



mmGrid: Distributed Resource Management Infrastructure for Multimedia Applications

Sujoy Basu, Sameer Adhikari¹, Raj Kumar, Yong Yan,
Roland Hochmuth, Bruce E. Blaho
Mobile and Media Systems Laboratory
HP Laboratories Palo Alto
HPL-2002-350
December 18th, 2002*

E-mail: sujoy.basu@hp.com

distributed
resource
management,
grid
computing,
multimedia

We are developing mmGrid (Multimedia Grid) as an extensible middleware architecture supporting multimedia applications in a grid computing environment. Our vision is to provide support for interactive applications from the following domains: graphics, visualization, streaming media and tele-immersion. The initial deployment will be within an enterprise as a mechanism for provisioning computing resources. However the scheduling system of mmGrid will be flexible, and will allow interactive and batch jobs to use the grid-computing paradigm. This paper presents our argument for using remote display technology in this environment. We also report on the use cases we support at this point, the system architecture for mmGrid and our research directions.

* Internal Accession Date Only

Approved for External Publication

© Copyright Copyright IEEE

¹ Work done during internship at HP Labs. Currently at College of Computing, Georgia Tech, Atlanta, GA 30332
To be published in and presented at the 17th International Parallel and Distributed Processing Symposium, 22-26 April 2003, Nice, France

mmGrid: Distributed Resource Management Infrastructure for Multimedia Applications

Sujoy Basu,* Sameer Adhikari,† Raj Kumar and Yong Yan
Hewlett-Packard Laboratories
Hewlett-Packard Company
1501 Page Mill Rd.
Palo Alto, CA 94304

Roland Hochmuth and Bruce E. Blaho
Workstation Global Business Unit
Hewlett-Packard Company
3404 E Harmony Rd.
Fort Collins, CO 80528

Abstract

We are developing mmGrid (Multimedia Grid) as an extensible middleware architecture supporting multimedia applications in a grid computing environment. Our vision is to provide support for interactive applications from the following domains: graphics, visualization, streaming media and tele-immersion. The initial deployment will be within an enterprise as a mechanism for provisioning computing resources. However the scheduling system of mmGrid will be flexible, and will allow interactive and batch jobs to use the grid-computing paradigm. This paper presents our argument for using remote display technology in this environment. We also report on the use cases we support at this point, the system architecture for mmGrid and our research directions.

1 Introduction

We are developing mmGrid (Multimedia Grid) as an extensible architecture supporting multimedia applications in a grid computing [4, 3] environment. The applications we are looking at currently belong to the following domains: graphics, visualization and multimedia streaming. In future, we might consider collaborative virtual environments (tele-immersion). In this paper, we describe aspects of the middleware in mmGrid that have already been developed. mmGrid is being designed for use within an enterprise as a means for provisioning resources. The middleware has to be easily manageable so that the barrier to entry in existing IT environments is low. We also require scalability with the size of the resource pool managed and platform independence so that it can be easily deployed in heterogeneous

environments. In fact one of the anticipated early uses of mmGrid will be to provision high-end Unix workstations running mechanical CAD applications such as crash simulations. The user is expected to access the workstation from his regular desktop computer and have full access to the workstation's desktop. He will then interactively launch the application. Fault-tolerance is also important due to the high probability of failure of some resource in a large resource pool. We also require interoperability with web services [6]. This is due to the current standardization of the Open Grid Services Architecture (OGSA) [3] on the web services paradigm. An installation of mmGrid could use this paradigm to dynamically add resources to the pool it manages from an external resource broker. This might allow a processor-design team to own an mmGrid installation with just adequate capacity for regular workloads, and buy additional compute power during peak periods near the end of design cycles. We also anticipate some applications running on mmGrid to be Grid-enabled. These applications might use Grid protocols to dynamically request additional compute resources during compute-intensive phases of their execution.

