ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 1, Issue 8, October 2012

GRID ARCHITECTURES FOR e-LEARNING SYSTEM

Sanjeev Narayan Bal

Dept.of Comp.Sc.
TACT, Bhubaneswar,Odisha,India

***Abstract*--E-learning has been a topic of increasing interest in recent years, due to the fact that increased scheduling flexibility as well as tool support can now be offered at a widely affordable level. As a result, many e-learning platforms and systems have been developed and commercialized; these are based on client-server, peer-to-peer, or, more recently. An e-learning system can make an enterprise more competitive by increasing the knowledge of its employees. E-learning has been shown to have impressive potential in e-commerce. At present, most e-learning environment architectures use single computers or servers as their structural foundations. As soon as their work loads increase, their software and hardware must be updated or renewed. The collaborative e-Learning services can be categorized as generalized services or need specific services. While it is easy to construct need specific services with the help of sophisticated tools available today, it is increasingly complex to build generalized collaborative e-learning services. In the design and implementation of generalized services, the major ones are e-Resources discovery, scheduling, security and overall management and administration. An organization may implement an e-learning system according to their requirements.**

***Keywords:* e-learning. grid computing, data grid, grid portal, Grid-based Resource Management. e-Learning Services, Collaborative e-Learning Environments ,e-Resource discovery, security and management.**

I. Introduction

Grid Computing delivers on the potential in the growth and abundance of network connected systems and bandwidth: computation, collaboration and communication over the Advanced Web. At the heart of Grid Computing is a computing infrastructure that provides dependable, consistent, pervasive and inexpensive access to computational capabilities. By pooling federated assets into a virtual system, a grid provides a single point of access to powerful distributed resources.

Researchers working to solve many of the most difficult scientific problems have long understood the potential of such shared distributed computing systems. Development teams focused on technical products, like semiconductors, are using Grid Computing to achieve higher throughput. Likewise, the business community is beginning to recognize the importance of distributed systems in applications such as data mining and economic modeling.

With a grid, networked resources -- desktops, servers, storage, databases, and even scientific instruments -- can be combined to deploy massive computing power wherever and whenever it is needed most. Users can find resources quickly, use them efficiently, and scale them seamlessly.

II. The Grid Concept

The term 'grid' is variously used to describe a number of different, but related, ideas, including utility computing concepts, grid technologies, and grid standards. In this paper the term 'Grid is used in the widest sense to describe the ability to pool and share Information Technology (IT) resources in a global environment in a manner which achieves seamless, secure, transparent, simple access to a vast collection of many different types of hardware and software resources, (including compute nodes, software codes, data repositories, storage devices, graphics and terminal devices and instrumentation and equipment), through non-dedicated wide area networks, to deliver customized resources to specific applications.

At the most general level Grid is independent of any specific standard or technology. Any practical grid is realized through specific distributed computing technologies and standards that can support the necessary interoperability. Today, there are no universally agreed grid standards, but there are freely available, open source and proprietary grid technologies that implement emerging standards recommendations. Separate web services standards are also emerging which have many grid-like capabilities. Indeed grids are already being built by integrating and enhancing web standards technology.

A. Practical Realizations

Practical grids are generally described in terms of layers (see Fig 1). The lowest layers (the 'platform') comprise the hardware resources, including computers, networks, databases, instruments, and interface devices. These devices, which will be geographically distributed, may present their data in very different formats, are likely to have different qualities of service (e.g. communication speeds, bandwidth) and are likely to utilize different operating systems and processor architectures. A key concept is that the hardware resources can change over time - some may be withdrawn, upgraded or replaced by newer models, others may change their performance to adapt to local conditions - for example restrictions in the available communications bandwidth.

The middle layers (sometimes referred to as 'middleware') provide a set of software functions that 'buffer' the user from administrative tasks associated with access to the disparate resources. These functions are made available as services and some provide a 'jacket' around the hardware interfaces, such that the different hardware platforms present a unified interface to different applications. Other functions manage the underlying fabric, such as identification and scheduling of resources in a secure and auditable way. The middle layer also provides the ability to make frequently used patterns of functions available as a composed higher-level service using workflow techniques.

The highest layers contain the user 'application services'. Pilot projects have already been carried out in user application areas, such as life sciences (e.g. computational biology, genomics), engineering (e.g. simulation and modeling, just in time maintenance) and healthcare (e.g. diagnosis, telematics). These services could include horizontal functions such as workflow (the linkage of multiple services into a single service), web portals, data visualization and the language/semantic concepts appropriate to different application sectors.

