A Single Interface for Providing User Services

Jay Boisseau and *Dale Clark,* Arctic Region Supercomputing Center, Fairbanks, Alaska

ABSTRACT: *We propose a single interface for providing user services, from online documentation to interactive services. User services are typically provided through various mechanisms with a heterogeneous set of interfaces, of which the typical user will be familiar with only a subset. For example, documentation may exist in a variety of formats, each requiring a different access mechanism. Interactive services such as account applications or requests for assistance likewise may each be handled differently. ARSC is developing a World Wide Web server that will provide a single, intuitive interface for all online user services, which will enable users to utilize the resources effectively.*

Introduction

The Importance of Providing User Services

Providing services such as documentation, training, and support to users at high performance computing (HPC) centers is necessary to ensure that the center's facilities are used with maximum effectiveness. HPC centers provide powerful hardware and software to enable simulations that cannot be performed via other means such as workstations. The hardware and software used at HPC centers typically has a very small user base, so there are very few people familiar with these resources. More importantly, such advanced hardware and software is necessarily improving at all times, and is thus in a constant state of development. This makes it difficult for users to master a system or application before new features are added, or before the system or package is succeeded by something even more powerful. These factors combine to create an environment in which users will have trouble developing expertise, and yet will need expertise to use the resources efficiently. Providing user services effectively therefore becomes an essential endeavor in the operation of an HPC center.

The Variety of User Services and Means of Provision

The effective provision of user services is complicated by the enormous number of mechanisms by which these services are typically provided. Considering the single example of trying to obtain help using a software package. Depending on the information being sought (purpose, default version, usage, options, etc.), the information might be available in one or more (but probably not all!) of the following ways:

- **man** pages
- vendor-supplied online help systems, which have an interface unique to the particular software package
- text files written locally or by the vendor, stored in a subdirectory that contains either pieces of the software package or documents describing other packages
- **docview** and similar utilities for system-specific software
- /etc/motd announcements
- /usr/news files
- locally written training class notes
- locally written documents on an anonymous ftp server
- e-mail to a consultant, or vendor support staff
- phone call to a consultant, or vendor support staff

The number of methods of obtaining information can be bewildering to the average user. Most users will be familiar with only a subset of these methods. Even the few users familiar with all of these methods will often be unaware of the "correct" method—the method providing the easiest access to the necessary information—for obtaining assistance with a given software package, because it varies from package to package. The upshot of the situation is that users will often have a difficult time obtaining the information they seek, and therefore may not always seek it even though it would aid them because they do not believe the time spent will be rewarded. This creates a situation in which everyone loses:

- The user's time is wasted doing things incorrectly or inefficiently and his/her research may suffer.
- The center's resources are utilized inefficiently and other users may suffer.
- The staff's time is wasted having to support users who do not access the information that is available, and the activities of the staff may suffer.

This example is indicative of the situation for *all* users services. To accomplish such basic tasks as acquiring accounts, increasing allocations or quotas, reporting a problem, receiving training, reporting results, and renewing accounts, users must employ a variety of mechanisms including those above and more. Knowing the best (*i.e.*, quickest, easiest and most thorough) method of obtaining every service becomes impossible, and the attempts to obtain these services become time-consuming and frustrating.

The World Wide Web

The World Wide Web (hereafter Web) has provided a remarkable increase in the ease with which information and services may be provided and accessed electronically. There are three major components of the Web that allow information and services to be provided and accessed effectively: the HyperText Markup Language (HTML) for creating Web documents, the Common Gateway Interface (CGI) scripts that allow information to be passed back to the server (and much more), and the HyperText Transfer Protocol (HTTP) for serving documents on the Web. These features are described briefly below.

HTML documents are *hypertext* documents. Hypertext documents contain links to other documents, graphics, sounds, etc. that are embedded within the document. These links make hypertext much more powerful than regular text. The information available in either kind of document is the same, but the hypertext document allows *related* information to be accessed much more easily. The reader can quickly access information that explains some facet of the original document in greater deal or explains some related topic. This is accomplished while reading the original document merely by clicking on the link. The user can quickly revert back to the original document or follow additional links for even more detail or more related information.

HTML is a *markup language*. It provides the capability to give a document logical structure, with the exact appearance being determined by the display device. The author marks headings, body text, lists, etc. HTML allows documents to be prepared with comparable ease to any word processing software, and allows the author to focus on content rather than appearance. These documents are displayed by the user's browser (client) software, which determines how to interpret a heading or a list. In addition, the user can save the documents in a variety of formats including standard (formatted) text or PostScript.

