



LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

Hopper File Management Tool

J. W. Long, N. J. O'Neill, N. G. Smith, R. R.
Springmeyer

November 17, 2004

Nuclear Explosives Code Development Conference
Livermore, CA, United States
October 4, 2004 through October 7, 2004

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

UNCLASSIFIED

Proceedings from the NECDC 2004

Hopper File Management Tool (U)

**J.W. Long,* N.J. O'Neill,* N.G. Smith,*
R.R. Springmeyer***

*Lawrence Livermore National Laboratory, Livermore, California 94550

Hopper is a powerful interactive tool that allows users to transfer and manipulate files and directories by means of a graphical user interface. Users can connect to and manage resources using the major file transfer protocols. Implemented in Java, Hopper can be run almost anywhere: from an individual's desktop machine to large production machines. (U)

Introduction

In a high-performance computing environment, managing files can become a difficult and time-consuming task that distracts from scientific work. Users must deal with multiple file transfer protocols, transferring enormous amounts of files between computer platforms, repeated authentication, organizing massive amounts of data, and other detailed but necessary tasks. This is often accomplished with a set of several different tools, each with its own interface and idiosyncrasies. Our goal is to develop tools for a more automated approach to file management that substantially improves users' ability to transfer, organize, search, and operate on collections of files. This paper describes the Hopper tool for advanced file management, including the software architecture, the functionality, and the user interface.

File Management History at LLNL

File management can certainly be accomplished with a collection of Unix command line tools such as ftp, ls, cd, cp, mv, and find. This requires that users correctly type file names, remember the correct syntax, and perform operations that require arcane knowledge and low-level manipulation of individual files. In the late 1990s two tools were developed at LLNL to provide users with graphical user interfaces for more easily managing files in an increasingly distributed computing environment: XFTP and XDir.

XFTP provides a simple point-and-click interface for moving files between two hosts. A particularly useful feature is its ability to do third-party transfers, whereby users can move files from one remote platform to another. The XDir program uses the familiar folder-icon metaphor to simultaneously display windows representing several hosts, rather than just two. Users initiate FTP transfers by dragging and dropping files between

Long, J.W. et al.

UNCLASSIFIED

UNCLASSIFIED

Proceedings from the NECDC 2004

windows, which represent any combination of local and remote machines. XDir can be used to browse directory structures, transfer files, delete files and directories, rename files and directories, and search directories on one or more hosts for entries matching a specified pattern. Based on UNIX, the C programming language, OSF/Motif, and FTP, XDir is highly portable.

Computer users at LLNL and elsewhere showed their preference for a graphical user interface by heavily using both of these FTP-based tools. With the introduction of new protocols and techniques for moving data, though, the file management picture has become more complex. In addition, the advent of the High Performance Storage System (HPSS) and ever-larger MPP systems have altered the scale of data being managed and archived. LLNL users now have additional requirements and goals for easing data access and management in our distributed, heterogeneous environment.

These new requirements include using secure protocols such as ssh and sftp, using high-speed or persistent protocols such as pftp or Endeavor (NFT), and using custom protocols such as the HPSS client API and Htar. Other requirements unmet by the older tools include concurrent operations, advanced searching by filename attribute and content, access checking, synchronizing directories between systems, and cross-platform portability.

Hopper addresses these requirements, providing an easy to use graphical interface that handles multiple protocols, runs on nearly all platforms, and performs operations concurrently. This is possible due to an architecture built on abstraction layers that was implemented with the use of Java, concurrent programming, and sophisticated graphical user interface programming.

Hopper Architecture

One of the key characteristics of Hopper's architecture is concurrency. Any number of transfer or search operations can occur simultaneously, all without interfering with the user's ability to interact with his or her directory windows. A sophisticated set of listeners and thread managers keep track of the various threads and make sure information is properly collected and reported.

Another important characteristic of Hopper's software architecture is its ability to provide access to new protocols without changing the core design. This generality was accomplished by using Java Interface classes and other object oriented techniques provided by the Java programming language.

Long, J.W. et al.

UNCLASSIFIED

UNCLASSIFIED

Proceedings from the NECDC 2004

Data Flow

The following diagram illustrates how data, commands, and metadata are distributed amongst various hosts when using the Hopper file management tool. Hopper is able to execute commands on remote hosts, causing actions to be taken and data to be moved. Metadata moves via a middleware server to the metadata database, or metabase (Figure 1.)

