The OMII Software – Demonstrations and Comparisons between two different deployments for Client-Server Distributed Systems

Victor Chang
OMII-UK, University of Southampton, United Kingdom
v.chang@omii.ac.uk

Abstract

This paper describes the key elements of the OMII software and the scenarios which OMII software can be deployed to achieve distributed computing in the UK e-Science Community, where two different deployments for Client-Server distributed systems are demonstrated. Scenarios and experiments for each deployment have been described, with its advantages and disadvantages compared and analyzed. We conclude that our first deployment is more relevant system administrators or developers, and the second deployment is more suitable for users' perspective which they can send and check job status for hundred job submissions.

1. Introduction

A distributed system consists of a collection of autonomous computers connected by a network and equipped with distributed system software, which allows resource sharing of system hardware, software and data amongst users, who often perceive a single, integrated computing facility even though it is implemented by grid or multiple clusters located in different locations [6]. This leads into the concept of distributed computing, which is defined as "a method of computer processing in which different parts of a program run simultaneously on two or more computers that are communicating with each other over a network" [14]; or grid computing, which is an emerging model of distributed computing that employs a dynamic pool of dispersed commodity computer resources to tackle large, time-consuming tasks. To enable grid computing, it requires a vast number of CPUs and resources joint together, however, current research focus is on applications, which include software development, integration, testing and user scenarios. That is one of main reasons motivating and launching UK e-Science Core Program, which defines that "e-Science is about global collaboration in key areas of science, and the next generation of infrastructure that will enable it. e-Science will change the dynamic of the way science is undertaken." [12].

1.1. Current status for distributed computing

In the high performance scientific computing or bioinformatics, where simulations and visualizations are increasingly complex and more difficult to analyze, e-Science is always the response for those scientific questions, as they can meet tougher demands and deal with greater challenges [2, 3]. There is a growing aspect in the scientific collaboration, where the combined expertise of the multidisciplinary research teams is required to work asynchronously or synchronously in the distributed environments. The term "virtual organizations" (VO) best describes the scenarios where groups of scientists or researchers working collaboratively in a secure networked environments, for which the real and specific problem underlies in the distributed computing is the coordinated resource sharing and problem solving in dynamic and multi-institutional VO, where sharing implies direct access to computers, software, data [1, 7]. In order to achieve full resource sharing and maintain the openness of the distributed system, the UK e-Science Program has introduced the first phase of £250 Millions funding [5] for regional and national e-Science projects, and the second phase for selected institutions, which the Open Middleware Infrastructure Institute UK (OMII-UK) and the National Grid Service play influential and critical roles in the strategic and evolutionary part of e-Science transformations [5, 12].

2. The OMII-UK

Founded in January 2006, OMII-UK [10] is a collaboration between the School of Electronics and Computer Science at the University of Southampton, the OGSA-DAI project at the National e-Science Centre and EPCC, and the ^{my}Grid project at the School of Computer Science at the University of Manchester. This partnership aims to be a leading provider of reliable interoperable and open-source Grid or e-Science components services and tools to support advanced e-Science enabled solutions in academia and industry. OMII-UK continues with its mission to provide the UK e-Science community and offer international collaborators software and support for enabling a sustained future.

2.1. Introduction for the OMII Software

The OMII-UK develops, fixes, tests and maintains a catalogue of e-Science applications and projects to assist users in achieving their scientific goals or research objectives, where the main activities for software engineering processes are truly reflected in the half-yearly software releases. The latest release, known as OMII_3.4.0 [10], is the integrated software of up to 13 components for client and up to 13 components for server, each of which has a dedicated service to meet various engineering challenges. The core technology is based on Tomcat 5.0.25 Axis 1.2-RC3, and only compatible with Java 5. The OMII software components include Account (Account Service), BPEL (Workflow for Business), FINS (Notification), FIRMS (Reliable Messaging), GeodiseLab (simulations for Integrated Services), GridSAM (Job Submission & Monitoring Service), Gridsphere (Portals and portlets), Grimoires (Registry Service), Integrated Services (combination for account, job, resource allocation and data service), OGM (Grid Management Framework), OGSADAI (Data Service for Grid), PlotWS (Gnuplot), Taverna (Workflow for Boinformatics), WSRF::Lite (Web Services Resource Framework) and AHE (Application Hosting Environment), which requires WSRF::Lite as a base component. All the software components provide its respective client and server components, and client-server interactions can be easily demonstrated by making the client component A to interact with server component A. The only exceptions are Taverna and GeodiseLab (clients only), which can interact with PlotWS and Integrated Services respectively.