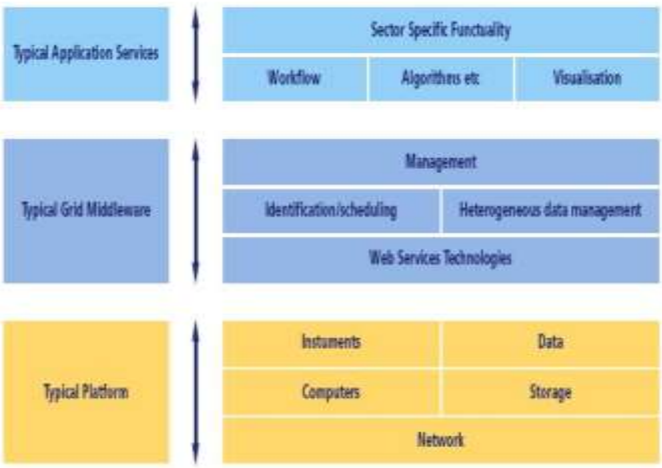


Fig. 1 Simplified Grid Architecture

B. Grid Developments and Deployment

A key issue facing the industry is the timing and mode of deployment of Grid technology to ensure that it is sufficiently mature to deliver the expected business benefits. There is emerging evidence that the technology can achieve significant operational benefits (e.g. in telemedicine), improvements in performance (e.g. in climate modeling and genomics) and a significant reduction in costs. Nevertheless, current grid technologies are not yet viewed as sufficiently mature for industry scale use, and remain largely unproven in terms of security, reliability, scalability, and performance.

Short term--For the short term (within the next two years), Grid is most likely to be introduced into large organizations as internal 'Enterprise grids', i.e. built behind firewalls and used within a limited trust domain, perhaps with controlled links to external grids. A good analogy would be the adoption into business of the Internet, where the first step was often the roll out of a secure internal company 'Intranet', with a gradual extension of capabilities (and hence opportunity for misuse) towards fully ubiquitous Internet access. Centralized management is expected to be the only way to guarantee qualities of service. Typically users of this early technology will be expecting to achieve IT cost reduction, increased efficiency, some innovation and flexibility in business processes. At the same time the distinction between web services and grid services is expected to disappear, with the capabilities of one merging into the other and the interoperability between the two standards being taken for granted.

Medium Term--In the mid term (say a five year timeframe) expect to see wider adoption – largely for resource visualization and mass access. The technology will be particularly appropriate for applications that utilize broadband and mobile/air interfaces, such as on-line gaming, 'visualization-on-demand' and applied industrial research. The emphasis will move from use within a single organization to use across organizational domains and within Virtual Organizations, requiring issues such as ownership, management and accounting to be handled within trusted partnerships. There will be a shift in value from provision of computer power to provision of information and knowledge. At the same time open standards based tooling for building service oriented applications are likely to emerge and Grid technology will start to be incorporated into off-the-shelf products. This will lead to standard consumer access to virtualized compute and data resources, enabling a whole new range of consumer services to be delivered.

Long term--In the longer term, Grid is likely to become a prerequisite for business success -central to business processes, new types of service, and a central component of product development and customer solutions. A key business change will be the establishment of trusted service providers, probably acting on a global scale and disrupting the current supply chains and regulatory environments.

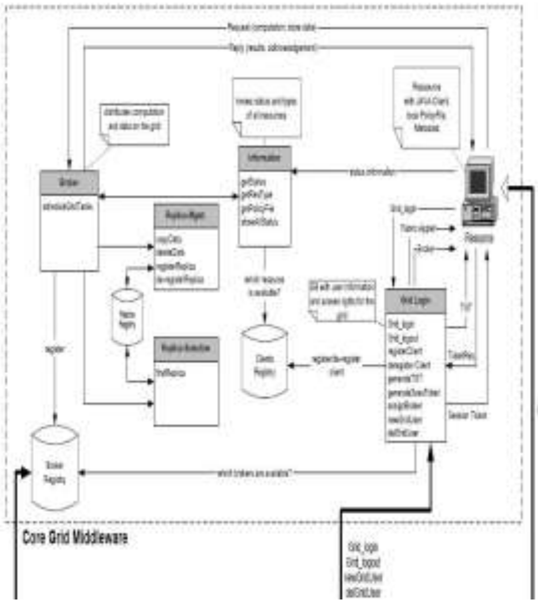
III. E-Learning Grids

A. Grid Architecture—I

There are many conceivable applications for e-learning grids. Medicine students could use photo-realistic visualizations of a complex model of the human body to prepare for practical exercises. Such visualizations, computed in real-time, could improve the understanding of the three-dimensional locations

ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 1, Issue 8, October 2012

of bones, muscles, or organs. Students should be able to rotate and zoom into the model and get additional information by clicking on each element of the model. With more advanced functionality such as virtual surgery, students could be provided with the possibility to grab, deform, and cut model elements (e.g.organs) with the click of a mouse. In biology courses the ability of grids to integrate heterogeneous resources could be used to integrate an electron microscope into the grid. We mention that the technical feasibility of this approach has already been demonstrated in the TeleScience project. However, this project could be widely extended to integrate the controls and output of the electron microscope into a learning environment so that students can be assigned tasks or read subject-related texts while operating the microscope. Similarly, in engineering courses complex simulations, e.g.in a wind channel, can be made accessible for each student by using grids.



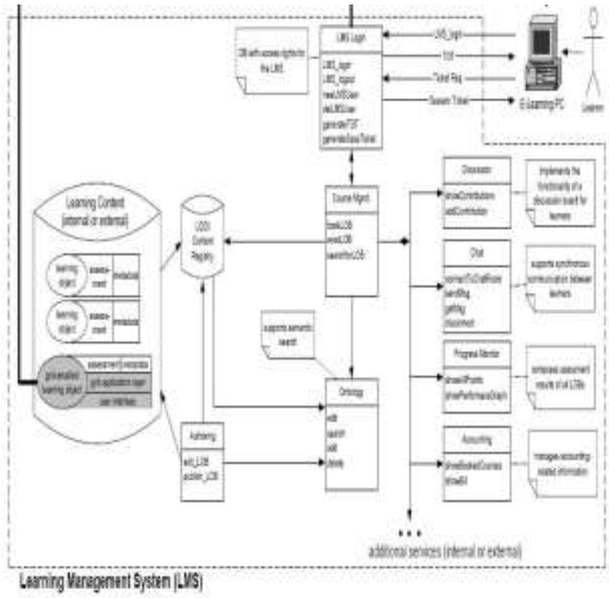


Fig. 2 Architecture of an E-Learning Grid.

We outline an architecture for e-learning grids. To demonstrate the technical feasibility, the architecture will be kept as simple as possible. It contains a Learning Management System (LMS) as well as grid middleware, which are both based on Web services and grid services, respectively (Figure 2). In this figure, grid and Web services are depicted as rectangles containing a name as well as the most important operations. Note that grid services (with grey name fields) can easily be distinguished from Web services. The LMS interacts transparently with the grid middleware so that a learner is not aware of the grid. Furthermore, the architecture is designed in such a way that a learner only needs a Java-enabled Internet browser to use both the LMS and the grid. The architecture shown in Figure 2.

Core Grid Middleware-- The grid middleware of an e-learning grid implements the various layers shown above in Figure 3. The fabric layer is implemented as a Java applet, which provides uniform interfaces to all resources in the grid. For the time being, we assume for simplicity that there are only computers and no specialized instruments like electron microscopes in the grid. We will explain possible extensions later. Furthermore, locally available policy files specify usage restrictions for computers in the grid (e.g.. maximal CPU load, usage hours, etc.): finally, meta-data describe each resource type. The fabric layer applet has to be started on each computer that participates in sharing computational capacity and storage space in the grid. This can be done while a user accesses a Web page with his or her Web browser to authenticate in the grid.

ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 1, Issue 8, October 2012

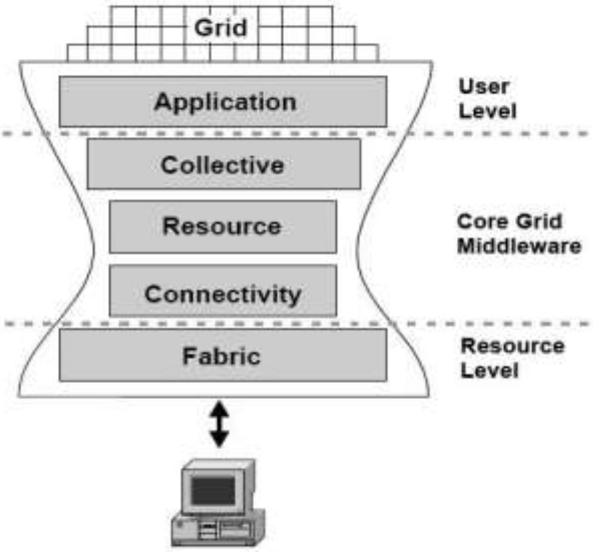


Fig. 3 Layers in a Grid Middleware.

