161. Anthony Skjellum, Lawrence Livermore National Laboratory, 7000 East Ave., L-316, P.O. Box 808 Livermore, CA 94551
162. Danny C. Sorensen, Department of Mathematical Sciences, Rice University, P.O. Box 1892, Houston, TX 77251
163. G. W. Stewart, Computer Science Department, University of Maryland, College Park, MD 20742
164. Paul N. Swartztrauber, National Center for Atmospheric Research, P.O. Box 3000, Boulder, CO 80307
165. Philippe Toint, Dept. of Mathematics, University of Namur, FUNDP, 61 rue de Bruxelles, B Namur, Belgium
166. Bernard Tourancheau, LIP ENS Lyon, 69364 Lyon cedex 07, France
167. Hank Van der Vorst, Dept. of Techn. Mathematics and Computer Science, Delft University of Technology, P.O. Box 358, NL-2600 AA Delft, The Netherlands
168. Charles Van Loan, Department of Computer Science, Cornell University, Ithaca, NY 14853
169. Jim M. Varah, Centre for Integrated Computer System Research, University of British Columbia, Office 2053-2324 Main Mall, Vancouver, British Columbia V6T 1Z1, Canada
170. Udaya B. Vempati, Dept. of Computer Science, University of Central Florida, Orlando, FL 32816-0562
171. Robert G. Voigt, I CASE, MS 132, NASA Langley Research Center, Hampton, VA 23665
172. Phuong Vu, Gray Research, Inc., 1900 Franz Rd., Houston, TX 77084
173–177. Reed C. Wicke, Department of Computer Science, Ayres Hall, University of Tennessee, Knoxville, TN 37996-1301
178. Daniel D. Winer, Department of Mathematical Sciences, O 154 Martin Hall, Clemson University, Clemson, SC 29631
179. Robert P. Waver 1555 Rockmont Circle, Boulder, CO 80303
180. Mary E. Wexler, Rice University, Department of Mathematical Sciences, P.O. Box 1892, Houston, TX 77251
181. Andrew B. White, Computing Division, Los Alamos National Laboratory, P.O. Box 1663, MS-265, Los Alamos, NM 87545
182. Margaret Wight, Bell Laboratories, 600 Mountain Avenue, Murray Hill, NJ 07974
183. David Young, University of Texas, Center for Numerical Analysis, RLM 13.150, Austin, TX 78731
184. Earl Zimmermann, Department of Computer Science, University of California, Santa Barbara, CA 93106
186–187. Office of Scientific & Technical Information, P.O. Box 62, Oak Ridge, TN 37831
140. Jesse Boone, Department of Computer Science, Ayres Hall, University of Tennessee, Knoxville, TN 37996-1301
141. Alex Rother, Department of Computer Science, Pennsylvania State University, University Park, PA 16802
142. Yuanchang Qi, IBM European Petroleum Application Center, P.O. Box 585, N-4040 Hafrsfjord, Norway
143. Giuseppe Radiati, IBM European Center for Scientific and Engineering Computing, via del Giorgione 159, I-00147 Roma, Italy
144. John K. Reid, Numerical Analysis Group, Central Computing Department, Atlas Centre, Rutherford Appleton Laboratory, Didcot, Oxon OX11 OQX, England
145. Werner C. Reinboldt, Department of Mathematics and Statistics, University of Pittsburgh, Pittsburgh, PA 15260
146. John R. Rice, Computer Science Department, Purdue University, W. Lafayette, IN 47907
147. Gary Rodrigue, Numerical Mathematics Group, Lawrence Livermore Laboratory, Livermore, CA 94550
148. Donal J.GP. Bode, Department of Computer Science, Duke University, Durham NC 27706
149. Edward Rothberg, Department of Computer Science, Stanford University, Stanford, CA 94305
150. Axel Ruhe, Dept. of Computer Science, Chalmers University of Technology, S-41296 Göteborg, Sweden
151. Joel Saltz, NASA Langley Research Center, Hampton, VA 23665
152. Ahmed H. Sameh, Center for Supercomputing R&D, 1384 W. Springfield Avenue, University of Illinois, Urbana, IL 61801
153. Michael Saunders, System Optimization Laboratory, Operations Research Department, Stanford University, Stanford, CA 94305
155. Martin H. Schultz, Department of Computer Science, Yale University, P.O. Box 2158 Yale Station, New Haven, CT 06520
156. David S. Scott, Intel Scientific Computers, 15201 N.W. Greenbrier Parkway, Beaverton, OR 97006
157. Lawrence F. Shampine, Mathematics Department, Southern Methodist University, Dallas, TX 75275
158. Andy Sherman, Department of Computer Science, Yale University, P.O. Box 2158 Yale Station, New Haven, CT 06520
159. Kermit Sigmon, Department of Mathematics, University of Florida, Gainesville, FL 32611
160. Horst Simon, Mill Stop 1045-1, NASA Ames Research Center, Moffett Field, CA 94035