2.2. OMII Software Components

The following descriptions are the summary for each software component [10, 11]:

(a) BPEL (Business Process Execution Language) provides scientific workflow modeling and Grid services orchestration, as it offers integrated environment to model, execute and monitor scientific workflows that are expressed in BPEL, which BPEL Designer for process modeling, and a tailored open source BPEL engine, ActiveBPEL on server, and Eclipse on client for process execution and monitoring.

(b) FINS (Federation and implementation of Notification Specification for Web Services) currently supplies open source implementations of WS-Eventing specifications and later a WS-Notification implementation.

(c) FIRMS (Federation and Implementation of Reliable Messaging for Web Services) represents the open source implementations of the WS-Reliable Messaging and WS-Reliability specifications.

(d) GeodiseLab offers three toolboxes that provide facilities for accessing computing resources for various problem solving environments, data management, file transfer and certificate handling.

(e) GridSAM is an open source job submission and monitoring service. GridSAM installs on the top of WS-Security (authentication) layer provided by the OMII server that has a variety of data input and output requirements. GridSAM implements the Job Submission Description Language (JSDL) and is available on the National Grid Service (NGS) [9] grid for scientists and researchers to carry on e-Science application development and research work.

(f) GridSphere is a standards-based portlet framework for building web portals, which interfaces with a set of portlet web applications. GridSphere provides a cohesive "grid portal" end-User environment for managing users and groups, supporting remote job execution, file staging and providing access to information services.

(g) Grimoires (Grid Registry with Metadata Oriented Interface: Robustness, Efficiency, Security) enables storage of service descriptions, distributed queries, WSDL documents and workflows. This registry also offers facilities for semantic annotation of information, and can be deployed together with other components such as GridSAM, OGM and Taverna. Grimoires will be available on the NGS for UK and European e-Science research and application work.

(h) Integrated Services and Account Service: Account Service can be deployed as a stand-alone application, where it is used to support the creation and management of accounts. Each account represents a trust relationship between a Service Provider and a user who is responsible for all service access on their account. The Integrated Services include four WS Services: account service, resource allocation service, job service and data service. The Integrated Services can be used to demonstrate banking and accounting services for financial transactions and also used for Cauchy Horizon Application, which represents a useful mathematical problem-solving scenario to showcase the Integrated Services.

(i) OGM (Open Grid Manager) is a lightweight open source grid management framework, providing a cohesive solution for monitoring and managing arrays of heterogeneous grid resources deployed within live production grids. OGM consists of an OGM-Agent, which is deployed on the managed grid resources and communicates with a suite of Web Services (OGM-Server) to expose persistent registry capabilities.

(j) OGSA-DAI (Open Grid Services Architecture Data Access and Integration) is a middleware which allows data resources, such as relational or XML databases, to be accessed via web services. An OGSA-DAI web service allows data to be queried, updated, transformed and delivered. OGSA-DAI can be used to provide web services that offer data integration services to clients, and it is available in leading open source e-Science software such as OMII and Globus Toolkit.

(k) PlotWS Web Service offers an interface to a subset of the Gunplot functionality, thus providing a graphical function through a Web Service API and making it easier to use than Gnuplot itself. It is often used with Cauchy Horizon Application or Taverna to demonstrate constantly-updated visualization.