Based on these requirements, we have designed the overall system architecture of mmGrid middleware. Software agents will run on all compute resources controlled by mmGrid. The agent running on the provisioned workstation will be instructed by mmGrid server to export its desktop to the requestor's desktop using remote display technologies. Before the user is given control of the exported desktop, she is assumed to have authenticated herself using mmGrid's authentication sub-system. Pluggable authentication modules (PAM) will allow authentication to be done using existing security infrastructure, or by utilizing the directory service that needs to be installed as part of mmGrid. After exporting the desktop, the agent will continue to monitor usage of the workstation, and periodically send notification messages to the mmGrid server. When the user logs out

* contact at sujoy.basu@hp.com

† Work done during internship at HP Labs. Current affiliation: College of Computing, Georgia Tech, Atlanta, GA 30332

of the workstation, the agent notifies the mmGrid server, which returns the workstation to the available pool of resources. If periodic messages do not arrive at the mmGrid server from an agent, that resource is marked as unavailable due to failure. The information maintained by mmGrid server is divided into two categories. More persistent information such as user and workstation profiles are stored in a LDAP directory for fast lookup. Information requiring high throughput of writes, such as events in the system that need to be logged and monitored information sent by the agents running on all the compute resources are stored in a database.

The interface to the mmGrid server will be important for both users and system administrators. We would like to give users the ability to take workstations they own and add them to the resource pool managed by mmGrid if they so desire. They also use mmGrid to authenticate themselves, request resources and connect to the resources. System administrators will use the interface to add new users to the system, monitor the status of the system and be notified of the location of faults in the system. To allow these differences in use cases, we use the user profile stored in the directory to distinguish normal users from administrators. The latter are presented with a superset of options allowed for normal users. The interface to users and administrators is provided through web browsers, so that they can connect from any computer without installing additional software. The presentation logic is decoupled from mmGrid server. It is handled by a mmGrid client, that runs as a servlet in a Tomcat container on a web server. The mmGrid client interacts with the mmGrid server using web services paradigm.

Since it is possible for users to request interactive access to a workstation, some form of remote display technology must be utilized. Several issues have to be weighed before deciding the solution to use. First, we must consider the fact that the workstations being provisioned can be running different operating systems: Microsoft Windows, Linux or HP-UX. Secondly applications or graphics libraries cannot be modified to support our needs. This rules out choices such as X protocol [9] which is not used by applications across all the operating systems of interest. Hence we chose to use VNC [10] which allows a frame buffer of one computer to be displayed on another. Another argument in favor of sending the pixels rather than the geometric models is the fact that the complexity of the geometric models to be rendered has been rising rapidly with every passing year. However the resolution of computer displays improves relatively less rapidly. So with increased network bandwidth available, it has become viable to send the compressed frame buffer contents from the provisioned workstation to the user computer. This also has the advantage that the user can connect to the workstation from a thin client that does not have cutting-edge graphics rendering hardware.

VNC serves our purpose of providing remote access to the desktop for graphics applications. However streaming media applications pose different set of problems. Imagine some MPEG content being streamed to a media player on the provisioned workstation. The frame rate may not be adequate for the media content viewed over VNC. Besides, the audio needs to be sent too. Capturing the audio at a low-level driver within the OS might be possible. But synchronizing it with the frame buffer will be a problem. We might be able to capture the MPEG stream just as it is passed by the media player to a decoder. However the need to support multiple media formats and operating systems makes it a formidable problem from the software engineering standpoint. Hence our current approach is to use the agent running on the remote workstation to send the URL of the content to the agent running on the user's computer, which then invokes the media player locally to play the content. Our approach requires the correct media player to be present on the user's computer. We expect the content accessed to be present somewhere in the intranet or Internet, rather than on the local disk of the remote workstation. For content stored on the remote workstation, additionally a streaming server has to be running on that workstation. The media player on the user's computer will then contact this server.

The main focus of our research efforts in the near future will be on scheduling. The mmGrid server will have to service requests from users for reservation of workstations. Often advance reservations will be requested. Besides, to be part of any grid-computing environment, a mmGrid installation should ideally support both advance reservation and coallocation of resources. Looking further ahead, we would like to make the information and security services available from mmGrid compliant to the OGSA protocols so that interoperability is possible.

In this paper we discuss the application requirements in Section 2. Then in Section 3, we present our system architecture. Section 4 takes the reader through a typical usage scenario using a sequence diagram. In Section 5, we present related work. Finally our conclusions are presented in Section 6.