The connectivity layer consists of a grid login service. This service needs to have access to a database in which user information and access rights are stored together with a hash value of the password. The service can create new users with a newGridUser operation and delete users with corresponding delGridUser operation. When a grid user wants to enter the grid, an operation Grid_login is called, which checks the user's login name and password against the values stored in the database. If the login was successful, the computer is registered with registerClient in a ClientsRegistry. the latter of which contains a list of all resources currently available in the grid. Furthermore, a broker is assigned which can handle requests to distribute computation or data across other computers in the grid. When a client logs out. the ClientsRegistry is updated through a deregisterClient operation. It uses a symmetric cryptographic algorithm and requires that each entity in the grid (i.e.. grid service or computer) must have a specific key which must be known to the grid login service. For all computers of a grid, the hashvalue of the user password could be used as a key. while grid services have a key generated and registered at the login service by the respective developers. The idea of the authentication algorithm, whose details are beyond the scope of this paper, is that a computer first requests a ticket-granting ticket (TGT). If a TGT is received, the authentication was successful and further session tickets can be requested for the communication with any other grid service. Before a computer can access the operation of a grid service, a valid session ticket must be presented- which is checked by the respective service. This procedure is executed every time an operation of a grid service is accessed- so that an additional graphical representation is omitted in Fig.2.

The resource layer contains an information service which is aware of the status and type of all resources in the grid. By accessing the clients registry it firstly determines which computers are available. It then requests from each computer some status information (e.g.. CPU load, unused storage space) with a getStatus operation, the resource type with getResType and the usage restrictions with getPolicyFile. The list with the collected information can be accessed with showAllStatus and is used to schedule computations or distribute data in the collective layer.

In the collective layer there is a broker, replica management and replica selection service. For simplicity, we assume that there is only one copy of each service. The broker implements a grid scheduling algorithm [1] and is responsible for distributing computations and data across the grid. The broker has to register in a BrokerRegistry. so that the grid login service and grid applications in the application layer can find it. For the distribution of data it uses the replica management and replica selection service, which implement the functionality that is typical of data grids. We assume here that the data to be replicated is read-only. The replica management service can create replicas of data on computers with the operation copyData. When data is replicated, an entry is added to the ReplicaRegistry with the exact information which pans of data were replicated on which machines. When replicated data is deleted with deleteData the corresponding entry in the registry has to be erased with de-registerReplica. With the operation findReplica of the replica selection service existing replicas can be found when data is accessed.

All assumptions made so far can be relaxed, which leads to a loss of simplicity of the architecture. The fabric layer applet could be extended to support instruments, like electron microscopes. In this case, the information and broker service would have to be adapted since instruments cannot execute computations or store data. Furthermore, there could be more than one service of each type. For example, several grid login services could be responsible for disjoint authentication domains. Clients could look up in a LoginRegistry which login service is responsible for the domain they are located in. A problem arises when clients need to access resources of different authentication domains, which requires cooperation between grid login services. It is also possible to have information, broker, and replica services for each authentication domain to increase the efficiency in large grids. Similarly to the problems encountered at the login services, a cooperation between all information, broker, and replica services of each domain could be necessary.

Learning Management System (LMS)--An LMS generally coordinates all learning-related activities. We assume that the entire LMS functionality including the learning contents are implemented as Web services. A learner who typically uses a PC for a learning session interacts directly only with the LMS and is not aware of a grid. The LMS offers both content which makes use of the grid as well as content that does not need grid functionality. The integration between LMS and grid will be described in the next subsection. All Web services of the LMS are accessed via Web pages, so that the learner only needs a Web browser to utilize the LMS.

In a first step the learner has to authenticate in the LMS, which is done by an LMS login service. This service is similar to the grid login service, i.e.. it draws on a database with access rights and uses the same authentication mechanism for the LMS. When the learner is logged in and authenticated- he or she can access a Web page for course management, the functionality of which is implemented in a course management service. The learner can look for suitable courses with a searchLOB operation, which searches for learning objects in a ContentRegistry. The bookLOB operation is called to enroll for a course. A class can be attended by calling the exec LOB operation.