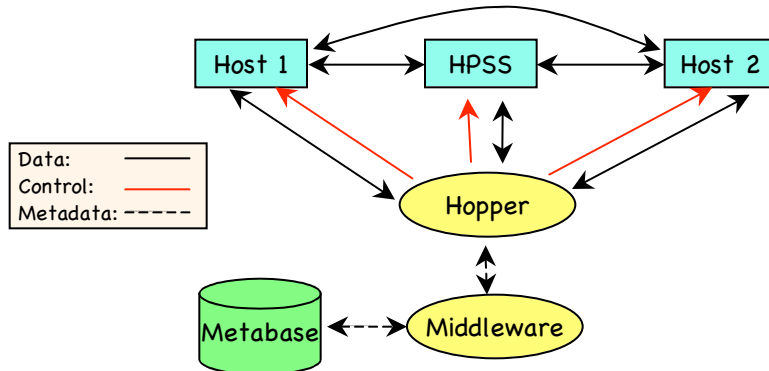


Figure 1. Hopper Data and Control Flow

Software Organization

Hopper uses abstraction layers to easily extend its capabilities. These abstraction layers allow new services and protocols to be supported without requiring changes to the parent application. The concept is very similar to plug-ins (Figure 2.)

Hopper		
FileSystem Interface	Metadata Interface	
File Systems	XML	MW Server
SSH FTP HPSS HTAR ...		Database

Figure 2. Hopper's key abstraction layers.

The most important abstraction is the virtual file system, which provides a uniform interface to all of the various underlying technologies supported by Hopper. We have implemented ten virtual file system interfaces thus far, including FTP, SSH, SFTP, and TAR. New virtual file systems are straightforward to implement and are easily incorporated into Hopper.

The key software abstractions in Hopper include:

- The FileSystem interface connects to underlying technologies such as FTP, SSH, TAR, and so on.

Long, J.W. et al.

UNCLASSIFIED

UNCLASSIFIED

Proceedings from the NECDC 2004

- The Metadata interface connects to local and remote metadata sources.
- The Copy interface connects FileSystems during transfers, providing recursion, two- or three-hop transfers, etc. as needed.

The power of Hopper's copy interface is exhibited when transferring items between dissimilar FileSystems. For instance, one can transfer from a remote tar file on one host to another remote host connected via ftp (Figure 3.)

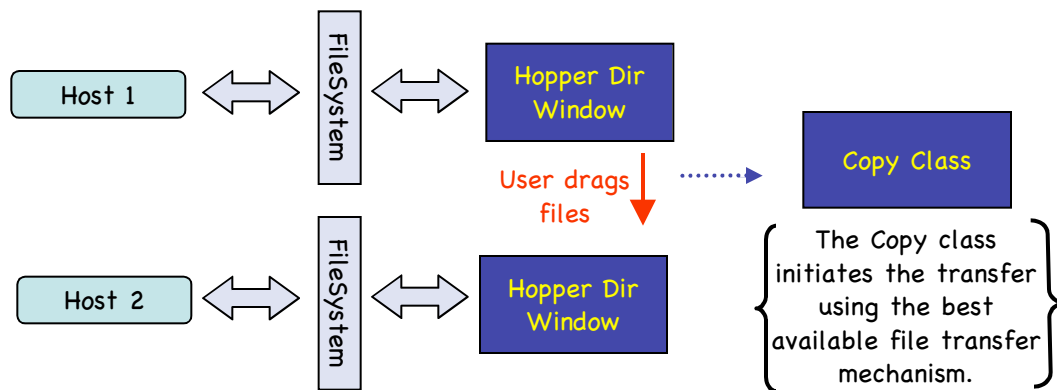


Figure 3. Hopper's copy class architecture.

Functional Capabilities

Directory browsing

The directory browsing function allows multiple display formats for any combination of local and remote hosts. There is no limit on the number of connections one can have with Hopper, nor on the number of windows. Directories can be displayed in a compact iconic view, where just the type (file or directory) and name are visible. The user can control the number of columns and the sort order used. Hopper also can display directories in a hierarchical, detailed view. This view shows a wide variety of attributes, including size and date, and lets the user sort on any of these fields. An optional caching feature speeds directory browsing of slow systems.

