] *ANSI/NISO Z39.50-1992: Information Retrieval Application Service Definition and Protocol Specification for Open Systems Interconnection.* NISO Press, July 1992.

[2] M. W. Berry, J. J. Dongarra, and B. H. Larose. PDS: A performance database server. *Scientific Computing*, 1994. (to appear).

[3] R. F. Boisvert, S. E. Howe, and D. K. Kahaner. GAMS: A framework for the management of scientific software. *ACM Trans. Math. Softw.*, 11(4):313–355, Dec. 1985.

[4] R. F. Boisvert, J. L. Springmann, and M. L. Strawbridge. The GAMS virtual software repository. In *Proceedings of the Thirtieth Semi-Annual Meeting*, pages 68–72, Gaithersburg, MD, September 1992. Cray User Group, Five Point Editorial Services.

[5] C. M. Bowman, P. B. Danzig, D. R. Hardy, U. Manber, and M. F. Schwartz. Harvest: A scalable, customizable discovery and access system. Technical Report CU-CS-732-94, Department of Computer Science, University of Colorado-Boulder, Aug. 1994.

[6] S. Browne, J. Dongarra, S. Green, K. Moore, T. Rowan, and R. Wade. Netlib services and resources. Technical Report UT-CS-94-222, University of Tennessee Computer Science Department, Feb. 1994.

[7] J. Demmel, J. Dongarra, and W. Kahan. On designing portable high performance numerical libraries. Technical Report CS-91-141, University of Tennessee, July 1991.

[8] J. J. Dongarra and D. W. Walker. Software libraries for linear algebra computation on high-performance computers. Technical Report ORNL TM-12404, Oak Ridge National Laboratory, Aug. 1993.

[9] G. C. Fox, W. Furmanski, K. A. Hawick, D. M. Leskiw. Exploration of the InfoMall Concept. Technical Report SCCS-634, Northeast Parallel Architectures Center at Syracuse University, Aug. 1994.

[10] G. C. Fox and K. A. Hawick. A Roadmap to HPCC Technology and Applications, Draft Technical Report, Northeast Parallel Architectures Center at Syracuse University, Nov 1994.

[11] K. A. Hawick. High Performance Computing and Communications Glossary (release 1.3a), Technical Report SCCS-629, Northeast Parallel Architectures Center at Syracuse University, Jul. 1994.

[12] U. Manber and S. Wu. Glimpse: A tool to search through entire file systems. Technical Report TR 93-34, Department of Computer Science, University of Arizona, Oct. 1993.

[13] K. Sollins and L. Masinter. Requirements for Uniform Resource Names. Technical Report draft-sollins-urn-02.txt, IETF Internet Draft, Aug. 1994.

[14] C. Weider, J. Fullton, and S. Spero. Architecture of the Whois++ index service. Technical Report draft-ietf-wnils-whois-02.txt, IETF Internet Draft, Oct. 1993.