119. John G Lewis, Boeing Computer Services, P.O. Box 24346, M/S 7L-21, Seattle, WA 98124-0346
120. Jing Li, IMM Inc., 2500 Park Wst Tower Che, 2500 City Wst Blvd., Houston, TX 77042-3020
121. Heather M Liddell, Center for Parallel Computing, Department of Computer Science and Statistics, Queen Mary College, University of London, Mile End Rd, London E1 4NS, England
122. Arno Liebmann, c/o EIH Rechenzentrum, Clausiusstr. 55, CH-8092 Zurich, Switzerland
123. Joseph Liu, Department of Computer Science, York University, 4700 Keele Street, North York, Ontario, Canada M3J 1P3
124. Robert F Lucas, Supercomputer Research Center, 17100 Science Drive, Bowie, MD 20715-4300
125. Franklin Luk, Department of Computer Science, Anns Eaton Building - #131, Rensselaer Polytechnic Institute, Troy, NY 12180-3590
126. Thomas A Mentzler, Department of Mathematics, University of Colorado - Denver, Campus Box 170, P.O. Box 173364, Denver, CO 80217-3364
127. Consuelo Mulino, Universidad Central de Venezuela, Escuela de Computacion, Facultad de Ciencias, Apartado 47002, Caracas 1041-A Venezuela
128. James McGraw, Lawrence Livermore National Laboratory, L-306, P.O. Box 808, Livermore, CA 94550
129. Paul C Messina, Mail Code 158-79, California Institute of Technology, 1201 East California Boulevard, Pasadena, CA 91125
130. Cleve Moler, MathWorks, 325 Linfield Place, Menlo Park, CA 94025
131. Neville Mory, Department of Mechanical and Industrial Engineering, University of Illinois, 1206 W. Green Street, Urbana, IL 61801
132. Horn P. OLeary, Computer Science Department, University of Maryland, College Park, MD 20742
133. James M Ortega, Department of Applied Mathematics, Thornton Hall, University of Virginia, Charlottesville, VA 22901
135. Chris Paige, McGill University, School of Computer Science, McConnell Engineering Building, 3480 University Street, Montreal, Quebec, Canada H3A2A7
136. Roy P. Porras, Department of Computer Science, Clemson University, Clemson, SC 29634-1906
137. Beresford N Parlett, Department of Mathematics, University of California, Berkeley, CA 94720
138. Merrell Patrick, Department of Computer Science, Duke University, Durham, NC 27706
139. Robert J. Hemmens, Departments of Mathematics and Computer Science, Box 7311, Wake Forest University, Winston-Salem, NC 27109
98. Don E. Heller, Physics and Computer Science Department, Shell Development Co., P.O. Box 481, Houston, TX 77001
99. Nicholas J. Higham, Department of Mathematics, University of Manchester, GI, Manchester, M3 9PL, England
101. Robert E. Huddleston, Computation Department, Lawrence Livermore National Laboratory, P.O. Box 808, Livermore, CA 94550
102. Ilse Ipsen, Department of Computer Science, Yale University, P.O. Box 2158 Yale Station, New Haven, CT 06520
103. Barry Joe, Department of Computer Science, University of Alberta, Edmonton, Alberta T6G2H1, Canada
104. Lennart Johnson, Thinking Machines Inc., 245 First Street, Cambridge, MA 02142-1214
105. Harry Jordan, Department of Electrical and Computer Engineering, University of Colorado, Boulder, CO 80309
106. Bo Kågström Institute of Information Processing, University of Umeå, 5-901 87 Umeå, Sweden
107. Mil y H. Kalos, Cornell Theory Center, Engineering and Theory Center Bldg., Cornell University, Ithaca, NY 14853-3901
108. Hans Kaper, Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Bldg. 221, Argonne, IL 60439
109. Linda Kaufman, Bell Laboratories, 600 Mountain Avenue, Murray Hill, NJ 07974
110. Robert J. Kee, Division 8245, Sandia National Laboratories, Livermore, CA 94551-0969
111. Kenneth Kennedy, Department of Computer Science, Rice University, P.O. Box 1892, Houston, TX 77001
113. Richard Lau, Office of Naval Research, Code 111 MA, 800 Quincy Street, Boston Tower 1, Arlington, MA 02217-5000
114. Alan J. Laub, Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106
115. Robert L. Lauver, Army Research Office, P.O. Box 12211, Research Triangle Park, NC 27709
116. Charles Lawson, M 301-490, Jet Propulsion Laboratory, 4800 Oak Grove Drive, Pasadena, CA 91109
117. Peter D. Lax, Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, NY 10012
118. James E. Leiss, R. 2, Box 142C, Broadway, VA 22815
77. Lars Eden, Department of Mathematics, Linkoping University, 581 83 Linkoping, Sweden