An ontology service supports the semantic search for courses. It basically contains an ontology defining semantic relationships between Web services that provide learning objects. Thus, for a search term like "programming" it possible to obtain results like "C++," "Java" or "Prolog." The ontology service provides operations to add. delete, edit, or search for entries in the ontology. Next, the authoring service provides an environment to create, edit, and publish e-learaing content in the ContentRegistry. so that they can be found by the LMS. In addition, entries can be added to the ontology for the semantic retrieval of content.

The web services which provide e-learning con fern consist of three fundamental components. The first part is a learning object, which is typically a lesson (e.g.. in HTML or XML format) or a course consisting of several learning objects. The assessment part defines online-tests so that students or teachers can check whether a lesson is well-understood and the material mastered. The last part contains meta-data for search engines that describes the content in a standardized way. The functionality of the grid is used with grid learning objects, which also integrate a grid application layer and a user interface and will be described in the next subsection.

The LMS also comprises other services. In discussion boards or chat rooms learners can interact with instructors or other learners and ask questions. A progress monitor composes the assessment results from all lessons into a general overview; this can also be used to create certificates. An accounting service manages all processes which are related to financial aspects. It shows, for example, all booked courses or the bill that has to be paid by an individual. Finally, it should be mentioned that the functionality of the LMS can be extended by other Web services, which can either be provided internally (i.e.. in a local network) or externally from other suppliers over the Web. The flexibility of Web services also allows to distribute services of the LMS on different machines in a local network or over the Web.

Integration of Grid Middleware and LMS-- After having discussed the functionalities of grid middleware and LMS in the previous subsection, resp., we will explain their integration next. The e-learning PC is used by learners to access the LMS. At the same time, such a PC can also be used as a resource for the grid. This has been modeled by the "is a" relationship in Figure 2, which also illustrates that at the same time not every resource of the grid needs to have access to the LMS.

The LMS login service makes it possible for the e-learning PC to become a resource in the grid. When the learner authenticates himself on the Web page that is connected to the login service of the LMS. the fabric layer applet of the grid can be transferred as mobile code and be started locally on the e-learning PC. This enables communication with the grid. The LMS login service transparently calls the Grid_login operation of the grid login service
with the data and the PC of the user as parameters. This completes the authentication process of the e-learning PC in the grid. If a user of the LMS is not registered at the grid login service, the LMS login service could be given the authority to create new users in the grid login service, based on a trust relationship between the two services. These steps keep the login procedure of the grid completely in the background, so that the learner is not aware of it.

B. Grid Architecture—II

Grid infrastructures support the sharing and coordinated use of resources in dynamic global heterogeneous distributed environments. This includes resources that can manage computers, data, telecommunication, network facilities, and software applications provided by different organizations . A Grid is a collection of distributed computing resources available over a local- or wide-area network that appears to an end user or application as one large virtual computing system.

Globus Toolkit --The Grid represents common properties that have very large, distributed dynamics that cross the boundaries of human organizations [2]. Due to these characteristics, its architecture must be very complex and its relationships like a maze. The vital task in realizing a Grid is establishing a common open standard. Its core is based on an open set of standards and protocols - e.g.. the Open Grid Services Architecture (OGSA) – that enable communication across heterogeneous, geographically dispersed environments. With Grid Computing, organizations can optimize computing and data resources, pool them for large-capacity workloads, share them across networks, and enable collaboration . Some protocols and applications have been proposed and implemented in actual works, such as the Globus Toolkit .

The Globus toolkit components most relevant to OGSA are the Grid Resource Allocation and Management (GRAM) protocol, the Meta Directory Service (MDS-2). And the Grid Security Infrastructure (GSI). These components provide the following essential elements of a service-oriented architecture:

1. GRAM protocol: this component provides for secure, reliable, service creation and management of arbitrary computations.
2. MDS-2: this component provides a uniform framework for information discovery through soft-state registration, data modeling, and a local registry.
3. GSI protocol: this component supports single sign-on. authentication, communication protection, and certification mapping.

Data Grid-- In This study, we adopted the Globus Toolkit as the Data Grid infrastructure. The Globus Toolkit provides solutions for such requirements as security, resource management, data management, and information services. Many research projects, such as GriPhyN , PPDG , and EU DataGrid, are based on the Globus Toolkit. The Globus Data Grid comes in two layers. On the Low Level are Data Grid Core Services, and on the upper layer are High Level Components [4]. Fig. 4 shows the Data Grid Architecture.