2 Application Requirements

Since mmGrid is focusing on graphics, visualization and streaming media applications and, in future, tele-immersion, advance reservation of network bandwidth is a key requirement for good performance. In addition to that, ability to submit batch jobs, as well as the ability to make advance reservation of a workstation for interactive applications are basic requirements. We are addressing the basic requirements by designing a scheduler for our mmGrid middleware. The scheduler manages the scheduling of interactive and batch jobs within the cluster. It is a local scheduler

in the Grid context. For applications to be able to use the grid-computing paradigm, we will support, in future, the translation of the user's credentials stored in the mmGrid authentication sub-system into GSI credentials required for accessing other grid resources. We will also allow interaction with higher-level Grid scheduler and other services, such as advance reservation of network bandwidth.

Although our work is focused on building the infrastructure that allows resource provisioning within a cluster and across Grids, we need to provide enabling technologies that make it possible for graphics, visualization, streaming media and tele-immersion applications possible in mmGrid. Our initial focus has been on graphics and visualization applications. The target market segments are design teams running mechanical and engineering CAD applications. A secondary focus currently is on streaming media applications.

The graphics and visualization applications in our target market segments have been witnessing an exponential growth in the complexity of the models that have to be rendered on the screen. With such a fast growth in the amount of the data, it has become clear that only high-end graphics workstations will have the capability to render such complex scenes, particularly when designers interact with the data set, requiring several frames to be rendered per second. The designer will connect to this workstation from a desktop computer which may or may not have high-end graphics hardware for rendering scenes at this frame rate. This suggests that instead of sending raw geometric data for rendering on the client desktop, it is better to send pixels contained in the frame buffer of the remote workstation after rendering. This will allow us to easily support a heterogeneous client population.

The second argument for sending the frame buffer contents rather than the geometric data from the remote workstation to the client desktop is based on the current evolution in technologies that are required. While the display resolution doubles in about 5 years, the amount of RAM available typically doubles every 12 months. This leads to a widening gap between display resolution and memory. With increased availability of memory and higher processing speed, the applications will tend to have larger and more complex geometric models for more realistic rendering. As we mentioned earlier, the complexity of these models tend to increase exponentially with time. So sending the geometric data to the client for rendering will always remain challenging. On the other hand, with the slower growth in display resolution, it will become possible to apply JPEG and other compression algorithms to the frame buffer contents and send it to the client desktop at faster and faster frame rates. The increase in frame rate will be due to processing power doubling every 18 months (Moore's Law) and communication bandwidth doubling every 6 months. The latter

is based on Gilder's Law, which asserts that capacity of networking equipment grows at least three times as fast as CPU speed.

An exception to the above argument has to be made for viewing streaming media content. Specifically, while using the remote workstation, the user might click on streaming media content and expect video and audio as if they are playing on the local machine. In this case, the approach of compressing the frame buffer may not be adequate for delivering the frame rate required. Also, capturing video and audio separately on the remote workstation leads to loss of synchronization information between the video and audio streams. Hence our current approach is to require the user to open the content with a program that passes the URL to the software agent running on the remote workstation. That results in communication with the agent running on the local machine. The latter launches the media player on the local machine to play the content. There are limitations of this approach. If the content is linked from a web page, and the URL of the streaming media cannot be passed to the agent, the web page will have to be accessed on the local machine. Our agent-based solution will work only if the URL of the streaming media is visible, such as when it is bookmarked during a previous access, or the web page provides a hyperlink with the full URL. Another limitation of this approach is that content present on the remote computer's disk requires a streaming server to be running on the remote computer.

3 System Architecture

The mmGrid system consists of the components shown in Figure 1. The components are: HTTP client i.e. browser running on user's desktop, HTTP server and mmGrid client, mmGrid server, authentication service, scheduling service, directory service and database service and mmGrid agents running on all the resources managed by mmGrid. Other services will be added in future.

3.1 HTTP Client

The HTTP client i.e. a browser provides the entry point for the user or administrator into the system. All possible user interactions are provided through web pages the user or administrator downloads as required.

3.2 HTTP Server

The HTTP server provides the web pages as required by the client. We use the Jakarta Tomcat servlet engine as the hosting environment with the server. The mmGrid client runs as a servlet and can invoke the mmGrid server using the web services paradigm.