78. Howard C. Ham, Computer Science Department, University of Maryland, College Park, MD 20742
79. Albert M. Brisman, Being Computer Services, Engineering Technology Applications, ETA Division, P.O. Box 24346, MS-7L-20 Seattle, WA 98124-0346
80. Geoffrey C. Fox, Northeast Parallel Architectures Center, 111 College Place, Syracuse University, Syracuse, NY 13244-4100
81. Paul O. Frederickson, NASA Ames Research Center, NAS, MS TD45-1, Moffett Field, CA 94035
82. Fred N. Bishopsby, L 316, Computing and Mathematics Research Division, Lawrence Livermore National Laboratory, P.O. Box 808, Livermore, CA 94550
83. Robert E. Ponder, Department of Computer Science, North Carolina State University, Raleigh, NC 27650
84. K. Gallivan, Computer Science Department, University of Illinois, Urbana, IL 61801
85. Dennis B. Gannon, Computer Science Department, Indiana University, Bloomington, IN 47405
86. Feng Guo, Department of Computer Science, University of British Columbia, Vancouver, British Columbia V6T 1W5, Canada
87. David M. Guy, Bell Laboratories, 600 Mountain Avenue, Murray Hill, NJ 07974
88. C. W. Gear, NEC Research Institute, 4 Independence Wy, Princeton, NJ 08540
89. W. M. Gentleman, Division of Electrical Engineering, National Research Council, Building M50, Room 344, Montreal Road, Ottawa, Ontario, Canada K1A 0R6
90. J. Alan George, Vice President, Academic and Provost, Needles Hall, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1
91. John R. Gilbert, Xerox Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto CA 94304
92. Gene H. Golub, Department of Computer Science, Stanford University, Stanford, CA 94305
93. Joseph E. Gehr, Division 8245, Sandia National Laboratories, Livermore, CA 94551-0969
94. John Gustafson, Ames Laboratory, Iowa State University, Ames, IA 50011
95. Per-Christian Hansen, U.C. Lyngby, Building 305, Technical University of Denmark, DK 2800 Lyngby, Denmark
96. Richard Hanson, IML Inc., 2500 Park West Tower One, 2500 City West Blvd, Houston, TX 77042-3020
97. Michael T. Heath, National Center for Supercomputing Applications, 4157 Beckman Institute, University of Illinois, 405 North Matthews Avenue, Urbana, IL 61801-2300
57. Bill L. Barbrie, Scientific Computing Division, National Center for Atmospheric Research, P.O. Box 3000, Boulder, CO 80307
58. Donald A. Calahan, Department of Electrical and Computer Engineering, University of Michigan, Ann Arbor, MI 48109
60. Ian Caver, Department of Computer Science, University of British Columbia, Vancouver, British Columbia V6T 1Z4, Canada
61. Tony Chan, Department of Mathematics, University of California, Los Angeles, 405 Hilgard Avenue, Los Angeles, CA 90024
62. Jagdish Chandra, Argus Research Office, P.O. Box 12211, Research Triangle Park, NC 27709
63. Eleanor Chu, Department of Mathematics and Statistics, University of Guelph, Guelph, Ontario, Canada N1G 2W
64. Mervyn Grant, National Science Foundation, 1800 G Street NW, Washington, DC 20550
65. Tom Coleman, Department of Computer Science, Cornell University, Ithaca, NY 14853
66. Paul Cusuc, Mathematics and Computing, Lawrence Berkeley Laboratory, Berkeley, CA 94720
67. Andy Gunn, IBM J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598
68. John M. Conroy, Supercomputer Research Center, 17100 Science Drive, Bowie, MD 20715-4300
69. Jane K. Cullum IBM J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598
70. George Cybenko, Center for Supercomputing Research and Development, University of Illinois, 104 S. Wright Street, Urbana, IL 61801-2932
71. George J. Davis, Department of Mathematics, Georgia State University, Atlanta, GA 30303
72. Tim Davis, Computer and Information Sciences Department, 301 CSE, University of Florida, Gainesville, FL 32611-2024
73. Donald J. Ducziak, Department of Nuclear Engineering, 110B Burlington Engineering Labs, North Carolina State University, Raleigh, NC 27695-7909
74. Iain Duff, Atlas Centre, Rutherford Appleton Laboratory, Didcot, Oxon OX11 0QX, England
75. Patricia Gerlekin, Department of Computer Science, SUNY at Buffalo, Buffalo, NY 14260
76. Stanley Eisenstat, Department of Computer Science, Yale University, P.O. Box 2158 Yale Station, New Haven, CT 06520
INTERNAL DI STRI BUTI ON