(l) Taverna is an open-source workflow tool, which provides a workflow language (SCUFL - Simple Conceptual Unified Flow Language) and graphical interface to facilitate the easy building, running and editing of workflows over distributed computer resources. Taverna Workbench allows users to construct complex analysis workflows from components located on both remote and local machines, run these workflows on their own data and visualize the results, and is particularly useful for biologists or bioinformatics.

(m) WSRF::Lite (Web Services Resource Framework Lite) and AHE (Application Host Environemnt): WSRF::Lite provides a Container for hosting WSRF compliant Web Services written in Perl and it also provides a number of base classes for building WSRF Web Services that provide the required operations. It is often used with AHE, which is a lightweight hosting environment that allows scientists to run applications on grid resources in a quick, transparent manner. The AHE provides resource selection, application launching, workflow execution, provenance and data recovery, exposing a WSRF compatible interface to job management on remote grid resources using WSRF::Lite as its middleware.

2.3. The OMII Security

The OMII software provides a temporary Certificate Authority (CA), which facilitates authentication for both client and server site. Based on WS-Security, or WSS4J in short, is anther security feature is WSS4J, and its main function is to provide enforced security handler. It is essentially a Java library that can be used to sign and verify SOAP Messages with WS-Security information, which allows for the processing and generation of the following SOAP bindings [10]:

- XML Security (OASIS SOAP Message Security 1.0 Standard)
    - XML Signature
    - XML Encryption
- Tokens
    - Username Tokens
    - Timestamps
    - SAML Token

Another layer of security is on Transport Level Security, where HTTPS becomes the default protocol for installations and all client-server activities since OMII 3.4.0 Release. HTTPS provides efficient message encryption with handshake authentication, which helps to resist denial of service attacks.

3. First Type of Client-Server Distributed System

Openness is the property of distributed systems such that each subsystem is continually open to interaction with other systems [10, 14]. A distributed system is open if it satisfies three conditions: (a) it is modular; (b) it is made up of components that are interchangeable and pluggable; and (c) those components can be obtained from many independent sources [8]. OMII software is made up of many components, each of which was originally funded to and contributed from commission software partners, with the integration and testing done mainly by the

Southampton team. In this respect, OMII software satisfies these three conditions as (1) each component has its unique functionality and have plug-and-play state; (2) each subsystem can interact with its own components or other subsystems and (c) it is possible to integrate or join up with other components to become a new or emerging service, where each component originally comes from independent sources.

3.1. First Deployment: Modular System's Multiple Client-Server Activities

Numerous clients and servers can be installed on a single machine, where client-server interactions take place between Client A and Server A, and between Client B and Server B, and between Client C and Server C, and between Client A and Server B, and between Client B and Server C, and between Client C and Server A. A server can contain a minimum of 1 component and serve a single function; on the other hand, a server can contain all the available components and serve all possible functions. Similarly, this applies to client, where a client can have one component and serve one function, or it can have all the components and allow all possible functions. With the plug-and-play state-of-art and feasibility for being either an integrated system, or single component, multiple servers can be presented as a single entity (server) within a machine and similarly, multiple clients can be presented as a single entity (client) within a machine. The left-hand side of Figure 1 best represents this multiple client-server activities scenario, which can be further simplified as the right-hand side of the Figure.

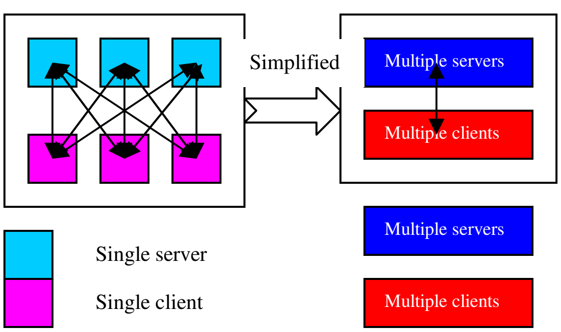

Figure 1: Multiple client-server activities scenario and its simplified representation

Client-server activities within the distributed system not only take place within a machine, but also between multiple machines in a cluster or in a pool, forming a higher-level of client-server interactions. This is often done in Southampton where tens of test racks have multiple client server activities, which can be grouped into three main categories: (1) individual component's client-server activities, which often take less than three minutes; (2) integrated components' client-server interactions, which range from three minute tasks to twelve hours tests; (3) stress test, which range from a minimum of four hours to seven days. Figure Two is the best representation for client-server interactions within a cluster or a pool of machines.