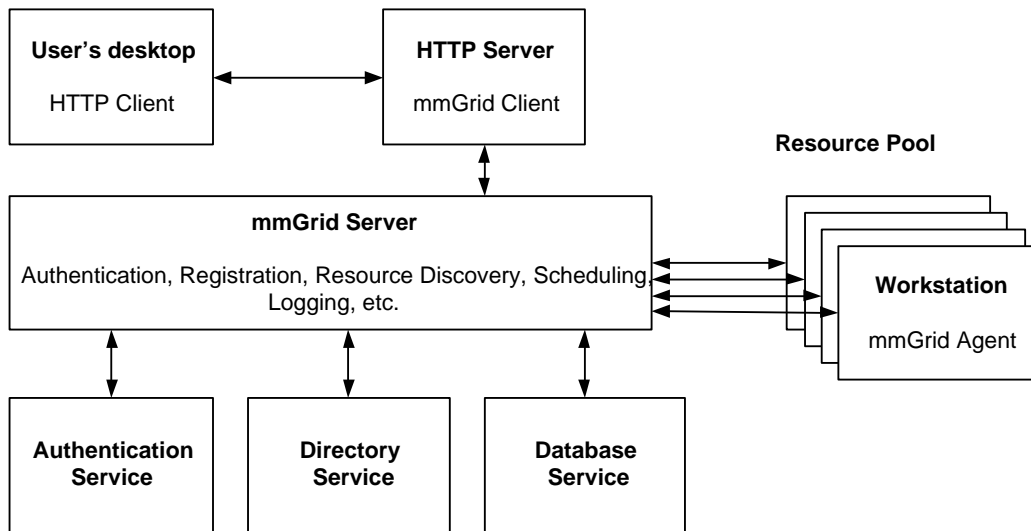


Figure 1. System Architecture: Each rectangle in this diagram represents a separate computer, although some of the services could potentially run on the same computer, such as the authentication and directory services

3.3 mmGrid Server

The mmGrid server uses the web services paradigm, and a secure communication channel can be used between the mmGrid client and server. The server provides several functions by interfacing to different services that are part of mmGrid. Before any service can be obtained on behalf of a user, authentication must be done. More details about it are provided in a following section. Following that, the user might want to search for resources matching certain criteria or browse the resources available in the local mmGrid installation. The server interfaces to the directory service to search for resources matching the requested criteria. The server might also have to communicate with the database service to obtain more dynamic information such as system load over the past few days on the selected computers. The latter information is needed when a user requests interactive use of a workstation for a certain period. The mmGrid server will have to identify one that best suits the user's needs. For batch job requests submitted by the user, the mmGrid server will forward the request to the scheduler we are currently designing. It will schedule the job on the local mmGrid cluster if possible. If needed, it will interact with a higher-level scheduler to request resources from other clusters that participate in the grid.

The mmGrid server performs other functions such as user registration which makes a person eligible to use the resources of mmGrid. Furthermore, users can register their own workstations. After registration, a workstation can be checked in and out of the pool managed by mmGrid accord-

ing to the user's convenience. Checking in implies that the user will not be using the workstation for the checked-in period. The workstation can be used by mmGrid to schedule other tasks. Another function performed by the mmGrid server is logging events of interest by sending them to the database service. The logs can be analyzed and the usage patterns detected can lead to improvements in the deployment of the system.

3.4 Resource Pool

Each member of the resource pool, such as a workstation, will have a mmGrid agent running on it. The agent receives requests from the mmGrid server, such as a request to launch the VNC server on behalf of the user, and connect to the user's workstation. The agent also monitors use of the workstation, and periodically sends status information to the mmGrid server. When the user disconnects from the workstation, the agent informs the mmGrid server, which returns the workstation to the pool of available resources.

3.5 Authentication Service

The authentication service grants an administrator or user access to mmGrid. Our intention is to support existing authentication mechanisms already in place in enterprise environments. Hence we decided to implement the authentication service using the Java Authentication and Authorization Service (JAAS) framework, which allows us to use Pluggable Authentication Modules (PAM), known as login

modules in the JAAS context. Our initial deployment is with a login module that allows authentication against the Netscape Enterprise Directory. The latter is commonly deployed in corporate intranets for single sign-on. Our choice of JAAS allows us to support Kerberos, Sun Microsystem's NIS (also known as Sun Yellow Pages) and a host of other authentication mechanisms.