2. K R Bennett  27. WA Shelton
3. T S. Durland  28–32. R E. Sincovec
5. J M Donato  38. P H Werley
11. G A Geist  40. ORNL Patent Office
13. E G Ng  42. Y 12 Technical Library
14. C E. Diver  43. Laboratory Records Department
15. B W Pyton  - RC
16–20. S A Ruby  44–45. Laboratory Records Department
21. C H Boone

EXTERNAL DI STRI BUTI ON

46. Cleve Ashcraft, Boeing Computer Services, P O Box 24346, MS 7L-21, Seattle, WA 98124-0346
47. Donald M Austin, 6196 EECS Bldg., University of Minnesota, 200 Union St., S.E., Minneapolis, MN 55455
48. Robert G. Babb, Oregon Graduate Institute, CSE Department, 19600 N W Nunn Drive, Beaverton, OR 97006-1999
49. Lawrence J. Baker, Exxon Production Research Company, P O Box 2189, Houston, TX 77252-2189
50. Jesse L Barlow, Department of Computer Science, Pennsylvania State University, University Park, PA 16802
51. Edward H. Bursis, Computer Science and Mathematics, P O Box 5800, Sandia National Laboratories, Albuquerque, NM 87185
52. Chris Bischof, Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439
53. Ake Bjorck, Department of Mathematics, Linkoping University, S-581 83 Linkoping, Sweden
54. Jean R S. Batur, Department of Computer Science, Ayres Hall, University of Tennessee, Knoxville, TN 37996-1301
55. Roger W Brockett, Pierce Hall, 29 Oxford Street, Harvard University, Cambridge, MA 02138
56. James C Browne, Department of Computer Science, University of Texas, Austin, TX 78712