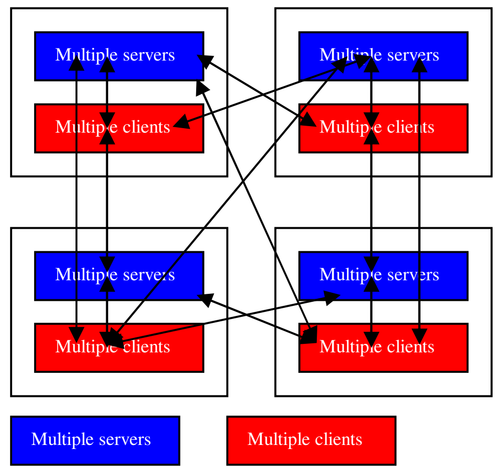

Figure 2: Multiple client-server activities within a cluster

For taking (2) integrated component's client-server interactions as an example, BPEL and Taverna can be used for illustrations, where Taverna require PlotWS to capture bioinformatics workflow, and BPEL require Eclipse and ActiveBPEL engine to publish workflow, and eventually allow BPEL to invoke client-server WSDL, which can be carried out for as short as three minutes or as long as running till next morning from the previous evening. Apart from the fact that system team running stress test harness with their Perl scripts or overnight builds, stress tests can be carried out for software components, for which OGM and Grimoires are good examples. Two types of tests are designed: The first test is to write a script to run OGM (which collects resources from Grimoires) every minute, and specify all physical memory of 4GB can be allocated to memory required by Tomcat (default takes 128MB), and run OGM for up to seven days, record whether OGM client-server activities will fail, and it does not from test results. The second test is to test OMII software on platforms that it has never been tested, which include Debian 4.0 and x86_64 platforms such as Redhat Enterprise 4 and SuSE 10.1. Any components can be used, and in this case, Cauchy, is set to run continuously, with the objective to test its capabilities under stress test, and eventually pass tests.

The size of the cluster is not a main factor in this demonstration, as the focus is to demonstrate client-server interactions, which can asynchronously and synchronously take place and are independent of geographical locations, time zones and continents. To demonstrate this, two sets of tests are designed. The first test is to begin client-server interactions between different clusters in the UK, one is based in researcher's home, another is based in the researcher's office, and another is based on University College London (UCL). The client-server events take place by both manual and automated operations (Cauchy in continuous mode), and allow clusters between three different sites to operate in smooth and well-balanced manners. The second test is to allow global client-server events. While taking factors such as high expenses and lack of resources into considerations, the most relevant way to carry out is to get installed clients in other parts of the world, to interact with servers located in the three sites in the UK. Several machines in Singapore, Australia and the UK simultaneously make client requests to the server clusters at researcher's home, researcher's office, and UCL. Concurrent client-server events between three clusters in the UK also take place, and this test can demonstrate that client-server activities are independent of geographical locations and time zone, and Figure 3 is the best representation for this, which purple rectangles symbolise the cluster, yellow rectangles represent machines located in other parts of world, where blue rectangles represent multiple servers and red rectangles symbolize multiple clients within the cluster.

