GridForce: A Comprehensive Model for Improving Technical Preparedness of our Workforce for the Grid

Bina Ramamurthy
CSE Department
University at Buffalo (SUNY)
bina@cse.buffalo.edu
http://www.cse.buffalo.edu/gridforce
Partially Supported by NSF DUE CCLI A&I Grant 0311473
Topics for Discussion

- GridForce Project
- Adaptation of Grid Technology to CS Curriculum
- Courses: Curriculum CSE4/586, CSE4/587, non-majors course
- Lab Exercises: problems and approaches to solution
- Pedagogical Issues
- Technologies and tools covered
- Grid infrastructure
  - Reusing old hardware (SparcGrid)
  - Grid with newer hardware (LinuxGrid)
- Challenges in Adaptation
GridForce: Grid For Research, Collaboration & Education

Hands-on Labs

Sample labs:
Grid Services Lab
Design and implementation of Grid services with standard capabilities

Virtual Organization (VO) Lab
Grid application design and implementation
Symbolic representation of VO components
Grid-based tax return filler

Courses/ Curriculum:
CSE486/586: Distributed Systems
CSE487/587: Information Structures

Dissemination Package:
Syllabus, Lecture Notes, Exams, Course Evaluations, Pedagogy, Applications, Lab descriptions, Solutions, Publications, and Infrastructure details.

Assessment Sample from Fall 2003
CSE486/586

Effectiveness of Grid Coverage

<table>
<thead>
<tr>
<th>Rating (1-best, 5-worst)</th>
<th>No. Students Avq Across Questions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
</tr>
</tbody>
</table>

Research Infrastructure

LinuxGrid Globus infrastructure supporting secure service oriented architecture

Staging Server
OS: Solaris 8.0
Grid: Globus 3.0.2
Function: Debug and test services

Production Server
OS: I
Grid: Globus 3.0.2
Function: Deploy services

CSECCRGGrid Collaboration with Center
For Computational Research (CCR); Reusable old Sparcs offering Condor grid and NSF Middleware Initiative suite.

Collaboration (SUNY Geneseo)

Labs Illustrating Grid Use for Non-CS Majors

Compute nodes: Data analysis and graph tools

Internet: Job submission, scheduling tools

Sample from Fall 2003
CSE486/586
Bottom line

What do we do in the courses?
How do we do it?
Are students benefiting?
What is the impact of this work?
What are some of the major issues we are facing?
How can we address these issues?
Please understand GridForce is a work in progress.
Deliverable: package that will help teachers mount a course (s) on grid, and practitioners get started on the emerging grid technology.
Adaptation of Grid Technology to CS-Curriculum

Introduce grid technology into the CS undergraduate curriculum.
- Goal: Able to design and deploy grid services and applications. Study grid application models.
- Focus on lab exercises to illustrate fundamental grid concepts, and development of grid services and applications.

Introduce grid to potential users of grid
- Goal: Publicize usage models of grid.
- Provide hands-on experience of existing grid applications for entry level courses in Sciences and Engineering.

Conduct seminars to industry.
- Goal: Overview of grid technology landscape and its alignment to common technologies and application models.
- Examine case-studies to expose potential applications of grid and publicize resources available for building grid-based applications.
Courses:

CSE4/586 Distributed Systems

- Learning outcome: fundamental concepts of distributed systems and grid.
- Lab exercises to support concepts:
  - Three-tier client server system using Web Services.
  - A simple grid framework.
  - Design and implementation of a grid service.
- Prerequisites: Data structures and algorithms, object-oriented design and development, working knowledge of Java.
### Courses:

#### CSE 4/586: Lab Exercises

<table>
<thead>
<tr>
<th>Exercise</th>
<th>Topic</th>
<th>Learning Objective</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lab1</td>
<td>Webservices</td>
<td>To understand the alignment of the grid technology to Web Services, WS Definition Language (WSDL) and service description using WSDL.</td>
</tr>
<tr>
<td>Lab2</td>
<td>Grid Infrastructure</td>
<td>A Webservices based grid.</td>
</tr>
<tr>
<td>Lab3</td>
<td>Grid Programming</td>
<td>Design and implement a grid-base service using Globus 3.0.2</td>
</tr>
</tbody>
</table>
Lab1: Web Services