File transfer

The file transfer function allows the recursive transfer of files and directories. Any number of transfers can occur simultaneously. Transfers can be performed from any readable file system to any writable file system. Hopper chooses the appropriate transfer protocol to use based on the protocols of the source and destination file system. If

Long, J.W. et al.

UNCLASSIFIED

UNCLASSIFIED

Proceedings from the NECDC 2004

necessary, Hopper can perform multi-hop transfers, e.g., using an intermediate step if there is no direct link between the source and destination hosts.

Directory and file manipulation

The directory and file manipulation function allows the user to rename entries, to delete entries (including entire directory substructures), to move entries between directories of same host, to create directories, to create symbolic links, and to set the permissions and ownership of entries. These capabilities are available on any writable file system. Recursion is provided automatically by Hopper for these operations, even if the underlying file system does not support recursion inherently.

Selection mechanisms

Entries can be selected in a variety of ways, including graphical – clicking individual entries, extending a selection from point A to point B, and sweeping a rectangular area, and textual – entering wildcard selections and matching typed characters on the fly.

Searching

This function provides the ability to search directory structures for files or directories matching specified criteria. The user can search by name using literal patterns, wildcards, or regular expressions, or can search by other file attribute such as file size, creation date, or owner. The efficient Unix search commands, `find` and `locate`, are used when possible because they produce rapid results, but searches can be done just as easily on any non-Unix host to which Hopper can connect. A search can be run concurrently with transfers or other searches.

Searches can also be performed on the contents of files on Unix-based systems. This feature allows the user to specify a pattern to look for inside of a file; matching files and lines are displayed. The search on content feature can be combined with search on name, creating a unique and powerful file location capability.

Directory synchronization

This function provides `rsync`-style directory synchronization operations. Like other data movement operations, this is initiated via a simple drag-and-drop interface. It is as easy as dragging a directory icon from one host onto a directory icon from a second host. Note that the synchronization operations can be done between non-uniform hosts, including between a local desktop machine and HPSS.

Before a synchronization operation is actually invoked, a tabbed dialog is displayed that describes the operation and allows one to set various options. Source files not present in the destination directory are always copied. Three main options allow the user to control the synchronization of the remaining files: (a) if a file exists at both the source and destination, the source file is copied only if it is newer, (b) if a file exists at both the

Long, J.W. et al.

UNCLASSIFIED

UNCLASSIFIED

Proceedings from the NECDC 2004

source and destination, the source file is copied if it differs, regardless of age, and (c) create an exact mirror of the source, including deleting destination files if necessary.

Authentication management

This function results in a reduction in the number of password prompts the user sees. Freed connections are cached and re-used, and authentication information is stored during the session and re-used whenever possible. Site-specific information is used when possible, such as detecting a one-time password and not caching it.

Configurable interface

The graphical user interface allows the user to tailor Hopper's behavior and appearance. Controls related to the look-and-feel include font size and family, window feature colors, double-click speed, the sort order for various listed presented in the GUI, and time/date format.

Behavioral controls in Hopper include whether or not to confirm operations such as delete, the keep-alive time for FTP connections, the behavior of the transfer monitor, and default initial options for various operations such as synchronizing directories.

History mechanism

This function automatically remembers search expressions, previously visited directories, and host and user names used in connecting to remote hosts. This information is remembered across sessions, providing continuity for the user.

Xterm launching

This function, available when running Hopper on an X Window system, allows one to launch xterm windows on a local or remote Unix host with a click of a button. For convenience, xterm will be started in the same directory that the Hopper browser window is viewing. In most cases authentication is handled by Hopper itself.

Access checking

This function reports the users and groups that can read or write a given file, taking into consideration the permissions of all enclosing directories as well as the file itself. This information can be presented in two ways: a symbolic form ("Anyone belonging to both group A and group B"), and an explicit form (a list of actual user names.)

This is useful for verifying that a particular person or group has access to a file, especially after making permission and group changes in an effort to make that file available. This feature is also useful from a security aspect in that one can easily get a list of all users who can read a given file. This information can be very time consuming to generate using other tools.

Long, J.W. et al.

UNCLASSIFIED

UNCLASSIFIED

Proceedings from the NECDC 2004

Metadata association

This function allows the user to associate notes with a file or directory. These notes are recorded either in a local textual representation or in a remote database serviced by a middleware server. The user can view and edit these comments in Hopper's GetInfo view.