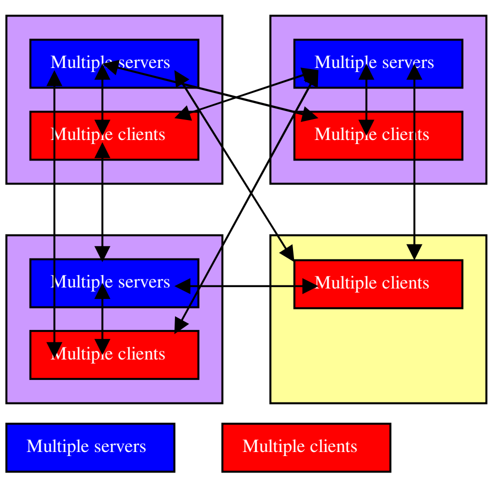

Figure 3: Multiple client-server activities between clusters

4. Second Type of Deployment

The second type of deployment is the most relevant for users' perspective, where a single client can send all or different jobs or requests to a significant amount of servers, so that either a big task can be executed across all the servers, or all the servers can run hundred of the same requests simultaneously. This type of distributed system often happens in the Grid, where a single Grid handles hundred of requests at all times, regardless where the requests come from provided if they fit into the security requirements.

4.1. The UK National Grid Service

The National Grid Service (NGS) [9] in the UK was founded in October 2003 and full production services were offered in September 2004. The UK NGS facilitates a core e-Infrastructure that underpins UK research, providing standardized access to compute resources, data resources and large scale facilities, enabling collaborative computing across the UK. There are currently nine sites, four of which are the NGS main sites located at the Universities of Oxford, Manchester, Leeds and Rutherford Appleton Laboratory (RAL), and the other five partner sites are based at the Universities of Westminster (London), Cardiff, Lancaster, Belfast and Bristol. Each site has up to 64 compute nodes with dual Intel Xeon 3.06 GHz per server node, and has various UNIX and Linux distributions. Computing resources at all the sites can be joined up, thus forming a significant resource equivalent to a Grid, which provides a service platform for scientists and researchers to carry out experiments, testing, job submissions and collaborative work amongst the UK Academic Community.

To apply for access, it requires UK e-Science X.509 certificate, where strict application process is required. There are several types of X.509, the common type is personal certificate, which should match the identity of the applicant. Another common type is server certificate, which is DNS specific, and is open and valid for servers hosted at the applying University. To demonstrate the second distributed system mode, only personal certificate is required. The access to clusters at different sites is dependent on MyProxy, where the personal X.509 certificate should be uploaded to a trust MyProxy server. Upon successful upload and authentication, the access to all sites of the NGS Grid is then achieved.

4.2. Experiments: Send hundred of jobs from a single client to the NGS Grid and 2 clusters

GridSAM is widely used amongst the UK Academic Community, and this is currently available on the NGS Grid, where Queen's University of Belfast takes care of it and joins other eight sites' GridSAM servers. Providing Job Submission Description Language (JSDL), GridSAM is the best component for demonstration, since it allows multiple job submissions and monitoring, which can take place synchronously and asynchronously based on different needs.

The experiment is to send a hundred jobs from a single client to the NGS Grid, the clusters in Southampton and the clusters in the UCL, and to test (1) the extent which the servers can cope with multiple and concurrent job submissions; and (2) the ability for a client to act as a control center. Since NGS Grid is shared and utilized by numerous UK researchers and thousands of jobs at the most of times, it is expected that the response time from the server can be slightly slower. A single client then sends jobs to all the computing resources, and upon completion, get the job ID and check its status. The server then replies back to the client that the job is active, undergoing a process and finally completed. This demonstrates that distributed system can be coordinated by a single client, which can send jobs to different computing resources and eventually completing the tasks. Figure 4 below is the presentation for this experiment.

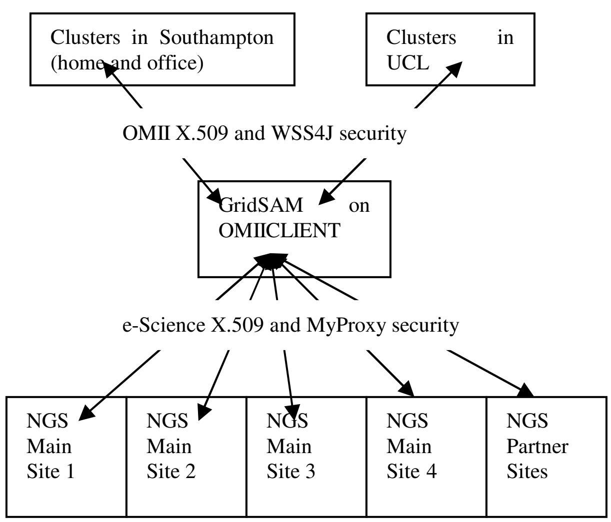

Figure 4: Multiple and concurrent job submissions to the NGS Grid and 2 other clusters.