1. National Weather Service Web Site
2. RMI Weather Server
3. RMI Client
4. Relational Database
5. Weather Web Service Container/Server

Connections:
- National Weather Service Web Site to RMI Weather Server via IP Socket API and http.
- RMI Weather Server to Relational Database via LAN.
- Relational Database to Weather Web Service Container/Server via Oracle 9i and SOAP XML.
Lab2: Web Services-based Grid

Based on A simple do-it-yourself framework for grid computing by Anthony Karre
1. SAAJ: SOAP with Attachments API for Java
Lab3: Grid Service using Globus

- **Logging**: Logger object; Levels of logging: Info, Warn, Error, Fatal. Filtering and redirecting to file, console.
- **Secure**
- **Notification**: Explained in next slides
- **Basic Service**
- **Routable**: Services with ability to migrate
- **Persistent**
- **Service Data**: Stores service properties and states; for discovery, monitoring, negotiations, etc.
- **Permanent services such as naming service that get activated and terminated with the container**
Courses:

CSE4/587: Information Structures

- Learning outcome: understand grid infrastructure and grid architecture, design and deploy grid services and grid applications.
- Lab exercises support:
  - Grid application in high performance area.
  - Service-oriented grid application.
- Prerequisites: Data structures and algorithms, object-oriented design and development, working knowledge of Java, fundamentals of client/server architectures.
# Courses:
## CSE4/587: Lab Exercises

<table>
<thead>
<tr>
<th>Exercise</th>
<th>Topic</th>
<th>Learning Objective</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lab1</td>
<td>Commercial Application</td>
<td>Study requirements of a commercial domain and implement an application. Implemented in J2EE.</td>
</tr>
<tr>
<td>Lab2</td>
<td>High performance Application.</td>
<td>Study requirements of scientific/business domain and implement compute intensive application. Markovitz model for gene expression analysis.</td>
</tr>
<tr>
<td>Lab3</td>
<td>Defining a high-level grid application</td>
<td>Virtual organization (VO): IRS Return Filer VO</td>
</tr>
</tbody>
</table>
Concepts illustrated: Virtual organization (VO) called IRS/Tax Filer that brings together virtualized capabilities of physical organizations of banking, personal profiles, and employment. Grid service handle (GSH) and Grid service reference (GSR), registry and handlemap, discovery of services, index services, application of notification, logging. High trust computing and social imperative of grid are also discussed.

6/29/2005
CCGrid2004
Pedagogical Issues

- Approaches used in grid books and online tutorial (Ex: Satomayer’s GT3 tutorial) are inadequate/unsuitable for teaching grid concepts.
- Whereas large amounts of information is available it is incredibly complex for a newcomer to the area.
- Role of an educator is to preprocess the data and to present it in a palatable form.
  - Learn and Teach
- As examples lets consider two representative issues one at container level and the other at service level:
  - Teaching “Notification” feature of the grid services
  - Container services: Container services; declarative vs programmatic
Notification

- **Foundational concepts:** messaging, queues, source and sink for messages, subscription model, loose coupling, push and pull notification
- **Grid related concepts:** Service data element (SDE), OGSINotification API
- Implement a service that is a producer of notification.
- Implement a client application that invokes a service that produces notification; an associated listener that consumes the notification.
- Extend the application to illustrate push and pull model.
Notification Explained

Notes:
Example: Grid service (GS) can be a Math Service with notifyChange to SDE on invocation of add Subtract methods.
GWSDL file: extends="ogsi": GridServiceogsi:NotificationSource (declarative vs programmatic)
Listener has: NotificationSinkManager to which is added a listener to Math Service's GSH and SDE.
Listener has deliveryNotification() method to process notification.
Push vs Pull

Pull model
Request-response under client’s control
Client can choose to ignore notification

Push model
Service data is pushed to the client along with notification
Container Services

