Portal of Damage on Grid

J. Floréz-López[2], V. Hamar[1], A. Hernández[1], H. Hoeger[3],R. León[1], L.A. Núñez[4], N. Ruíz[1], M. Uzcátegui[1]

[1] Centro Nacional de Cálculo Científico Universidad de Los Andes (CeCalCULA), Corporación Parque Tecnológico de Mérida, Mérida 5101, Venezuela
vanessa, horence, rleon, nicolas, maylett@ula.ve

[2] Departamento de Ingeniería Estructural, Facultad de Ingeniería, Universidad de Los Andes (ULA), Mérida 5101, Venezuela
iflorez@ula.ve

[3] Centro de Simulación y Modelado (CESIMO), Facultad de Ingeniería, Universidad de Los Andes, Mérida 5101, Venezuela, and Centro Nacional de Cálculo Científico Universidad de Los Andes (CeCalCULA), Corporación Parque Tecnológico de Mérida, Mérida 5101, Venezuela
hhoeger@ula.ve

[4] Centro de Física Fundamental, Departamento de Física, Facultad de Ciencias, Universidad de Los Andes, Mérida 5101, Venezuela, and Centro Nacional de Cálculo Científico Universidad de Los Andes (CeCalCULA), Corporación Parque Tecnológico de Mérida, Mérida 5101, Venezuela
nunez@ula.ve

Abstract. The kind of problems currently addressed by scientists are constantly increasing in complexity, always requiring more computing power and storage, both for input data as well as results, in addition to specialized software specific for the different branches of science. There is also frequently a need to distribute geographically, both computation and data in a safe and efficient manner. Traditionally, scientific computation had been performed in centers having supercomputers or clusters, but in many cases the computational capacity of such centers are insufficient for the requirements placed by such computations. The Portal of Damage is a finite element scientific application based on the new called Theory of Concentrated Damage, that combines fracture mechanics, continuous damage theory and the concept of plastic hinge. It can be accessed over the Web using a regular web browser (MS Internet Explorer, Firefox, etc.) to simulate the behavior of reinforced concrete framed structures – typically buildings – under earthquakes or others exceptional overloads conditions with the goal of determining the structural response when collapse or structural failure occurs. The theory of concentrated damage is based on the assumption that all inelastic phenomena can be lumped at special locations called plastic or inelastic hinges. Therefore a frame member is considered as the assemblage of an elastic beam-column and two inelastic hinges at the ends of member. This theory is obtained by the introduction of a variable, called damage, which measures the crack density in the element. The damage ranges from zero to one, where zero represents elements with no damage (no cracking) while a value of one represents a totally cracked state. It must be point out that through the portal only two dimensional (2D) reinforced concrete framed structures can be analyzed. The first version developed allows for

the calculations to be performed on a computational cluster located at the Universidad de Los Andes National Center for Scientific Computations (CeCalCULA). Because of the very active fault of Boconó, the Andes region is a highly seismic zone, thus there is a lot of demand for this application in the region. This means that the current cluster version of the Portal of Damage can not adequately fulfill the user's requirements, causing long job queues, especially during training courses of the tool. Due to this limitation and the expected heavy use of the portal, rises the need to migrate the portal so that it can use the computational resources available on the Grid, given that CeCalCULA is already a partner in some grid projects, like Grid Venezuela and the EELA-2 project, and for the EELA project. This allows for the utilization of computational resources in Venezuelan universities, as well as in Latin America and Europe. This document describes the experience gained in CeCalCULA through on going migration of the Portal of Damage application. It describes it's architecture, gives a brief description of the software, explain in general terms the run of some preliminary tests sent to the grid, shows the first results obtained and compares run times. Finally, it describes future plans for forthcoming tests and the development of the Portal of Damage to three dimensions (3D) on Grid.
Key words: Computational Grids, Damage Theory, Finite Elements, Portals, Portico.