Htar and tar archive reading and creation

This function allows users to open and create htar and tar archives. In directory listings, htar archives are shown with a custom "H" icon. Double clicking on one of these icons will show the directory listing for the htar archive. Files can be copied out of an htar directory just like any other directory. Files cannot be copied into an htar directory, however, as htar archives are read-only. Tar archives appear with a "T" icon and are operated on in the same fashion as htar archives. In addition, htar and tar archives can be easily created using the drag-and-drop interface.

User Interface

A graphical, drag-and-drop interface to file management capabilities is an essential element of a successful file management tool. Hopper's user interface is modeled after key elements of the XDir file management tool, plus other Internet applications such as web browsers.

Using the Java language for Hopper makes it a true cross-platform application, capable of running under Unix, Windows, and OS X. This makes it possible for users to use the same data management tool on the large production machines as they use on their local desktop machines. Having a common user interface on all machines was an important consideration in the choice of Java.

Command line options

On Windows and Mac systems, starting Hopper is as easy as double clicking the Hopper icon. On Unix systems, Hopper is invoked via a simple start-up script by typing "hopper" at the command prompt. Options to the script allow the user to choose different versions of the code to execute (e.g., the current beta version or the development version), as well as activating debugging features.

Iconic view

Hopper provides an option for viewing file systems in an iconic representation. File names and basic file type information are displayed via icons and names (Figure 4.) The user can control the number of columns, as well as the alphabetic sort order within the window. Options include by-row, by-column, and the inverse of each.

The current directory is displayed in a text field, which the user can either edit directly or choose from previously visited directories by clicking on the history arrow.

Long, J.W. et al.

UNCLASSIFIED

UNCLASSIFIED

Proceedings from the NECDC 2004

Navigational and functional operators are available via the toolbar and menus, and are identical in this view as in the detailed view.

The navigational buttons on the left of the toolbar were modeled after web browsers, and provide a convenient way to return to previously visited directories from the current session. The up arrow allows direct access to any parent directory.

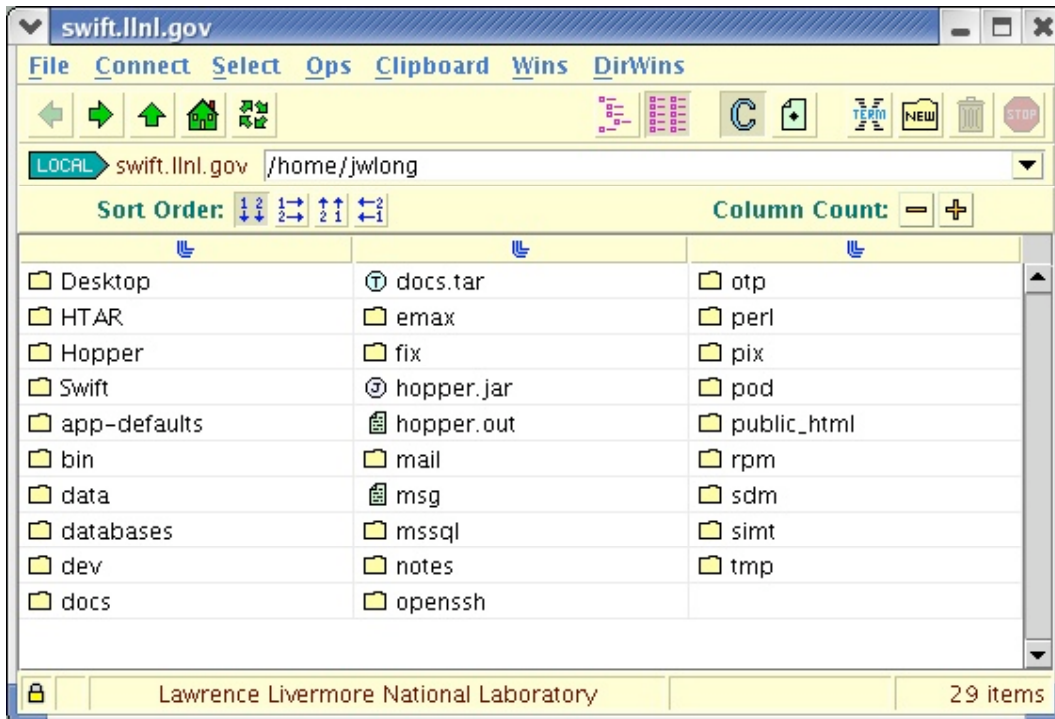


Figure 4. Hopper's iconic view. Note custom icons for certain file types.