rate execution may be a better use of resources from both the distributor's and user's point of view. Toolpack [7], a large collection of Fortran software tools, and f2c [5], a Fortran to C compiler, are logical candidates for this remote execution service. Users, instead of downloading installing and executing these programs, could submit suitable input and have a machine at a remote server site execute the program and return the output.

Another proposed capability will allow users to add their own servers to the collection of servers globally available through xnetlib. Any xnetlib user could then access this contributed software by adding the appropriate server to this active server list. This feature will greatly expand the amount of software available through xnetlib.

We also plan to expand the scope of the xnetlib system. In the future, xnetlib's distributed repository will include more mathematical software and more reports. Additional software and document collections will be linked into xnetlib's existing collection.

Starting with LAUGxnetlib began distributing entire libraries. This service will be extended to other libraries.

Xnetlib already provides fast and easy access to a large collection of mathematical software. In the future, xnetlib will provide greatly expanded capabilities and will be much closer to being a complete problem-solving environment.

6. Summary

Software distributed by netlib comes with the disclaimer that "any free code comes with no guarantee." In contrast to commercial vendors like NAG and IMSL, netlib offers no support beyond whatever documentation and help authors choose to provide with their code. These caveats also apply to xnetlib.

On the other hand, both netlib and xnetlib provide free, easy access to a large body of high-quality code, and the phenomenal growth of netlib over the past eight years attests to the value of this service. While the xnetlib, by making this high-quality code even more accessible, will encourage software developers to make their code freely available and will make good programming easier for the scientific computing community.

Acknowledgments

xnetlib is the product of the efforts of many people. Jim Berson, David Blat, Shirley Bone, Jennifer Giger, Tony Giger, San Genn, Brian Ilarde, Sharon Jekos, Keith More, Andrew Pearson, Jim Richardson, Bill Reiner, and Adele Van Hill all made valuable contributions to the design, development, and testing of various versions of xnetlib. We also thank the many users of early versions of xnetlib for their constructive comments and for their patience.

7. References

4. Getting started

Xnetlib requires release 4 (or later) of X11 and the Athena widget libraries as supplied by MIT. The executable for the xnetlib client requires approximately 20 kilobytes on a Sun SPARCstation2. The locally cached indexes occupy zero to 80 kilobytes depending on how many index files are cached.

The first step in installing xnetlib is to obtain the source code for the xnetlib client. The software for the xnetlib client is itself available from both xnetlib and netlib, so a simple way to obtain the source is to send the message send xnetlib.tar from xnetlib to netlib@cornl.gov. Netlib will respond by sending a shar file containing all necessary source code and documentation. Xnetlib is also available by anonymous ftp from cs.utk.edu in the pub/xnetlib directory. The xnetlib distribution includes an INSTALL file so installation is usually trivial if the XWindows System has been configured properly on the client machine.

Xnetlib is easily customized. The common customization at multi-user sites is to have a single cache of index files so indexing information can be shared by all local users.

5. Plans

Plans are already underway to expand xnetlib. One major addition will be the capability of remote execution. Many useful utility programs are large, making distribution tedious, or are more expensive to build than to execute. In such cases, allowing re
A user may also wish to search by keyword instead of viewing the contents of a particular library. In this mode, descriptions of files are searched by a keyword string the user provides. The keyword search can be a search on the intersection or union of the words in a search string, a literal search for an exact string (with or without case sensitivity), or a fuzzy search based on the latest semantic indexing technique [2].

A recent semantic indexing uses statistical analysis to find useful matches that may not be uncovered by other types of searches. In contrast, the fuzzy search capability in WAIS is based on a heuristic rather than statistical approach.