Another experiment is designed to test the capabilities that the NGS Grid and the other two clusters can handle when an addition of two GridSAM clients send jobs simultaneously with the existing one. Each GridSAM client is installed on different operating systems, with one on Windows XP, one on Macintosh 10.4, and one on Red Hat Enterprise 4. The test is the same as the previous experiment, except that three GridSAM clients send at the same time. It is noted that the entire process works fine if manual operations are used throughout the test execution, since human judgment helps to identify where to send jobs when a resource is less utilized. However, some jobs fail to be completed if relying on automated system and the diagnosis is that a scheduling system is required to improve job monitoring. OMII-Europe has a software component known as Crown, which can improve the scheduling system but the detail for client- server demonstration is not described in this paper.

5. Discussions

We have demonstrated OMII software as two different deployments for client-server distributed systems, and each deployment should address for different needs. Before concluding the suitability for its respective potential groups, we list down the advantages and disadvantages for the two deployments as summarized in the tabular form below:

5.1 Modular client-server systems

Advantages	Disadvantages
- Can use a minimum one machine for setting up client-server systems. - Suitable for those who make frequent changes to their software or system such as developers or system administrators, as client-server results can be checked live. - There is no need to rely on third-party to fix problems as servers and clusters are usually controlled and maintained by researchers.	- Often require setting up machines and clusters, which can be sometimes time-consuming and expensive to maintain - Occasionally multiple client-server activities between clusters can become too complex to handle. - Require good knowledge to configure and modify servers or scripts.

5.2 Central Client to the Grid

Advantages	Disadvantages
- Convenient, easy to deploy jobs, as all can be done in a few minutes. - Suitable for users' perspective, as there is no need to set up machines and servers, which can be complicated and require specific knowledge to deploy. - Can save up funding for computing and human resources on users' side. - Provide a more secure, robust and collaborative platforms for national and international research and development.	- Setting up National Grid needs millions of pounds and large resources to support. - Heavily rely on the third party services in the National Grid. If one site's cluster is down, jobs might not be completed. - Have no system and policy control on the National Grid. - If there are thousands of requests to the Grid, the response time is very slow and may sometimes result in system crash.

5.3 Analysis

Methods of deployments depend on the organizational needs, research goals or individual preference. While setting up modular client-server systems in multiple machines or clusters may sound complicated for some researchers, it provides good learning opportunities because it requires researchers to learn new knowledge and keep skills up-to-date. Flexibility is another added value as it allows researchers to install or uninstall any components, make any client-server activities between any machines or clusters, as well as learn characteristic, strength and weakness for each software component. Researchers themselves become the centralized model as co-ordinations for any client-server activities depend on their schedule and can best fit into their plans.

On the other hand, sending hundreds of tasks from a single client to the Grid and the clusters is seen as convenient and quick to finish tasks, by disseminating all the tasks, which are shared and then completed by computing resources located in different places. Users can just 'walk away' from connecting to the Grid at any time without having the responsibilities to look after servers and clusters. There is no need to hire staff to look after machines, and require no detailed training on each software component's configurations and troubleshooting. Opposing to the first deployment, OMIICLIENT is the centralized model as it can co-ordinate all client-server events across the Grid and act as a control center for all reply-response activities. As stated in the table, drawbacks include high costs to start and sustain the National Grid, and the amount and complexity to provide such services can be sometimes too complex or too optimistic before starting any experiments. Another issue to upset users is that users have no control over the system and policy issues. One example is that the changes of e-Science certificate format result in the review and rewrite of software codes, testing and documentation. Another example is that if the Grid is over-demanded with thousands of jobs to be finished sent by all researchers in the UK, delay or even system crash on the client side may be expected to happen.