1 Introduction

As soon as computers were developed, they became a fundamental tool, enabling the computations that would solve problems being faced by the scientific and engineering communities. The computing requirements by these communities are constantly growing. As a response to this needs, computational clusters were developed. Nowadays a new set of requirements are emerging, among them is the use of heterogeneous hardware and sharing resources among geographically distributed organizations. From these requirements the Computational Grids or simply Grids have risen.

The term Grid first appeared as an analogy to the electrical grid, to refer to a technology that enables scientists to access distributed computing power transparently, i.e., grid users can connect and use it at any time without having to worry about the computing power source. Among the benefits of computing grids is the ability to share computing an storage resources among geographically distributed locations, allows to use the heterogeneous hardware already available, as well as being dynamic in nature as new sites and institutions join the Grid. The ultimate goal is for one Grid just like there is a one Web [7]. The user will not need to worry about the infrastructure complexity, but rather will be able to use a digital authentication (based on certificates) to send jobs via an intuitive portal. The Grid middleware, a software layer interfacing the distributed infrastructure with the applications, will take care of finding the resources, authorizing its use, executing the job optimize data access, and finally return the results to the user [13].

Recently, this type of omnipresent computing infrastructure has lead to a new way of research to produce and disseminate knowledge called *e-science*, e-engineering or more ambitious *e-research*. The term "e-Science" introduced by John Taylor in 2000 and recently recoined as *e-research* is a new trend describing the strategies and tools for global collaborations in key areas of science. It defines a set of computational hardware and middleware and data services that enable service oriented science [8,9,6].These infrastructures and facilities have made it possible to develop computational "collaboratories" [10], defined as places where scientists work together to solve complex interdisciplinary problems despite geographic and organizational boundaries. Such collaboratories provide uniform access to computational resources, services and/or applications. They also expand the resources available to researchers, foster multidisciplinary collaborations and problem solving, increase the efficiency of research and accelerate the dissemination of knowledge.

Modeling civil engineering structures does not escape from this need of simulations based on heavy calculation and data analysis. The Portal of Damage[5] (PoD) allows the user to compute the cracking density at the edges of each of frame element. With the results the structural design can be calibrated and therefore increase or decrease the reinforcement or the section's dimensions, adjusting the structure to a desired resistance level. The relevance of the portal derives from the fact that in Venezuela (an in most of the Latin American countries) there exist risky buildings that need to be reinforced. Thus, the PoD was developed to become a tool to diagnose reinforced concrete structures [12].

The PoD started as a web based client-server application that performed calculations in the same server where the portal was installed. Considering the benefits of increased computing power, the portal was updated to be able to send jobs to a dedicated cluster. Looking forward, the portal is upgraded again to take advantage of the web and grid ubiquity.

This document describes an ongoing project to migrate the PoD from the cluster platform to a Grid environment, along with the gains achieved. Is worth noting that both the cluster version and the Grid version are available to the engineering community.

2 Cluster as a commodity tool

A cluster is a group of computers, connected via local area networks, working together in such a way that can be considered as a single computer. The formal basis of computational clusters as a way to run parallel tasks was envisioned by Gene Amdahl, who in 1967 showed mathematically the speedup that can be expected by parallelizing a job compared to its sequential execution [1]. More than 25 years later, in 1994, the first high performance computing clusters became available when Sterling and collaborators developed the Beowulf project for NASA [15]. Since then clusters had become almost a commodity tool for scientists and engineers because they can reach supercomputer performance at an affordable cost.

[5] http://portaldeporticos.ula.ve

There is currently a variety of cluster software and information on the Internet, like The Beowulf Project[6], OSCAR[7], Rocks[8], Warewulf[9] and most of the TOP500[10] list of the world's fastest supercomputers are clusters. Presently, it is even possible to customize a particular cluster configuration online [3].