Clicking on the download button causes xnetlib to display a list of selected software and documents. Clicking on the download path button allows a user to change the directory to which files will be downloaded. The dependency checking button is a rool switch. If dependency checking is off, xnetlib will send only the selected routines. If it is on, xnetlib will send the selected routines and any routines that they call. If the user is satisfied with the selection list and the target directory, he should click on get files now to initiate the transfer. Figure 5 shows the downloading of selections made from the MAC, UNIX, and INLibs. These three libraries reside in repositories at separate sites. The selected files will appear in the specified directory usually within a few seconds.

The who command allows a user to search the NASTWEP pages [4,6], a database containing information about individuals interested in numerical analysis and other disciplines. For simple searches the user need only enter an individual’s last name. The modify search feature can be used for more elaborate searches, and the modify listing feature can be used for controlling the format of the output.
Figure 2: Library nnm

Figure 3: LAPACK Index
client and server, requested indexes are cached locally. Frequently requested information can therefore be quickly retrieved from local caches instead of repeatedly retrieved from the remote server. Other requests are passed to the server via sockets. Section 3 describes the use of the interface in more detail.

3. Features

Xnetlib features and capabilities include:

- Access to a distributed repository
- Searching by a software libraries list
- Searching by software classifications
- Searching by keyword
- Software and document retrieval
- Access to the NMTWrite pages
- Online help

Many Internet sites have sizable collections of documents or software. It is both unnecessary and undesirable to require that these collections reside at a single site. Xnetlib gives users access to a distributed repository of software and documents by establishing socket-based links with the repository sites. Users have access to any or all of these repositories through a single interface.

Xnetlib users enter which sites are linked into the distributed repository using the set up button. Great repository sites include netlib@ornl.gov, spark.brl.mil, and softlib@rice.edu. Ciding on the timely message button displays news about individual repository sites and clicking on the index button displays their general indexes.

Clicking on the library button show the libraries that are available through xnetlib. Figure 2 shows a unified list of software and documents available at the repositories of Oak Ridge National Laboratory, Rice University, and the US Army Research Laboratory.

Clicking on a library name lists the contents of that library. For example, clicking on a pack displays a partial listing of LAPack contents (Figure 3). The complete contents list of LAPack is too large to fit in the window, but the provided scrollbar allows the user to scroll through the rest of the list. Clicking on the box adjacent to a routine name selects that routine for future downloading. In Figure 3, the user has selected sgetrf and sgetrs from the LAPack library.

The classification feature allows a user to narrow a search. The classification of the xnetlib software libraries is based on the ACM[15] classification system augmented to include classifications other than mathematical software. Selecting linear algebra causes the names of the libraries with linear algebra software to be displayed (Figure 4).
Portability. The system's implementation should be as portable as possible.

Accessibility. System should be accessible to a large number of users.

Existing software and document retrieval systems satisfied these requirements. FTP and Archie are fast and portable but are not sufficiently flexible in their search or record keeping capabilities. In addition, in Archie the indexing mechanism and the large volume of accessible material make fully up-to-date indexing information impractical. Gopher is geared to browsing through the Internet rather than to retrieval of materials, while WAIS is better suited for retrieval of documents than for retrieval of software.

Archie, gopher, and WAIS have different goals than xnetlib and should not be regarded as competitors. Their use can, in fact, be complementary. For example, gopher can be used to provide improved accessibility to netlib and xnetlib.

Figure 1 shows the basic configuration of the xnetlib system. The system consists of the xnetlib server processes running on machines at the repository sites, xnetlib X client processes running on users' local machines, and TCP/IP socket-based communication links between these clients and servers.

Xnetlib's server process runs continually at an xnetlib repository site, listening for incoming requests from xnetlib client processes. Typically, the xnetlib server runs on the same machine and uses the same software repository as the netlib server. Upon receiving a request, the xnetlib server determines the nature of the request and responds by transferring the appropriate file from the repository to the xnetlib client process.

