

The PDCLab Grid Testbed at UPRM

Kennie Cruz, John Sanabria, Fernando Cintron, Wilson Rivera

Parallel and Distributed Computing Laboratory
University of Puerto Rico at Mayaguez
P.O.Box 9042, Mayaguez, Puerto Rico 00681, USA

Abstract

The Parallel and Distributed Computing Laboratory (PDCLab) at the University of Puerto Rico, Mayaguez has deployed an experimental grid testbed to perform research in the area of grid computing. The PDCLab grid testbed was deployed using components that allow flexible re-configuration, management and programmability. This paper provides discussion about the hardware and software configurations of the grid testbed and the research issues being investigated.

Keywords

grid computing, grid testbed deployment, adaptive scheduling, replication

1. Introduction

Grid computing (Foster and Kesselman, 1998) involves coordination, storage and networking of resources across dynamic and geographically dispersed organizations in a transparent way for users. The Open Grid Services Architecture (OGSA) (Foster et. al., 2002), based upon standard Internet protocols such as SOAP (Simple Object Access Protocol) and WSDL (Web Services Description Language), is becoming a standard platform for grid services development. Operational grids based on these technologies are feasible now, and a large number of grid prototypes are already in place (e.g. Grid Physics Network (GridPhyN)¹ and Teragrid² among many others).

Despite the recent advances in grid computing deployment, still there are research challenges. These problems include dynamic scheduling to achieve Quality of Service (QoS) and integrating sensor networks to grid infrastructures. The PDCLab grid testbed, deployed at the University of Puerto Rico at Mayaguez, is an experimental grid designed to address the afore-mentioned research issues. The next sections in this paper provide a discussion related to hardware and software configurations of the grid test-bed and the research issues being investigated.

¹ <http://www.griphyn.org/>

² <http://www.teragrid.org/>

2. The PDCLab Grid Testbed Hardware and Software Specifications

The PDCLab grid testbed aggregates a number of heterogeneous resources including a cluster of 65-dual processor nodes and 8-dual processor Itanium-based nodes (see Figure 1). The hardware specifications of the PDCLab grid testbed are listed as follows:

- A Linux Beowulf Cluster that consists of 65 2-Way SMP Intel Pentium III at 1.2GHz with 1 GB of Memory.
- Eight (8) IA-64 Itanium servers (each server is dual processor at 900 MHz, 8GB of memory and 160GB of SCSI Ultra 320 storage).
- Two (2) IA-32 Pentium IV servers (each server is dual processor at 3.06 GHz, 1GB of memory and 160GB of ATA-100 storage)
- One (1) IA-32 Pentium III server (dual processor at 1.2 GHz, 2GB of memory and 40Gb of SCSI Ultra 160 storage)
- One (1) Intel Xeon server (dual processor at 3.60GHz, 2 GB of memory and 2TB of storage)
- One (1) Intel Xeon server (dual processor at 2.80GHz, 1 GB of memory and 200 GB of storage)



64 nodes Linux Cluster (IA-32)



8 nodes Itanium Servers (IA-64)

Figure 1: PDCLab Grid Testbed Hardware

The heterogeneous nature of resources is an important issue since it poses a number of administrative and performance considerations. For example, configuration and deployment are quite different for Itanium based resources versus i-32 based resources. In terms of execution of applications it is difficult to hold transparency when submitting jobs to the grid. Applications targeting i-64 Itanium based resources often require extra tuning efforts to achieve performance (Lugo et. al., 2004). As a consequence, an important effort in our research plan has been the development of tools to facilitate transparent access to these heterogeneous architectures.

The PDCLab grid testbed components run CentOS 4.2³ and the Globus Toolkit 4.0.1⁴. The Globus Toolkit includes a basic installation of Java WS Core and base grid services such as a security infrastructure (GSI), data transport service (GridFTP), execution services (GRAM), and Information services (MDS). Software associated to the pre-installation of Globus includes: OpenPBS, PostgreSQL, Apache Ant version 1.6.5, Java SDK version 1.5 and Jakarta Tomcat version 5.5.9. A complete guide of installation is available at the PDCLab Grid Portals⁵

³ <http://www.centos.org>

⁴ <http://www.globos.org>

⁵ <http://pdcsrv.ece.uprm.edu>

The PDCLab grid testbed is also connected to other non grid based resources (see Figure 2). For example, raw data from sensors may be sent to a data server via wireless communication. GridFTP is used to improve data transport from the data server to the PDCLab grid testbed. Data exchange between server and the grid testbed is authenticated using Grid Security Infrastructure (GSI).