The Enterprise Directory used for authentication is separate from the directory service we show for our system architecture in Figure 1. The latter stores information such as user profile and workstation profile. In fact, irrespective of whether authentication is done initially against Enterprise Directory, Kerberos or Yellow Pages, in the second phase of our authentication, the user profile is accessed from our directory service. This allows the mmGrid server to designate the authenticated individual as normal user or administrator. Access control at a finer granularity than this can also be easily implemented by adding the groups a user is member of to his user profile. This is already part of our schema for user profile. Similarly if there is need to limit the access to certain workstations to certain groups, the workstation profile can indicate that.

3.6 Directory Service

The directory service maintains relatively static information. Currently that information relates to users, compute and storage resources and application profiles. The main aim of the user profile is to uniquely identify a user, his or her group memberships and privileges and account number to be charged for his or her usage of resources in the grid. The user profile can also maintain preferences of the user, such as the hardware platform and applications typically requested by the user. Compute and storage resources are important since they allow users to search for resources that meet their application's requirements. Thus a user might specify that he needs a workstation running HP-UX with processor frequency at least 1 GHz and with available disk space exceeding 20 GB. This will be sent as a search string to the directory service. Application profiles are used by system administrators to keep track of applications installed on mmGrid resources. Users can also specify that a certain application be available on the resource allocated to them.

3.7 Database Service

The database service maintains frequently changing information. The active users' table maintains the list of users logged in the system, along with other information such as the IP address from which they connected and the timestamp of the initial connection. The table of free resource maintains the list of resources available for allocation to the users. The table of allocated resources maintains the list of

resources currently in use and the assignment of users to these machines. The tables for free resources and allocated resources jointly list the complete set of resources managed by this mmGrid installation. The event log table maintains a chronological list of all the events of interest in the system. Each entry will list the event name, timestamp, user ID, IP address of the user and other relevant information. There is an event for every use case listed in Tables 1 and 2. Logging the events allows us to analyze the system. System performance as a function of various factors such as user activity, resource usage etc. can be studied. Logging also aids in debugging when the system behaves erroneously.

3.8 Agents

A software agent runs on every resource managed by mmGrid. The mmGrid server sends messages to the agents to perform actions required by the user. Currently the most important action is to launch the VNC server and connect the remote workstation to the VNC client running on the user's local machine. The agent can also monitor the system it is running on. This can be to account for the time during which the user utilized the system. The agent can also ensure that the quality of service criteria used in selecting the system initially is maintained over time. In case the criteria gets violated the agent can aid in taking corrective action.

4 Usage Scenario

mmGrid distinguishes between normal users and administrators. The distinction is based on the actions that each class is allowed to take. Users access the system solely for computing functions. Administrators access the system for system management and computing tasks. A user is able to perform the actions listed in Table 1 only. An administrator is able to perform all the actions listed in both Tables 1 and 2.

The typical usage scenario for a normal user who logs into the mmGrid system, and then allocates a workstation for his interactive use is shown in Figure 2. Steps A through J constitute the first phase in which the user logs into the system. The login page returned in Step B is shown in Figure 3. Notice that the mmGrid server needs to do two things in parallel. In step E, it sends the authentication request to the authentication system. It also needs to query the directory service in step F to determine whether the user seeking access is a normal user or administrator. In the scenario presented in Figure 2, it turns out to be normal user. Hence the HTML document returned in step J is the main menu for normal users. If he were an administrator, he would have got a superset of the user's main menu. This latter case is shown in Figure 4. It is also worth noting that Figure 2 is a

simplification. Thus the authentication message sent in step E can result in several messages being exchanged depending on the authentication protocol before the final success notification is sent in step G.

Steps L through U constitute the second phase in which the user is allocated a workstation for his use. Notice that in this case the mmGrid server first queries the directory service using the search filter specified by the user. For the matching resources returned by the directory service in step Q, the mmGrid server can get additional information from the database service in step R. The additional information may include current state of the resource as well as historical information. Based on the information, the mmGrid server allocates a resource. Steps T and U result in that allocation being communicated to the user.