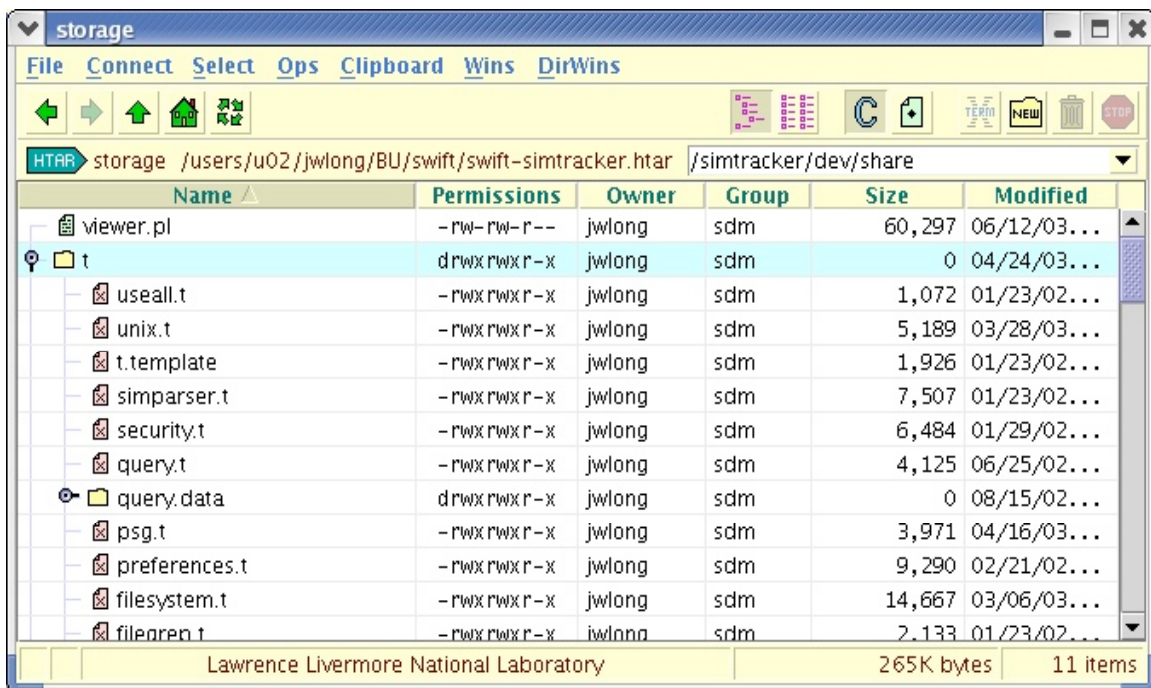
UNCLASSIFIED

Proceedings from the NECDC 2004

Detailed view

Hopper also provides an option for viewing files with a detailed representation, where file attributes such as size and permissions are displayed. The user interface allows control over which file attributes are seen (Figure 5.)

Sorting on a particular attribute is as easy as clicking on the relevant header. By clicking on the header again the sort is inverted. The number of items and the combined size of files in the directory are shown at the bottom of the window. Directories can be viewed in place by clicking on the triangle next to a directory icon. There are a variety of ways for opening a new window showing a given directory.



Name	Permissions	Owner	Group	Size	Modified
viewer.pl	-rw-rw-r--	jwlong	sdm	60,297	06/12/03...
t	drwxrwxr-x	jwlong	sdm	0	04/24/03...
useall.t	-rwxrwxr-x	jwlong	sdm	1,072	01/23/02...
unix.t	-rwxrwxr-x	jwlong	sdm	5,189	03/28/03...
t.template	-rwxrwxr-x	jwlong	sdm	1,926	01/23/02...
simparsar.t	-rwxrwxr-x	jwlong	sdm	7,507	01/23/02...
security.t	-rwxrwxr-x	jwlong	sdm	6,484	01/29/02...
query.t	-rwxrwxr-x	jwlong	sdm	4,125	06/25/02...
query.data	drwxrwxr-x	jwlong	sdm	0	08/15/02...
psg.t	-rwxrwxr-x	jwlong	sdm	3,971	04/16/03...
preferences.t	-rwxrwxr-x	jwlong	sdm	9,290	02/21/02...
filesystem.t	-rwxrwxr-x	jwlong	sdm	14,667	03/06/03...
filearen.t	-rwxrwxr-x	jwlong	sdm	2,133	01/23/02...