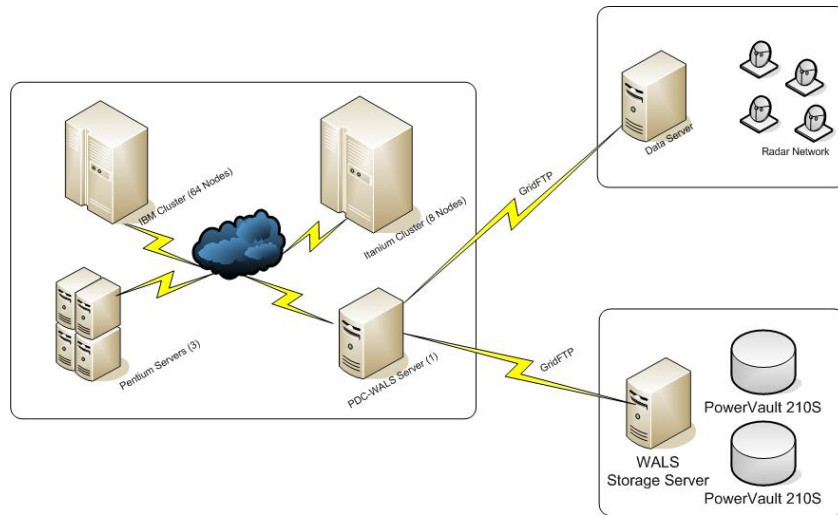


Figure 2: PDCLab Grid Testbed Connectivity

We have developed a customized in-the-box distribution for grid deployments based on CentOS4.0 and Globus Toolkit 4.0.1. This package of scripts has been developed to facilitate and speed up the configuration and installation of grid nodes and clients. Table 1 summarizes some of these scripts.

Table 1: Configuration and Installation scripts

Script	Description
addgridnode.sh	Add nodes to our grid-node file database
addgriduser.sh	Create user accounts on the grid
gt-preinstall.sh	Manage the installation of the globus toolkit requirements
gensshkeys.sh	Create SSH keys
centos-config-common.sh	Configure a minimal grid node with CentOS
centos-config-server.sh	Configure a minimal grid node with CentOS (server edition)
OpenPBS-Client-Setup.sh	Install and configure a node as an OpenPBS client
OpenPBS-Server-Setup.sh	Install and configure an OpenPBS server
Torque-Client-Setup.sh	Install and configure a node as an Torque client
Torque-Server-Setup.sh	Install and configure a Torque server
makesshgkh.sh	create a global known hosts file for SSH

Although applications can be built using basic grid services, this low-level activity requires detailed knowledge of protocols and component interactions. In contrast, grid portals hide this complexity via easy-to-use interfaces, creating gateways to computing resources. An effective grid portal provides tools for user authentication and authorization, application deployment, configuration and application execution, and management of distributed data sets.

The Open Grid Computing Environments (OGCE)⁶ portal software is the most widely used toolkit for building reusable portal components that can be integrated in a common portal container system. The OGCE portal toolkit includes X.509 Grid security services, remote file and job management, information and collaboration services and application interfaces. The OGCE portal toolkit is based on the notion of a “portlet,” a portal server component that controls a user-configurable pane in the user’s web browser. A portal server supports a set of web browser frames, each containing one or more portlets that provide a user service. This portlet component model allows one to construct portals merely by instantiating a portal server with a domain specific set of portlets, complemented by domain-independent portlets for collaboration and discussion. Using the toolkit, one wraps each grid service with a portlet interface, creating a “mix and match” palette of portlets for portal creation and customization.

Grid portals related to specific research projects have been developed by PDCLab researchers. The PDCGrid Testbed Portal⁷ and the Student Testbed portal⁸ are instances of this effort.

3. Research and Development Issues

The PDCLab grid testbed has been thought to provide an easy-to-use infrastructure with flexibility to plug in new resources. To achieve this goal we have deployed a number of tools to facilitate administrative and end-user utilization via a package of scripts for configuration and installation and grid portals to access resources and services. To complement the spectrum of work in grid computing technologies we have developed specific research ideas including adaptive scheduling and data replication mechanisms. The ultimate goal is to apply these ideas in our grid infrastructure and extrapolate them to other grid based infrastructures. A natural direction of development is to deliver the adaptive scheduling and replication strategies developed by PDCLab as a package of grid services on top of Globus toolkit 4.0.1.