In 1997 CeCalCULA was born as a joint effort by the Universidad de Los Andes (ULA)[11], the Fondo Nacional para la Ciencia y la Tecnología (FONACIT)[12] and the Corporación Parque Tecnológico de Mérida (CPTM)[13], with close collaboration from IBM Venezuela, Sun Microsystems and Silicon Graphics. In the past decade CeCalCULA had strove to provide the local and national scientific community with the required computing power, training in IT and IT consulting. During this time several kinds of clusters had been deployed, both in testing and production environments for several computer intensive applications such as data mining, database servers, Web servers, graphics rendering, to mention a few. We had also disseminated much of the experience through training courses, workshops and documentation in Spanish[14]. Next section will be devoted to describe one of these applications running on a cluster environment.

3 Cluster or Grid: To be or not to be

Computing is a process of constant evolution and history has shown these by improvements in technology, architecture, software and applications. This improvement provides greater computational capabilities for better solutions. Clusters give access to more computing power than PCs or workstations offer, allowing parallel and distributed calculations for which wide range of applications have been developed. The next step in this evolution corresponds to the Computer Grids, showing that the north is the addition of computational resources in such a way that they are perceived as a single machine.

- Resources are shared among multiple organizations.
- Are geographically distributed.
- Heterogeneous.
- Are managed by multiple organizations, therefore with different use policies.
- Demand-based access to computational power and storage.

From these characteristics a Grid can be defined as a hardware and software system that allows to share resources, data and storage among multiple organizations over the network in a way that is transparent to the final user. It uses

[6] http://www.beowulf.org/
[7] http://oscar.sourceforge.net/
[8] http://rocksclusters.org/
[9] http://www.warewulf-clusters.org
[10] http://www.top500.org/
[11] http://www.ula.ve/
[12] http://www.fonacit.gov.ve/
[13] http://www.cptm.ula.ve/
[14] http://www.cecalc.ula.ve/

software - middleware - in charge of coordinating this system among dynamic and geographically distributed organizations. These organizations are known as Virtual Organizations (VO).

4 PoD on a Cluster environment

The PoD, in its cluster version, is a web-based finite element working environment for structural analysis described in detail in[12] and depicted in Plate A, Fig. 4. It was developed under the same conditions as similar portals: secure access, data handling, process handling and execution, information services, collaborative tools and services, visualization services and data analysis.

The scientific basis for this portal, that is, the physical model on which it is based is a novel development: Theory of Concentrated Damage (see [5,2,4] and references therein). The Theory of Concentrated Damage combines fracture mechanics, continuum damage theory and the plastic hinge concept. The goal of this theory is to simulate the damage and collapse process of civil structures and industrial installations under mechanical overloads, typically earthquakes.

The PoD allows the user to numerically simulate cracking processes and collapses of reinforced concrete structures subjected to mechanical overloads, e.g. earthquake loadings. The system structure consists of 3 main modules: Pre-processor, Processor and Post-processor. The main screens corresponding to each of the modules are shown in Figure 4.

- **Pre-processor Module**: It is used to digitize the portico and the demands placed on it. This digitization describes the porticos' geometry, the demands, the properties of the cross sections and of the concrete and steel used. It is a Java module which provides the environment for building the input structure and for evaluating its load (see Plate B, Fig. 4). The user needs to create an account, digitize the portico, generate an input file and upload it into his/her account in the server, since the numerical simulation does not run on the user's computer but on the server. The preprocessor module generates a file containing the raw data and sends part of it to the generator (a piece of code that transforms these untreated data into information that can be used by the finite element simulator) to be refined. With these refined data, the preprocessor creates an input file for the analysis of the structure.
- **Processor Module**: The processor module accesses the user input files and uses the refined data file as an input to the finite element engine. Through an other Java interface the user can follow (or abort) the analysis, (Plate C, Fig. 4). This engine is a dynamic, nonlinear finite element program written in Fortran, whose physical model is based on the above referred theory of Lumped Damage Mechanics [2]. The simulator computes and quantifies the density and location of concrete cracking and reinforcement yielding a set of state variables. In particular, the concrete cracking density is described by a damage variable that takes values between zero (no damage) and one (complete concrete destruction).