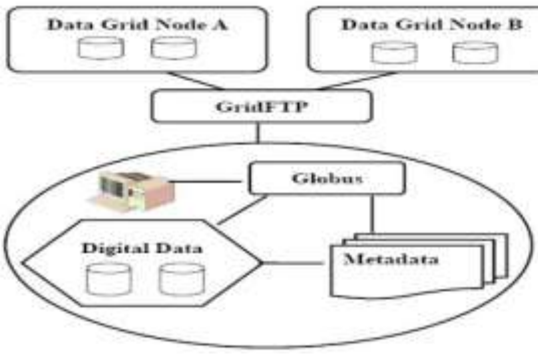


Fig. 4 Data grid architecture.

The storage system is a basic data grid component. It defines and covers all storage technologies capable of adding, deleting, reading, writing and operating file instances , such as HPSS. and DPSS (Distributed Parallel Storage System). Data access services are set up to access, manage, and transfer data in the storage system [5].

Resource Management is responsible for the storage system, networks and other data grid resources needed to assure end-to-end efficiency, technical assessment of efficiency testing, and crucial resources. The Grid Security Infrastructure (GSI) provides environment authorization and certification mechanisms for large numbers of users.

Underlying the Data Grid framework is a key element known as replica management , which is important for successful processing of large amounts of data by the Data Grid. It mainly decides when and where to set up replicas and provides information about replica locations, including these key functions:

1. Registration: this function adds new files to the Replica Catalog and shows where the new files may be accessed by users.
2. Creation and deletion: these functions create and delete registered replicas.
3. Publishing: this function reproduces unpublished files in destination storage systems for publication.
4. Copying: this function reproduces files among storage systems.
5. Query: this function checks physical locations in storage systems where replicas are stored.

Data Grid applications produce huge amounts of data, and managing replicated data in a Data Grid is a major problem. The Replica Location Service (RLS) provides a mechanism for registering and discovering replica data on the grid. The RLS architecture contains the following five elements:

1. Local Replica Catalogs (LRCs) that contain mappings from logical file names (LFNs) to Physical file names (PFNs):
2. Replica Location Indexes (RLIs) that aggregate state information about one or more LRCs and support the discovery of replicas at multiple sites using soft state protocols, such as LDAP. to maintain RLI states:
3. optional compression of soft state updates to reduce communication, CPU. and storage overheads: and
4. management of RLS membership for locating LRCs and RLIs.

Grid Portal--Grid portals allow communication between grids and the outside world, and are always huge, complex frameworks. The main portals are the Application Portal (AP) and the User Portal (UP). AP portals [6] allow specific grid operations for specific applications, such as the Astrophysics Simulation Collaborator (ASC) portal and the Diesel Combustion Collaborator}* (DSC) portal . User portals provide special services to specific members of the public and researchers, such as the HotPage Portal user portal,the Gateway project [7], and UXICORE.

From a user standpoint, the primary requirements of a Grid portal system include [8] the following:

1. Security: Users visit portals using web browsers and are authenticated by means of user IDs and passwords. While better authentication technologies exist, this one is demanded by users. More secure systems, such as small cards, are possible, but unlikely to be deployed anytime soon.
2. Remote access: Tools for accessing file metadata directories and remote file archives are a central portal requirement. Simple Grid FTP tools are essential, but many files are likely to be managed by virtual data systems in which data is cataloged, curetted, and staged by back-end grid services.
3. Remote execution: The ability to submit jobs to the Grid for execution and monitoring is a standard portal requirement. Users allocating specific resources want to be able to see the job queues on those resources and consult scheduling assistants. They also need to be able to read logs to keep track of job execution and know when operations fail.
4. User information services: Access to directories and index tools is an essential portal function. All users should have private, persistent stores of references to important information they have stored on the Grid.
5. Application interfaces: The key to scientific portals is being able to hide Grid details behind useful application interfaces. Users need to be able to launch, configure, and control remote applications in the same way they use desktop applications.
6. Access to collaboration: All Grid organizations must permit resource sharing. This includes real-time and asynchronous collaboration.

The Front End - Grid Portal--Using Access Grid technologies, schools can integrate training courses and materials within the Grid environment, which provides greater teaching flexibility. It is first necessary to set up a Portal Web Site in the Grid system. This provides services to Grid members and acts as a teaching platform for other institutions via the Internet.

ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 1, Issue 8, October 2012

Organizations or individuals who have not joined the Grid will be restricted in their access to some resources.
Here the Grid utilizes the NMFs OGCE Portal [9] as the network Grid Portal Site. The OGCE Portal includes the following functions:
1. posting on the Discussions board and communicating with friends in Chat rooms, and providing users with the latest grid-related technological updates in the News section:
2. monitoring Grid resources, such as node operating conditions:
3. submitting tasks for Grid operations:
4. transmitting data within the Grid via GridFTP: and
5. managing the Grid CA via the Proxy Manager.
The OGCE Portal framework provides a general portal architecture that supports virtual organizations composed of scientists and project developers, and also provides the API for the development of reusable, modular components that can be used to access the services being developed within the Grid organization.
The Back End - Data Grid Contents—The system is designed in accordance with the architecture shown in Fig. 5. CAI software additions, along with growing numbers of users and increasing amounts of teaching materials will continuously increase the server load. Therefore, the storage device may need to be updated or replaced after a certain period of time. To overcome this problem, we will work on connecting idle storage devices within the institution using Data Grid techniques in place of expensive storage devices, such as a Disk Array or NAS. in serving as an e-learning platform storage device.

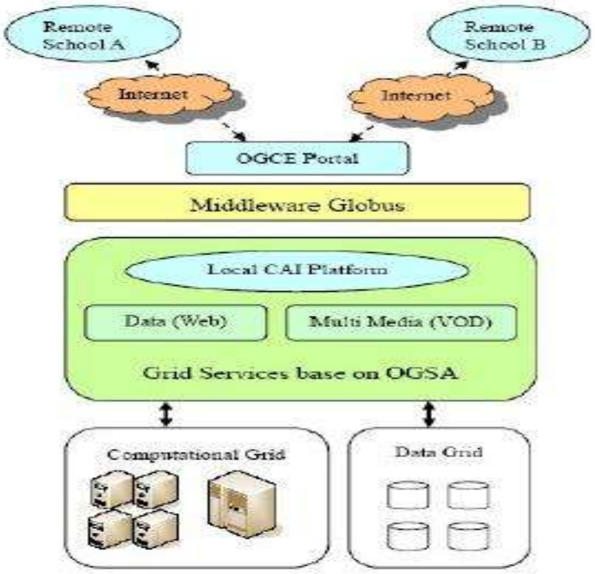


Fig. 5 E-learning system framework.

C. *Grid Architecture—III*

Resource Management deals with number of challenging issues with important ones being resource discovery, resource scheduling, resource allocation, resource protection, resource monitoring and control. Today grids are mature enough to handle these issues effectively. The most notable resource management middleware solution is the Grid Resource Allocation Manager (GRAM). This resource management middleware provides robust job management services for users some of which includes job allocation, status management. data distribution and jobs start/stop configurations [10]. It provides a set of standard interfaces and components to collectively manage a job task and to provide resource information. This information can be used for various purposes. Following figure describes the basic architecture of GRAM.
Master Host Environment and User Host Environment enable separation of functionalities and give an improved abstraction on functions executed by each environment. The master host is direct point of contact for client. It provides information on aggregated resource state or status, and manages its user hosts start and launch services. The user host environment (UHE) executes all the jobs and provides specific abstraction capabilities and securities to execute job [10]. Master Managed Job Factory Service is responsible for receiving the client request on aggregated resource queries and subscriptions. It manages the aggregated service data by aggregation providers for resource status and through notifications received from resources [10]. Virtual Host Redirect Handler is the core component for redirecting all the calls to UHE. These Calls include creation of job and invoking of job operation [10].
When client request to execute a job. the virtual hosting engine directs the calls to starter. This Java class is responsible for Security mapping, user validation and for ensuring that job is executing so that virtual host can redirect the calls to executing job service. When user host is not up and miming, it uses the help of Lunch UHE Java class to start that host under user's credentials [10].
Managed job Factory Service exposes CreateService method, which accepts an RSL-specified job. It then creates managed job instance for user. Moreover it acts as a local schedular. monitoring its status and sending notifications.
Managed Job Service will start two file streams factory services; one for job's stdout, and the other for the job's stderr.
File stream Factory Service/File stream service are helpful services to manage the data needed for the job execution. The factory service creates two file stream services: stdout and stderr. Each of these services has two service data results: the URL for the stream destination, and a flag to indicate the activity [10]. Grid Resource Identity Mapper (GRIM) service is executing in the UHE to create a user host certificate. The user host certificate is utilized for mutual authentication between MJS service and the client [10].
Resource Information Provider Service (RIPS) is a specialized notification service providing data about a scheduling system, file system, host [10].