CGI scripts are used in HTML documents extend the capabilities of the Web almost infinitely. HTML documents can be accessed by any user with a web browser, but CGI scripts allow the user to pass information back to the server. A common use of a CGI script within an HTML document is to return the input in an HTML form to the Web server. This two-way flow of information enables the Web to provide interactive *services* as well as information. Writing scripts is more difficult than preparing HTML documents, but should become easier as the Web grows in popularity and more scripts are made available.

An HTTP daemon allows a computer (the web server) to serve information to users anywhere on the Internet with access to a web browser. Virtually any computer on the Internet can be a web server. HTTP allows information to be accessed on any web server on the Internet, regardless of the previous connection, with speed and ease. Thus, the information on the Internet can be linked in a true 'web'. This web of hypertext links is essentially transparent to the user, who merely follows links to documents on web servers regardless of geographical location.

The Web thus provides a language for creating documents with embedded links in a common format, and CGI scripts to extend the power and flexibility of a web server to exchange information. The interface is uniform for all documents, as all documents contain hypertext links. Web browsers provide simple navigation controls in addition to the links. Understanding how hypertext links and browser navigation controls work is sufficient to navigate all documents available on the Web. Thus, as a user interface, the Web provides unrivaled potential for distributing information and providing services electronically, and is limited only by the efforts of the people responsible for preparing HTML documents and writing CGI scripts.

Developing a Web Server for Providing User Services

Rationale

The Web is a natural means for providing services to users at HPC centers. All users will already be online, and all will need support in order to learn to use new, powerful software and hardware effectively. Because of the unique features described above, the Web has tremendous ability to provide all manner of user services, from documentation to interactive assistance services. It can replace all of the current interfaces for providing user services in most cases, and the single interface makes acquiring these services easier for the average user. This promotes more efficient utilization of time and resources.

Considerations

There are two important considerations for designing a web server for providing user services. The first is to *design a web server for HPC users*. The emphasis in the development of the server documents should be on providing information and services to encourage HPC users and promote optimal usage of resources. This rather obvious consideration is often ignored in the development of web servers, and not only at HPC centers. The Web is a fascinating new technology that allows essentially anyone to publish information accessible to virtually anyone else online. All too often the potential of this worldwide audience seduces web server developers into creating servers with mass appeal instead of functionality for their clients. Such an approach is valid if the purpose of the server is only to gather new clients from all demographic groups, but is inappropriate for providing services to existing clients of a specific demographic. A survey of many HPC centers' web servers reveals that many seem to have been developed to appeal to the casual web "surfer" rather than HPC users. It is more difficult to find necessary information on such servers, and impatient users are likely to give up and resort to multitudinous "tried and true" methods for obtaining user services, which are in reality less effective due to the difficulties in mastering each interface.

The second consideration is to *provide access to* ***all*** *information and services with a single, consistent interface.* The importance of this consideration cannot be overstated. Developing a web server takes time, but if a web server is incomplete, it does no more than provide yet another method by which some, but not all, important information is made available. If users must regularly resort to a different interface to obtain some information or service, they will not adopt a 'Web first' approach. It is wise to delay the availability of a web server initially until the most useful, most requested information is available on the server.

Caveats

As a tool for serving information, the Web has become extremely popular. However, as with any new method to present information, there is the potential to misuse features. Some of the common pitfalls of web documents are:

- *overuse* of features—too many links, too much graphics, etc.
- focusing on *appearance* instead of content and structure
- forgetting *target* audience and preparing documents for global audience
- not editing documents with the same strict procedures used for hardcopy documents

When developing web documents, the emphasis should remain on informing the intended audience. Overuse of hypertext links and graphics will distract the user without providing additional information. The documents should be written for the intended audience, not for the global community, and should receive the same attention to detail that hardcopy documents for the intended audience receive.

Benefits

The benefits of providing all user services via a simple, intuitive online interface developed with HPC users in mind are numerous. Better informed users will use resources more efficiently, which will enable more high quality work to be performed. Such users also require less assistance; the effort put into developing an effective web server can be more than compensated for by the savings in staff time spent repeatedly addressing common issues: basic questions, help with software usage, access to application materials, etc. A well-developed server will make user services very accessible.