- **Post-processor Module**: It allows to visualize the results with variable versus variable graphs, variable versus time and damage distribution maps at any time of the analysis. In short, it displays the structure's behavior using graphics, distributions and animations.

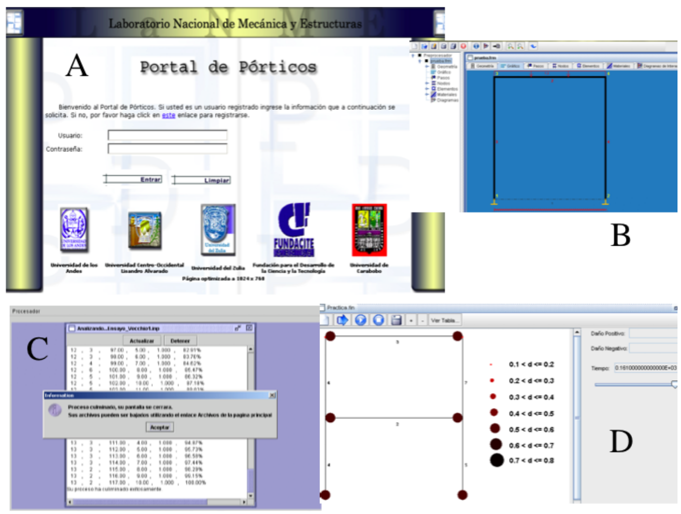

Fig. 1. Portal of Damage. Plate A: Portal of Damage Homepage http://portaldeporticos.ula.ve/. Plate B: Pre-processor. Plate C: Monitoring an analysis through the portal. Plate D: Graphic post-processor

5 Portal of Damage on Grid

The PoDs Grid architecture retains the same cluster modules for Pre-processing - Processing - Post-processing, aiming for a transparent use of the grid by the user while getting the same results.

In order to develop the PoD on Grid the GridPort Toolkit [16,17] was used. This package consists of a set of tools that enables fast development of highly functional grid portals, simplifying the use of underlying grid services by the end user. GridPort has a set of portlets, which is a Web component built in Java and managed through a portlet container that handles user queries and produces dynamic content. Users have the option to select which portlets will be executed via the Web interface. GridPort was designed for grid portals, portlets and application developers. A portal and a portlet container can be used as a single component among a set of applications or can be separated entities in a portal application.

The portal framework GridSphere [14] provides a open source Web portal. GridSphere allows developers to display Web application portlets that can be run and managed via the GridSphere's portlet container. The main reason to select GridPort is the fact that it uses the GridSphere portal framework, which offers the following features:

– The API implementation is fully compatible with JSR 168. The Java portlet specification (originally developed via JSR-168) provides a standard for the development of portal components using the Java programming language.
– Portlet development using the JavaServer Faces (JSF) standard.
– The API implementation is almost fully compatible with IBM's WebSphere 4.2.
– Support for easy development and integration of new application portlets.
– High level model to build portlets using visual "beans" and GridSphere's user interface tag library.
– Flexible presentation layer based on XML that can be easily modified to create customized portal layouts.
– Sophisticated portlet service model that can encapsulate reusable portlet logic in services that can be shared among multiple portlets.