Lawrence Livermore National Laboratory 265K bytes 11 items

Figure 5. Hopper's detailed view. Note directory controls on the left side.

File management operations

Hopper provides a graphical way to perform simple and complex operations. The interface provides access to the essential options of the underlying capability while hiding the unnecessary complexity from the user. In some cases, such as the search operation, this is accomplished by providing both "simple" and "expert" modes. In other cases, such as the change permission dialog, user preferences allow the user to see as many or few of the power-user options as they wish.

Long, J.W. et al.

UNCLASSIFIED

UNCLASSIFIED

Proceedings from the NECDC 2004

The synchronize interface has default values which make standard sync operations very simple (Figure 6.) The user wishing to do more complex sync operations merely has to define them from a well-defined set of power-user controls in other tabbed windows.

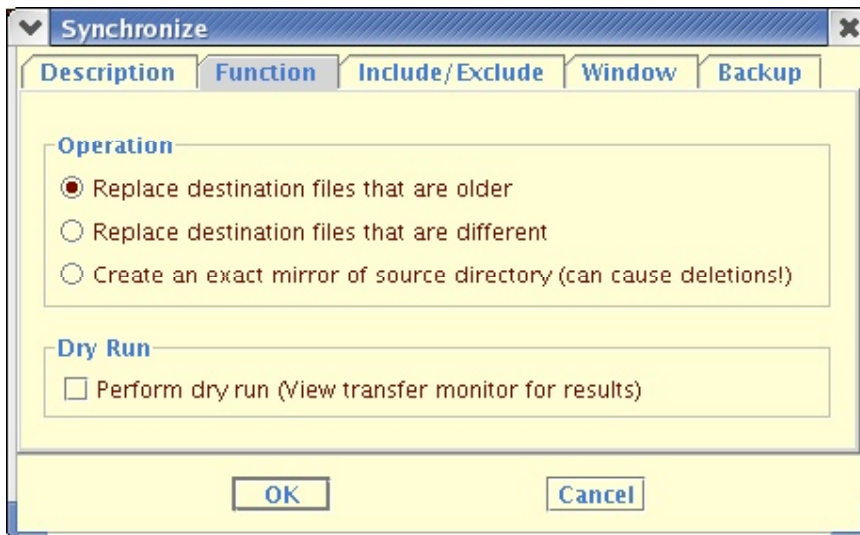


Figure 6. Hopper's synchronize directory feature makes it easy to perform standard sync operations while hiding unnecessary complexity.

Conclusion

Hopper is a powerful tool that is useful for managing files on an individual's desktop as well as for navigating file management issues across all production platforms and archival storage. It provides the advanced capabilities needed by today's users.

Acknowledgements

We acknowledge Neale Smith as the author of XFTP and XDir. We thank the users who have contributed to Hopper by providing feedback throughout the entire development process. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract number W-7405-ENG-48.

References

Long, J. W., Spencer, P., and Springmeyer, R. "Simtracker – Using the Web to Track Computer Simulation Results," Proceedings of the 1999 International Conference on Web-Based Modeling and Simulation, San Francisco, California, 1999. Simulation Series Vol. 31, No. 3, The Society for Computer Simulation.

Long, J.W. et al.

UNCLASSIFIED

UNCLASSIFIED

Proceedings from the NECDC 2004

Intelligent Archive: Integrated Data Access and Organization for Scientists, LLNL technical Report, UCRL-TB-118571 (1995). Available at <http://www.llnl.gov/ia/>.

LLNL XDir: Advanced Graphical FTP Client, LLNL Technical Report, UCRL-TB-118584 (1997). Available at <http://www.llnl.gov/ia/xdir.html>.

Hopper Web Site, UCRL-WEB-204807. Available at <http://www.llnl.gov/hopper>.

Long, J.W., O'Neill, N.J., Smith, N.G., and Springmeyer, R., "Hopper Poster", NECDC 2004, October 4-7, 2004, UCRL-PRES-207012.

Long, J.W. et al.

UNCLASSIFIED