- The concept of grid container and persistent services offered by the container along with descriptors (xml files) is very hard to get across without a proper platform.
- To solve this problem we introduced J2EE (Java 2 Enterprise Environment) and container resource specification such as JDBC and Queues. Students work with real J2EE graphical development environment to understand the concept of container.
- The ideas are later referenced when ever we discuss grid container.
- A permanent solution is needed: we need a grid application development environment.
Tools and Technologies covered

Technologies include:
- XML and SOAP
- Web services (service definition, implementation and deployment)
- Java 2 Enterprise Edition (Enterprise Java Beans)
- Globus Toolkit 3.0.2 (GT3)
- CONDOR

Tools include:
- UML (Unified Modeling Language) for design representation
- Apache Ant: XML-based build tool
Effectiveness of Adaptation (CSE486/586)

- Survey with 42 multiple choice questions pertaining to coverage of grid in CSE4/586.
- Average rating among 20 students who took the survey is shown.

External evaluator identified 7 areas for improvement.
- Two of these pointed to unavailability of grid programming infrastructure for students to use.
- We have remedied this situation with more than one grid lab infrastructure.
Infrastructure: LinuxGrid

Goal: To facilitate development of service-oriented applications for the grid.

Two major components: Staging server and Production grid Server.

Grid application are developed and tested on staging server and deployed on a production server.

Production grid server:

- Three compute nodes with Red Hat Linux and Globus 3.0.2 instance.
- One utility gateway node with Free BSD and Globus 3.0.2.
Development Environment

Staging Server

OS: Solaris 8.0
Grid: Globus 3.0.2
Function: Debug and test services

Production Server

OS: FreeBSD
Grid: Globus 3.0.2
Function: fileserver, firewall

OS: Red Hat Linux 9.2
Grid: Globus 3.0.2
Function: Deploy services

Cerf
Postel
Mills
Vixen

6/29/2005
Infrastructure: SparcGrid

- **Goal**: To run jobs submitted in a distributed manner on a Condor-based computational cluster Condor.
- **Composed of**: 50 Sun recyclable used Sparc4 machines, which form computational nodes, headed by a front-end Sun server.
- **Installation scripts**: custom-written facilitating running of jobs in a distributed manner.
- **Partially supported by**: Center for Computational Research (CCR).
CSECCR Grid

Compute nodes: Data analysis and graph tools

Gatekeeper: Job submission scheduling tools

Students work on computationally intensive application. Gene expression analysis; Markovitz model for financial portfolio picking; Students also learn About Condor, job submission, scheduling, building grid infrastructure out of discarded computers.

MySQL Database

Internet: remote job submission
SparcGrid Monitor (Ganglia)
Industrial Training

In collaboration with The Center for Industrial Effectiveness (TCIE) of University at Buffalo, we decided to offer a industrial training (breakfast) seminar on grid (free of cost).

Poor response since audience targeted were small industries who were struggling with even their basic IT needs.

More over the seminar was advertised as High Performance Computing.

We plan revisit the seminar in Summer or Fall 2004.
Challenges in Adapting Grid Technology

- Adding to existing curriculum.
  - Solution: Addressed through labs.
- Adapting to versions of software and toolkits.
  - Solution: Discusses differences; however work with the latest version.
- Managing students with deficiencies in their technical background.
  - Solution: Special coverage during recitations.
- Maintaining grid infrastructure for hands-on labs.
  - Solution: Include a system administration support person in the implementation of curriculum.
- Lack of appropriate text books:
  - Solution: Good area for anybody with expertise to write a book.
- Lack of development environment:
  - Development is incredibly complex. Reference implementations are evolving.
- Coverage:
  - How to expand the coverage from core services on to base services and higher level services?
  - Then on to solving many research issues and get involved in contributing to open source and GT3 standardization?
  - Is top-down approach to grid is better than bottom-up approach we used?
Feedback from NSF & AAAS Conference

- NSF Directors (other than NMI people) were curious to know about “grid”
- Some of CS professors (grantees) were curious about grid, however were skeptical about what it can do. (“Is it a hype?”)
- Some are considering grid enabling (similar to how “web enabling” was a fad once upon a time).
- I met others who were wondering how to handle large dynamic datasets (ex: astronomy, geology)
- Others are quite happy with Internet. “Why grid?”
- Can we arrange for a informative workshop?