3.1 Adaptive Scheduling

We have developed an adaptive scheduling algorithm, referred to as QB-MUF algorithm (Lozano and Rivera, 2006), to provide quality of service for wide area large scale applications. We assume that the resources are connected via two-level hierarchical networks. The first level is a wide area network that connects local area networks at the second level. Users submit job specifications with their QoS requirements. The scheduler then discovers appropriate resources for processing the job and schedules the tasks on the resources. In order to discover suitable resources, the scheduler has to predict execution times on the available resources and verify QoS capabilities and availability of the resources. Re-scheduling mechanisms are then implemented to adapt scheduling to service dynamics. The scheduling strategy focuses on providing high priority to jobs with low probability of failure. To achieve this, an urgency criterion is introduced to account for relevance, laxities and probability of failures of incoming jobs. The proposed urgency criterion is a combination of one static parameter and two dynamic parameters.

Figure 3 shows the order of execution of jobs for the QB-MUF algorithm with respect to two other scheduling approaches: The Minimum Laxity First, denoted as Laxity, and the First In First Out (FIFO) scheduling algorithm. Notice that for QB-MUF jobs with high QoS deliveries are first executed. Experimental results show also a reduction of waiting processing time of the QB-MUF over laxity and FIFO approaches

We are currently working on the deployment of this scheduling strategy as a grid service on top of Globus toolkit 4.0.1. To complement this idea we are also working on the problem of how multiple services should be orchestrated in a grid environment to provide adaptive functionalities. The need for adaptation

⁶ <http://www.ogce.org>

⁷ <http://pdcsrv.ece.uprm.edu:8080/gridsphere/gridsphere>

⁸ <http://136.145.59.3:8080/gridsphere/gridsphere>

in grid infrastructures arises due to both resource and service demand uncertainty. Next generation of grid middleware must provide mechanisms to efficiently deal with uncertainty. Several key issues in this problem space will be addressed to evolve, scale and respond to unpredictable service demands and events. An example is the development of adaptive resource management middleware that dynamically decides how many resources to allocate to a request and where a request should run. Such a middleware will allow for network sensor and application models Quality of Service re-negotiation and support adaptation at multiple levels.

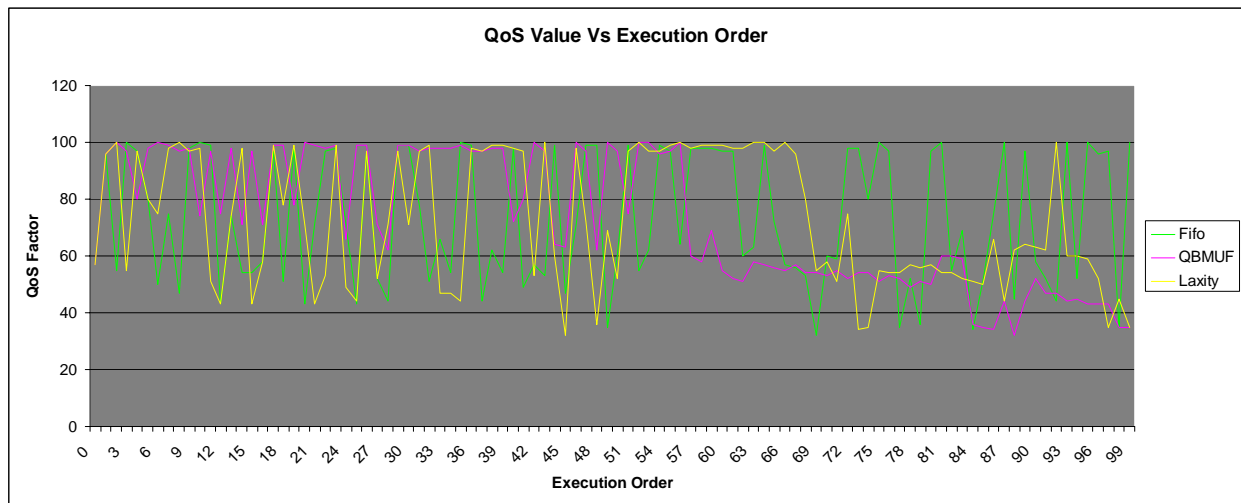


Figure 3: Quality of Service Guided Execution Order; jobs=100; arrival rate=0.35

3.2 Integrating Sensor Networks to Grid Infrastructures

The integration of grid computing and sensor network technologies enables the complementary strengths of these technologies to be realized in an integrated platform. However, it poses several challenges such as the need to comply with emerging APIs for grid and Web services, the coordination of communication, and the requirement of a more data-centric infrastructure focused on distributed services. Preliminary experiments demonstrate the feasibility of such interaction, when independent and non grid based applications can be integrated to the grid infrastructure with minimum requirements. A large amount of data was transported using GridFTP protocol with GSI support, and the integrity of the data was preserved successfully. We have implemented an Information dispersal algorithm to perform distributed data management of data acquired by sensor networks.