ARSC's Web Server

Design and Development

The ARSC web server has been designed from the ground up—or in this case the home page up— as a server for HPC users first and foremost. The organization of the documents closely resembles the format of a user's guide, with the added power of hypertext. We used a table of contents approach for the home page, as HPC users will often be familiar with manuals. This approach makes the home page easy to scan for useful information, and minimizes the time spent searching for the desired document. The home page is the launching point for most users, and should not serve as an extra layer through which users must pass to obtain useful information. Some HPC centers have not followed this approach, providing instead only a few links and lots of attractive graphics with mass appeal. These home pages, however, make it difficult for a user to discern the correct link to obtain a very specific piece of information, since the use of only a few links on the home pages forces categories to be very broad.

Some features that illustrate how ARSC's web server was designed for HPC users are listed below:

- The emphasis is on assistance and usage rather than descriptive information.
- The home page reads like a table of contents from a User's Guide. Non-technical information about ARSC is provided in the first section ('Introduction').
- The middle sections ('Chapters') give the information necessary to decide what resources to use, how to acquire an account, and how to use the resources
- Information not directly related to a User's Guide is provided in the last section ('Appendices'?).
- The button bar enables quick access to help, system status, and new items, and allows searching for quick access to information.
- The User Services department develops and maintains the web server, and forces the focus to remain on users instead of the general public.

The ARSC web server was designed from day one with the first consideration mentioned above in mind. We have spent the last year modifying the server with respect to the second consideration: providing a single consistent interface to *all* user services. Any ARSC user goes through a procedure similar to the one listed below to use ARSC resources. Our server allows a user to perform essentially all of these steps without having to resort to any other mechanism:

1. find out about ARSC, and whether it is an appropriate site to apply for computing resources
2. investigate ARSC hardware and software resources
3. apply for an ARSC account and service units (SUs)
4. learn how to use ARSC resources
5. learn how to optimize code for ARSC resources
6. get help with using ARSC resources
7. submit results obtained with ARSC resources
8. apply for more SUs

Features

We have attempted to provide the features necessary for a user to carry out all of the required steps to use ARSC resources within his/her Web browser. We have distributed two surveys to poll our users as to which features they liked the most, and we have used document logs to determine which features they are accessing most often. Below we include a list of features that are most often used and/or appreciated by users. We also include a few that we think are particularly useful and expect to become equally popular after they have been on the server for a while.

- button bar, especially the Help, Search and Status buttons
- most frequently asked questions (FAQ) lists, with links to more detailed documents in the answers
- minimal use of graphics so that pages load quickly
- "Getting Started" sections for all platforms to enable users to log in and start working right away
- procedures documents for all procedures (storing files on CRL tapes, etc.)
- •staff directory listing and personal home pages (very popular—users claim it makes ARSC seem more personal); lynx browser available on Denali for users with vt100 terminals
- surveys to elicit user feedback
- ftp server is 'within' web server

Usage

Our server is currently logging about 10,000 document requests per week, compared to about 1500 requests per week a year ago. Some of these documents, such as our newsletters, may be of general interest, and no doubt some of this increased traffic is due to increased Web traffic in general. Other documents of a purely technical nature, however, are experiencing an increased access rate, indicating as we achieve a critical mass more of our users turn first to our web server for the information they seek, a trend indicated also by the number of users responding to our second web survey and the generally favorable remarks we received.

Future

Our users have indicated two primary areas in which they would like to see future web server development. First and foremost, they ask for more technical information, especially of the how-to variety. Secondly, they ask for more interactive services, such as online forms. Accordingly, we are developing mini user guides for a variety of hardware and software tools, and also looking at additional means of providing interactive services via CGI scripts. Our goal is to provide *all* user services via out web server. When this is accomplished, virtually all users will look to the web server first to obtain user services. We may then concentrate all development on only this mechanism, thus saving valuable staff time as well as user time.

Summary

The Web is a powerful tool for providing user services at HPC centers. It provides a single, simple interface and enables two-way communication; these features make it ideal for providing essentially all user services, not just documentation and descriptive information. The benefits of providing all user services via a single interface are increased efficiency of the user's time and the center's resources.

To provide user services effectively, the HPC center must develop a web server that is designed for *users*, and that includes *all* user services. When this is accomplished, we believe users will migrate towards the Web as a single source for obtaining essentially all of these user services. The increase in efficiency with which these services are provided benefits both the users and the center.

ARSC has designed a web server with these considerations in mind. The early results are promising; users are beginning to access the server for services other than documentation and descriptive information. We believe that users will adopt a 'Web first' approach in the next year.

For the future, we intend to continue to improve the means by which user services are provided via the Web, and to continue to add technical information to the server.