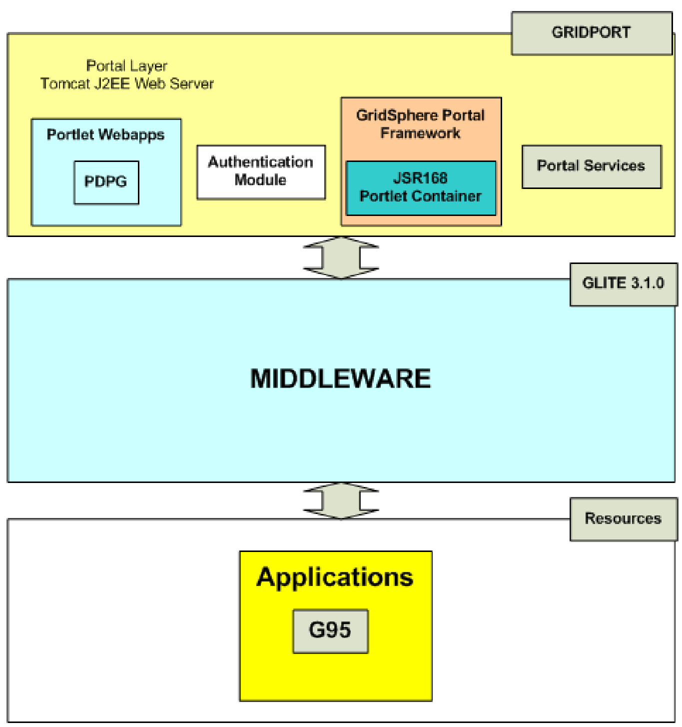

Fig. 2. Architecture of the Portal of Damage on Grid

An important characteristic of the basic GridPort portlets is that they allow single user authentication when the user logs into the portal with a username and private key. A proxy is instantly created. The basic GridPort portlets also facilitates the customization and layout of the portlets. For the PoD on Grid, GridPort and an UI (User Interface) using the gLite 3.0.1 middleware were installed in a single machine (see [11] and references therein) and some customization of the portal was necessary. The modifications required for the correct behavior of the portal were:

- Modify GridPort's code so that the jobs are sent, monitored and retrieved using the commands required by the gLite middleware.
- The portlet must generate a JDL file in order to be able to dispatch a task. The JDL file specifies the input and output files as well as the task requirements so that the task can be run using the gLite middleware.
- The portal interface was modified so that a user can send multiple jobs successively.
- Changes to the way output files were retrieved.

The portal sends the input file and the Portal of Damage executable to the Grid enabling execution in a wide range of grid sites. Figure 5 shows a graphical representation of the architecture of the Portal on Grid:

CeCalCULA had also developed other portlets for a bioinformatics portal using the same architecture, in order to give access to programs like MPQC and Gaussian.

6 Tests

In order to validate the ported application, the three reinforced concrete porticos shown in Figure 6.2 were analyzed:

The main idea when conducting these tests is not to compare between execution times in a cluster and a grid, as both the cluster and in the various sites where they were executed have different architectures, but to achieve satisfactory results in all runs.

6.1 Cluster Environment

The cluster that presently hosts and runs the PoD has 6 nodes (Pentium IV, 1GB RAM and 80GB hard disk). The cluster uses Rocks 4.2 (Hallasan), a Linux distribution – a collection of free software – that enables building, managing and monitoring computational clusters. Sun Grid Engine version 5.3 is used for queue management, and the portal uses Java, Tomcat, and Postgresql as a database.

6.2 Grid Environment

The ULA Grid is located at CeCalCULA and includes a series of services distributed over heterogeneous hardware. The hardware is described in Table 1.

There is also a Sun 3300 storage element with 1 TB installed capacity. The Operating System used accross ULA Grid is Scientific Linux version 3.0.8. The

Amount	Brand	Model	Memory	Hard Disk
4	SUN	V20z	4GB	147 GB
4	SUN	X4100	8GB	147 GB
8	White Box	Pentum IV	512MB	147 GB

Table 1. Hardware used for Grid projects.

main component in any Grid installation is the middleware. gLite 3.1.0 is the middleware used in ULA Grid. Components installed on each equipment depend on the type of node. The different types of node in ULA Grid are:

– **User Interface (UI)**: The user gets access to the Grid through it, authenticates and sends jobs through the UI, command lines or using portals.
– **Computing Element(CE)**: In charge of managing the nodes that perform the actual computation. It has a planner and a queue manager that enables it to accept and send jobs to the worker nodes.
– **Worker Node (WN)**: In charge of executing the jobs sent to the grid.
– **Storage Element (SE)**: Provides uniform access to a wide rage of storage.
– **LCG File Catalog (LFC)**: A catalog with information regarding data located in the different SEs. It also allows data replication.
– **Workload Manager System Logging and Bookkeeping (WMSLB)**: Responsible for the collection, preparation, verification and visualization of task outputs.
– **Berkeley Database Information Index (BDII)**: In charge of collecting status information produced by the different resources and notify the users of the Grid status.
– **Proxy Server (PX)**: For long-running tasks, a renewal proxy server is used, comprising:
  - Proxy Renewal Service (PRS) over the WMSLB.
  - Proxy Server over a dedicated machine.
– **Virtual Organization Membership Service (VOMS)**: Authentication based on a central DB. One per Virtual Organization. The DB is accessed by the WMSLB, CE and SE to build a list of locally authorized users. ULA Grid has a virtual organization for local users.
– **Certification Authority (CA) and Registration Authority (RA)**: In charge of generating and signing the ULA user certificates.

Jobs sent to the Grid did not specify any particular resource to run on, therfore used different sites with different resources and grid middleware versions used in the project EELA.

As it can be seen on Figure 6.2, for each one of the test cases the number of elements and nodes was increased, so that performance comparisons can be done between small, medium and large job size. Each one of the tests was submitted 10 times to the cluster, to the ULA Grid using the ULA Virtual Organization and to the European grid using the EELA Virtual Organization. Table 2 shows the best and average CPU time for the grid and the cluster runs.

It can be seen that:

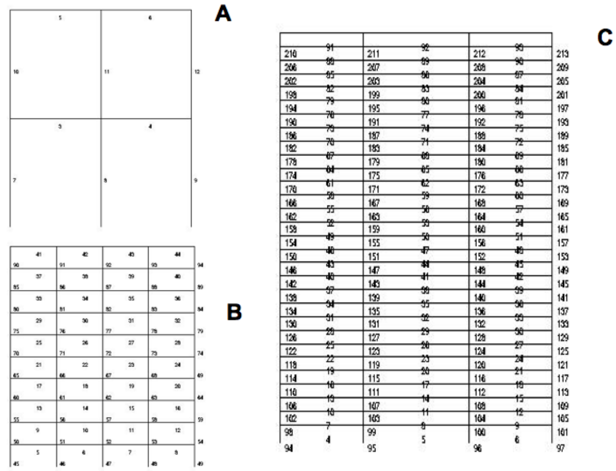

Fig. 3. Plate **A** 2-story Portico: 10 elements / 9 nodes. Plate **B** 10-story Portico: 90 elements / 55 nodes. Plate **C** 2-story Portico: 210 elements / 124 nodes.

Portico	Cluster	Best ULA	Best EELA	Average ULA	Average EELA
2-story	437.743	236.77	193.800	259.763	245.383
10-story	3101.666	1709.860	1732.794	1853.2812	1907.809
30-story	9885.170	4902.440	3442.150	5431.449	4926.040

Table 2. CPU time of running Portico problems over cluster and grid.

– All the results correspond to expected ones in both clusters and the grid.
– CPU time on the cluster is almost double than CPU time on ULA or EELA grids. This is due to the limited computing capacity of the cluster.
– Average ULA Grid runtimes are quite close to the EELA sites runtime.
– Even though there are some variations on the grid run times, it presents an attractive target to run these kind of jobs due to increased availability.

7 Conclusions and Future work

The Portal of Damage system is an on going project based on a finite element engine to evaluate cracking risks for reinforced concrete structures in two dimensions (2D). This program presents a simplified view of the structures behavior. In order to achieve a better understanding of the internal effects over the many structural components as a result of the internal and external actions, a three dimensional (3D) analysis is required. Future work on the portal would be geared towards the development more specific test on different 2D models and to develop an extended version on a three dimensional web-based computational tool for the non-linear analysis of structures, capable of coupling existing finite elements with user subroutines, and with the goal of simulating the damage of civil and mechanical engineering structures under overload conditions. It is also important to be able to use efficient and complete models from the computational point of view, being the Grid as it is, a powerful tool that enables the timely completion of the computations generated by these kinds of applications.