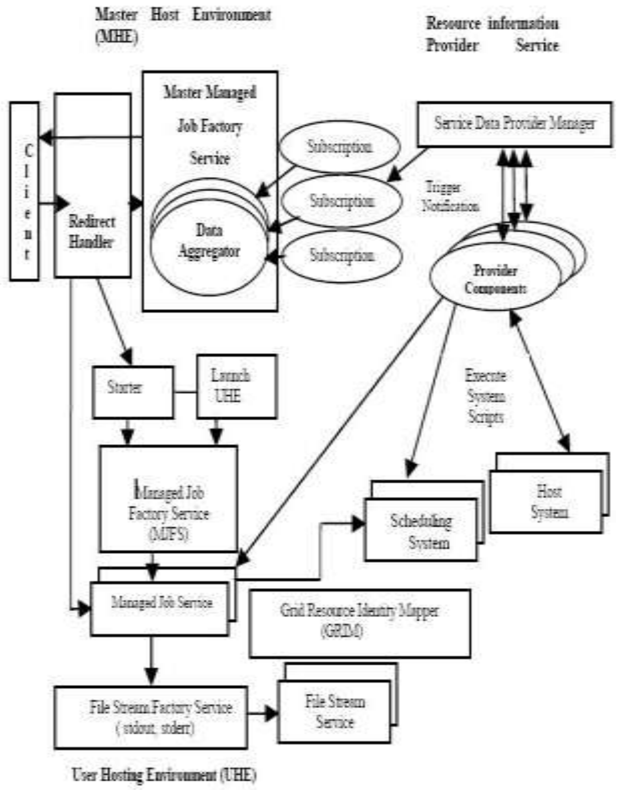


Fig 6 GRAM Architecture

IV. CONCLUSIONS

This paper is intended mainly as an attempt to draw up a research agenda for an exploitation of grid computing in e-learning. and many details remain to be filled in. Yet it appears feasible to pursue the area, as there is considerable hope for being able to extend the achievements of electronic learning beyond the limits of individual computers. Currently, we are going though the implementation phase Future issues to be studied include, for example, transactional guarantees for service executions over a grid such as atomicity, or recovery protocols that help restore an operational state after a grid failure . in future we are planning to implement it with the trust and privacy features also. For this, the work in the areas for privacy and trust is also going on.

REFERENCES

[1] Hamscher V., U. Schwiegelshohn, A. Streit. R. Yahyapoiir. "'Evaluation of Job-Scheduling Strategies for Grid Computing," in R. Buyya andM. Baker (Eds.:) GRID 2000, LNCS 1971. Springer-Verlag, Berlin. 2000. pp. 191-202.

[2] A. Chervenak. I. Foster. C. Kesselman. C. Salisbury, and S. Tuecke. "The data grid: towards an architecture for the distributed management and analysis of large scientific datasets." Journal of Network and Computer Applications, Vol. 23. 2001. pp. 187-200.

[3] G. Fox. "Grid computing environments." Computing in Science and Engineering, Vol. 5. 2003. pp. 68-72.

[4] S. Vazhkudai. S. Tuecke. and I. Foster. "'Replica selection in the globus data grid." In Proceedings of 1st IEEE/ACM International Symposium on Cluster Computing and the Grid, 2001. pp. 106-113.

[5] S. Malaika. A. Eisenberg. and J. Melton. "Standards for databases on the grid." ACM SIGMOD Record. Vol. 32, 2003. pp. 92-100.

[6] M. Thomas. M. Dalian. K. Mueller, S. Mock. C. Mills, and R. Regno. "Application portals: practice and experience. Journal of Concurrency and Computation: Practice and Experience, Vol. 14. 2002. pp. 1427-1443.

[7] M. E. Pierce. C. H. Youn. and G. Fox. "The gateway computational Web portal." Journal of Concurrency and Computation: Practice and Experience. Vol. 14. 2002. pp. 1411-1426.

[8] J. Novotny. "The grid poital development kit." Journal of Concurrency and Computation: Practice and Experience. Vol. 14. 2002. pp. 1129-1144.

[9] R. Stevens. M. E. Papka. and T. Disz. "Prototyping the workspaces of the future IEEE Internet Computing, Vol. 7, 2003, pp. 51-58.

[10] Joshy Joseph and Craig Fellenstein, "Grid Computing". Pearson Education. 2004.

Author Biographies

Sanjeev Narayan Bal
MBA, MCA MTech(CSE)
Assistant Professor (CSE)
Trident Academy of Creative Technology
Bhubaneswar
Odisha
Life Member – ISTE