Finally in step V, the user connects to the allocated resource using the VNC client. This step can be automated if user agent software is installed on the user's computer. In that case the mmGrid server can instruct this agent to launch the VNC client, and listen for a connection. The mmGrid server can then instruct the agent on the allocated resource to launch the VNC server, and also specify the IP address and port number at which the VNC client is listening for the incoming connection.

5 Related Work

The Globus Toolkit [7] has been widely deployed at universities and national labs across the United States and Europe for building grids. Several companies, including HP, IBM and Platform Computing, have supported the Globus toolkit. Companies like AVAKI, Entropia, Sun Microsystems and United Devices also sell grid computing software. These grids allow challenging problems in the life sciences, financial and other business sectors to be solved by harnessing the compute and storage capacities across a diverse set of resources, ranging from the personal computers idling at users' desks to the supercomputers available at a few sites. We believe that the development of mmGrid will be orthogonal to these products. After accessing a workstation through mmGrid for interactive use, the user might do interactive work for a while and then need to launch a compute-intensive simulation. This task can be submitted to the scheduler for mmGrid that we are currently designing. Here we can leverage research projects such as Condor-G [5], as well as the commercial products. Other projects which have focused on computing infrastructure with a portal for easy access to scientific applications include Legion [2] and PUNCH [8].

MmGrid uses the VNC server and clients for remote display of the allocated workstation's desktop to the user's desktop. There are many existing products in this space. There are companies that sell enhancements to VNC. There

are other products such as Citrix MetaFrame from Citrix Systems, Inc. that have similar approach. There are thin clients such as the Sun Ray from Sun Microsystems. We have focused in this work on resource provisioning from workstation clusters, idle computers checked in by their owners into the mmGrid resource pool, and external resources available using the grid computing paradigm. VNC server and client are just the means to achieve our primary goal of providing remote access. We are also not trying to build portal servers that allow users to access all of their applications and data through a web interface. Rather our goal is to allocate workstations for exclusive use by the mmGrid user.

6 Conclusions

We have described mmGrid, an extensible architecture supporting multimedia applications in a grid computing environment. Our vision is to provide support for graphics, visualization, streaming media and tele-immersion applications that require interactive use of computer systems. Towards that end, we have presented the rationale for developing remote display technology. We also intend to support both interactive and batch jobs that require grid-computing. In the immediate future, our efforts will be focused on the integration of our authentication service with Grid Security Infrastructure [1], and the design of our scheduler.

Action	Description
Login	Get access to mmGrid services
Logout	Logout from mmGrid
Check-in Resource	Add a resource to mmGrid's resource pool
Check-out Resource	Remove a resource from mmGrid's resource pool
Allocate Resource	Assign a resource from mmGrid's resource pool to a user
Free Resource	Return an allocated resource to the free state
Browse	Browse the resources and applications available
Search	Allow a user to search for a resource meeting his criteria

Table 1. List of user functions

References

- [1] R. Butler, V. Welch, D. Engert, I. Foster, S. Tuecke, J. Volmer, and C. Kesselman. A national-scale authentication infrastructure. *IEEE Computer*, 33(12):60–66, 2000.