The Grid empowers user by offering an alternative environment where they can run jobs for porticos' analysis, whether it is in a local cluster or on the Grid. Is important to note that – from the user perspective – there is no difference in using one or the other, aside from the possible difference in run time. This was accomplished by the integration of GridPort and the original cluster portal. A critical point is that the Grid version can send jobs to different sites, significantly increasing the computing power available to the end user and the system's reliability.

Acknowledgements

One of us (LAN), gratefully acknowledge the financial support of the Fondo Nacional de Investigaciones Científicas y Tecnológicas under projects S1-2000000820 and F-2002000426.

References

1. G. M. Amdahl. *Validity of the single processor approach to achieving large scale computing capabilities*, pages 79–81. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000.
2. A. Cipollina, A. López-Inojosa, and J. Flórez-López. A simplified damage mechanics approach to nonlinear análisis of frames. *Comp. & Struct., Struct.*, 54(6):1113 – 1126, 1995.
3. W. Dieter and H. Dietz. Designing a cluster for your application. *Computing in Science and Engineering*, 9(4):72–79, Jan 2007.
4. J. Flórez-López. Frame analysis and continuum damage mechanics. *European Journal of Mechanics - A/Solids*, 17(2):269–283, Sep 1998.
5. J. Florez-Lopez, A. Benallal, G. Geymonat, and R. Billardon. Two-field finite element formulation for elasticity coupled to damage. *Computer Methods in Applied Mechanics and Engineering*, 114:193–212, Apr. 1994.
6. I. Foster. Service-oriented science. *Science*, 308:814–817, May 2005.
7. I. Foster, C. Kesselman, J. Nick, and S. Tuecke. The physiology of the grid. In F. Berman, G. Fox, and T. Hey, editors, *Grid Computing: Making the Global Infrastructure a Reality*. Wiley Interscience, 2003.
8. T. Hey and A. E. Trefethen. e-science and its implications. *Phil. Trans. R. Soc. Lond. A*, 361:1809–1825, 2003.
9. T. Hey and A. E. Trefethen. Cyberinfrastructure for e-science. *Science*, 308:817–821, May 2005.
10. R. T. Kouzes, J. D. Myers, and W. A. Wulf. Collaboratories: doing science on the internet. *Computer*, 29:40–46, 1996.
11. E. Laure, S. Fisher, A. Frohner, C. Grandi, and P. Kunszt. Programming the grid with glite. *Computational Methods in Science and Technology*, Jan 2006.

12. M. E. Marante, L. Suárez, A. Quero, J. Redondo, B. Vera, M. Uzcategui, S. Delgado, L. R. León, L. N
'uñez, and J. Flórez-López. Portal of damage: a web-based finite element program for the analysis of framed structures subjected to overloads. *Advances in Engineering Software*, 36(5):346 –358, May 2005.
13. J. Marco de Lucas. A Grided World? *Journal of Physics: Conference Series*, 53:397–412, 2006.
14. J. Novotny, M. Russell, and O. Wehrens. Gridsphere: a portal framework for building collaborations. *Concurrency and Computation: Practice and Experience*, 16(503-513), 2004.
15. T. L. Sterling, D. Savarese, D. Becker, J. E. Dorband, U. A. Ranawake, and C. V. Packer. Beowulf: A parallel workstation for scientific computation. In *ICPP (1)*, pages 11–14, 1995.
16. M. Thomas, S. Mock, M. Dahan, K. Mueller, D. Sutton, and J. R. Boisseau. The gridport toolkit: A system for building grid portals. In *10th IEEE International Symposium on High Performance Distributed Computing (HPDC-10 '01)*, volume 00, page 0216, Los Alamitos, CA, USA, 2001. IEEE Computer Society.
17. M. P. Thomas and J. R. Boisseau. Building grid computing portals: The npaci grid portal toolkit. In F. Berman, G. Fox, and T. Hey, editors, *Grid Computing: Making the Global Infrastructure a Reality*. Wiley Interscience, 2003.