6. Conclusion and Further Work

This paper has introduced the OMII software, each of its components and how OMII software can be deployed into two types of client-server systems. The first deployment, modular client-server system, is suitable for developers or system administrators who can make frequent changes to their system or software, so that they can check client-server results on the live system at any time. There is no need to rely on any other Grid or e-Science vendors or service providers to fix problems as servers and clusters are usually controlled and maintained by developers or system administrators. The second deployment, central client to the Grid, is relevant for users' perspective, as they do not require setting up servers and systems and can just complete all the tasks as short as a few minutes. Additional e-Science X.509 and MyProxy security provide a more robust and secure platforms for collaborative or virtual work.

Currently an OMII component, AHE, is about to be fully in service in the UK National Grid Service (NGS), and ready to serve up all e-Science researchers across the UK. Grimoires, another popular component in the OMII software, is scheduled to be on the NGS for full services before 2008, and is expected to serve UK and EU e-Science researchers as they have been widely used in the UK Open Source and Academic Communities. BPEL recently wins a JAX Innovation Award for the best European contributions to Java, Eclipse and Service Oriented Architecture (SOA), and its Workflow application can be demonstrated on how SOA can be deployed to help organizations in improving its internal operations and generating more revenues. As a core component to both Gloubus and

OMII software, OGSADAI is another component that is under NGS experimentations and this software has been active in the global e-Science community. Finally, we hope to update the status and describe how AHE, BPEL, Grimoires and OGSADAI can be demonstrated for the client-server distributed systems in the coming year.

7. References

[1]: Chang, V., "Implementations of Virtual Organization in the Wireless Home Networking Environments." In Proceedings of IEEE 2006: International Conference on Mobile Communications and Learning, April 23-28, 2006.

[2] Coles, J.S., Frey, J.G et al "ECSES – examing crystal structures using 'e-science': a demonstrator employing web and grid services to enhance user participation in crystallographic experiments", Journal of Applied Crystallography, ISSN 0021-8898, 5 August 2005.

[3] Coles, J.S., Frey, J.G, Mills, H., et al, "An e-Science environment for service crystallography from submission to dissemination", Journal of Chemical Information and Modeling, 46(3) page 1006-1016.

[4] De Roure, D. et al, "The Collborative Semantic Grid", Applied Artificial Intelligence 19(9-10) pp. 881-904

[5] De Roure, D., Henderson, P., Hey, T., et al, "Web Service Grid: An Evolutionary Approach", UK e-Science Core Program Strategy Paper.

[6] Emmerich, W., "Distributed Systems Principle", 1997 UCL Teaching Materials, http://www.cs.ucl.ac.uk/staff/ucacwxe/lectures/ds98-99/dsee3.pdf

[7] Foster, I., Kesselman, C., and Tuecke, S. "The Anatomy of Grid: Enabling Scalable Virtual Organizations", International Journal of Supercomputer Association, Vol. 15, 2001.

[8] Henderson., P., "Architecture and Open Systems" a Short Course, http://users.ecs.soton.ac.uk/~ph/OpenSystems/

[9] NGS Website http://www.grid-support.ac.uk

[10] The OMII-UK Website http://www.omii.ac.uk

[11] Papay, J., Newhouse S, Chang V. et al, "The OMII Software Distribution". In Proceedings of The All Hands Meeting 2006, Nottigham.

[12] Trefethen, A., "ShibGrid – Integrating NGS into the academic framework" Proposed Project Report, University of Oxford.

[13] Taylor, J., Strategic Talk about UK e-Science Program, http://www.nesc.ac.uk/nesc/define.html

[14] Wikipedia, Distributed Computing http://en.wikipedia.org/wiki/Distributed_computing