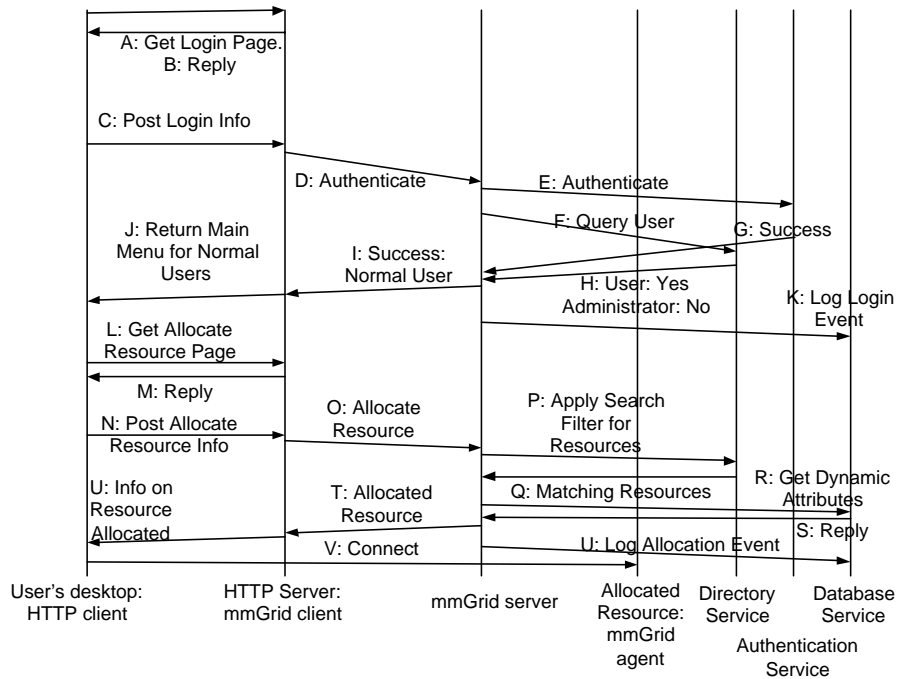


Figure 2. Sequence diagram for user authenticating with mmGrid, requesting allocation of a workstation and then connecting to the workstation



Figure 3. Login Screen for mmGrid

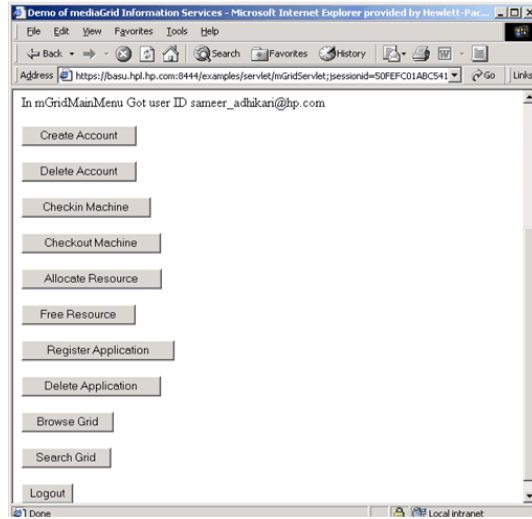


Figure 4. Main Menu for mmGrid administrators

Action	Description
Create Account	Allow user to access mmGrid
Delete Account	Revoke user's access to mmGrid
Register Application	Register an application with mmGrid
Delete Application	Delete an application from mmGrid

Table 2. List of extra administrator functions

and Applications; special issue on High Performance Distributed Computing, 2(2):153–164, sep 1999.

- [9] A. Nye, editor. *X Protocol Reference Manual*, volume 0. O'Reilly, 1994.
- [10] T. Richardson, Q. Stafford-Fraser, K. Wood, and A. Hopper. Virtual network computing. *IEEE Internet Computing*, 2(1):33–38, Jan/Feb 1998.

- [2] A. S. G. et. al. Legion: The next logical step towards the world-wide virtual computer. *Communications of the ACM*, 40(1), jan 1997.
- [3] I. Foster, C. Kesselman, J. Nick, and S. Tuecke. The physiology of the grid: An open grid services architecture for distributed systems integration, June 22 2002. Available at <http://www.globus.org/research/papers.html#OGSA>.
- [4] I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. *International Journal of Super-computer Applications*, 15(3), 2001. Available at <http://www.globus.org/research/papers.html#anatomy>.
- [5] J. Frey, T. Tannenbaum, M. Livny, I. Foster, and S. Tuecke. Condor-g: A computation management agent for multi-institutional grids. In *Proceedings of the Tenth International Symposium on High Performance Distributed Computing (HPDC-10)*. IEEE Press, aug 2001.
- [6] G. Glass. *Web Services: Building Blocks for Distributed Systems*. Prentice Hall PTR, 2002. ISBN 013066256-9.
- [7] Globus Toolkit. <http://www.globus.org>.
- [8] N. H. Kapadia and J. A. B. Fortes. Punch: An architecture for web-enabled wide-area network-computing. *Cluster Computing: The Journal of Networks, Software Tools*