The proposed Information dispersal algorithm (Arias and Rivera, 2006) shows a better access reliability than the traditional replication algorithm. As a reference point, for an access reliability $R = 0.9$ when the probability of failure is $p = 0.4$, $m = 5$, the added redundancy for IDA is $AR = 120\%$, while in the replication approach the added redundancy must be approximately $AR \approx 300\%$ (Figure 4(a)). Note that, for replication algorithm, AR increment is every 100%, because the redundancy is performed using multiplication with integer numbers. Figure 4(b) shows the behavior of the algorithms when the probability $p = 0.6$ and $m = 16$. The reliability of replication approach is quite deficient if the probability failure increments.

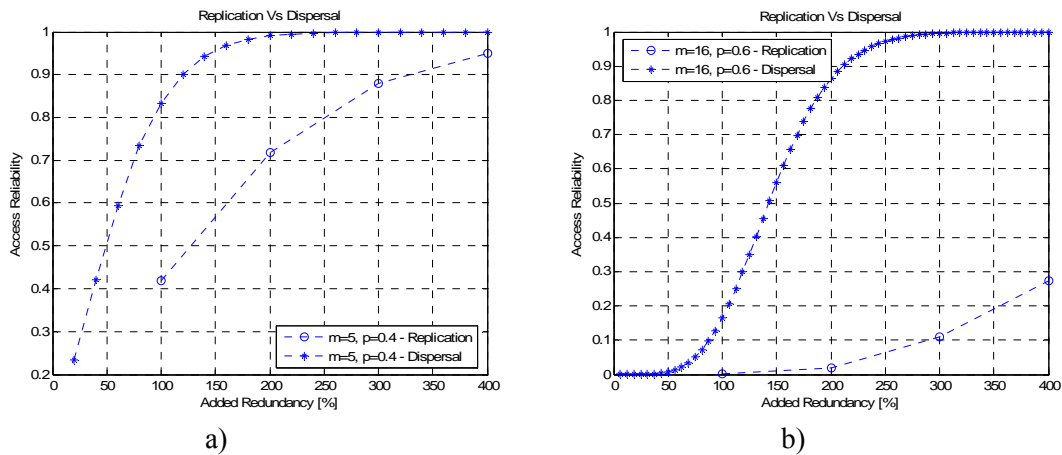


Figure 4: Reliability vs Added Redundancy comparison. a) $m=5, p=0.4$, b) $m=10, p=0.6$

4. Conclusions

The PDCLab grid testbed is an experimental deployment of grid computing technologies. To provide an easy-to-use infrastructure with flexibility to plug in new resources we have deployed a number of tools to facilitate administrative and end-user utilization via a package of scripts for configuration and installation and grid portals to access resources and services. We have developed also specific research ideas targeting adaptive scheduling and data replication that are in progress to be delivered as grid services on top of Globus toolkit 4.0.1.

References

- Foster and C. Kesselman (1998), "The grid: blueprint for a future computing infrastructure." *Morgan-Kaufmann Publishers*
- I. Foster, C. Kesselman, J. Nick, and S. Tuecke (2002), "The physiology of the Grid: An open Grid services architecture for distributed systems integration, *Technical report, Open Grid Service Infrastructure WG, Global Grid Forum.*
- W. Lugo-Beauchamp, C. Carvajal-Jimenez and W. Rivera (2004), "Performance of hyperspectral imaging algorithms on IA-64. *Proc. IASTED International Conference on Circuits, Signals, and Systems*, pp. 327-332.
- W. Lozano and W. Rivera (2006), "An adaptive quality of service based scheduling algorithm for wide area large scale problems. *To appear in IEEE Workshop on Adaptive Grid Computing*
- D. Arias and W. Rivera (2006), "Using grid computing to enable distributed radar data retrieval and processing". *To appear in IEEE International Conference on Network Computing and Applications.*

AUTHORIZATION AND DISCLAIMER

Authors authorize LACCEI to publish the papers in the conference proceedings. Neither LACCEI nor the editors are responsible either for the content or for the implications of what is expressed in the paper.