The xnetlib client running on the user's local machine provides an X-Windows interface to the xnetlib repository. It is programmed in C using the Athena widget libraries. This interface makes searching through the software and document collection easy. For example, an xnetlib user can view the contents of any library simply by clicking a button. Other commands, such as keyword searching or requesting software, also require just a few button clicks. To add unnecessary communication between the
Netlib is a new software distribution tool recently developed at the University of Tennessee and Oak Ridge National Laboratory. The goal in developing netlib was to provide Internet users faster and easier access to netlib's large collection of software, data, and documents. Unlike netlib, which uses e-mail to process requests for software, netlib uses an X-Windows interface and socket-based communication between the user's machine and the netlib server 1 to process software requests. This enables users to search through a large, distributed collection of software easily and to retrieve requested software in seconds.

1. Background

Netlib's predecessor, netlib, grew from a need to have a quick and easy method for distributing small pieces of mathematical software. Netlib services began in 1985 at two sites, Argonne National Laboratory and ANL laboratories, and distributed software from about 30 libraries. For additional information about netlib's operation and use see the introductory paper by Kogbara and Golub [3].

One of the changes since netlib's introduction has been the transfer of netlib services from Argonne National Laboratory to Oak Ridge National Laboratory. Also, the availability on netlib of the netlib programs itself has enabled many other sites to set up their own software repositories. 2 The netlib software collection has now grown to well over 100 libraries. The number of software requests sent to netlib also has grown dramatically. The most heavily used netlib server, at Oak Ridge National Laboratory, processed over 130,000 requests last year.

2. Overview

When netlib's design began, the following requirements were specified:

- **Speed.** Retrieving software should take seconds, not minutes as typically required by e-mail.

- **Usability.** The user interface should make searching through a large collection of software and documents easy.

- **Organization.** The system's repository should be an updated collection, with up-to-date indexes, and a database organized to facilitate searching and ease of retrieval. The repository may be distributed over several sites.

- **Record Keeping.** The system should have the capability of logging requests so that dates and locations can be reported to users.

- **Security.** The system should be secure from accidental or intentional misuse.

---

1Throughout this paper, server refers to the process handling software requests and not to the X display server.

2Send the message send index or send sites from netlib to netlib@ornl.gov to receive a list of netlib sites.
SOFTWARE DISTRIBUTION USING XNETLIB

Jack J. Dugaraa
Tomas H. Ruan
Red C. Wei

Abstract

Xnetlib is a tool for software distribution. Xnetlib's predecessor netlib uses e-mail as the user interface to its large collection of public-domain technical software. Xnetlib uses an XWindow interface and socket-based communication. Xnetlib makes it easy to search through a large distributed collection of software and retrieve requested software in seconds.
Contents

1 Background .................................................. 1
2 Overview .................................................... 1
3 Features ..................................................... 3
4 Getting started .............................................. 6
5 Pans .......................................................... 6
6 Sunny .......................................................... 7
7 References ................................................... 7
SOFTWARE DISTRIBUTION USING XNETLIB

Jack J. Irigoyen  ††
Tom H. Ryan  ‡
R. C. Wade  †

† Department of Computer Science
University of Tennessee
111 Ayres Hall
Knoxville, TN 37996-1301
‡ Mathematical Sciences Section
Oak Ridge National Laboratory
P.O. Box 208
Oak Ridge, TN 37831-2080

Draft Released: June 1993

Research was supported by the Applied Mathematical Sciences Research Program of the Office of Energy Research, US. Department of Energy, and by the Defense Advanced Research Projects Agency and the Army Research Office under contract DAAG 37-86-C-0157.

Prepared by the
Oak Ridge National Laboratory
Oak Ridge, Tennessee 37831
managed by
Martin Marietta Energy Systems, Inc.
for the
US. DEPARTMENT OF ENERGY
under Contract No. DE-AC05-84O28400