

Lecture 12 : Grids and Cluster Management

Dr. John Wallin

November 28, 2007

Grids

- ▶ grids are the natural extension of Beowulf clusters
 - ▶ many machines or clusters
 - ▶ connected between rooms
 - ▶ between buildings
 - ▶ between sites
 - ▶ across the Internet
- ▶ the software complexity of grids is VERY high

Grids vs Clusters

- ▶ clusters
 - ▶ generally homogeneous computers
 - ▶ small set of users with open access to all machines
 - ▶ low level tools for development and running of user developed codes
- ▶ grids
 - ▶ heterogeneous computers
 - ▶ large/huge user community
 - ▶ limited access to grid resources
 - ▶ generally high level tools

The Software Development Cycle

- ▶ prototype
- ▶ experimental code
- ▶ small scale production use
- ▶ widespread usage - large user community
- ▶ ubiquitous tool
- ▶ legacy code

Grids

- ▶ grids are best used for large, distributed user communities
 - ▶ widespread, computationally intense use of selected codes
 - ▶ access to data from a variety of sensors/sources
 - ▶ common goals within the user community

Grid Issues

- ▶ security
- ▶ monitoring
- ▶ computation
- ▶ data
- ▶ collaboration

Security

Grids

- ▶ if you have a huge user community, how do you keep the system secure?
- ▶ if you have a system spread across the Internet, how do you manage accounts, state and access to resources?
 - ▶ this is NOT a single machine, but a loose confederation of machines with different operating systems and configurations
 - ▶ how does a single account get you access many different machines?

Web Evolution

Portal Technologies

- ▶ 1990 - HTML
- ▶ 1993 - CGI
- ▶ 1995 - Java Applets
- ▶ 1997 - Java Servlets
- ▶ 2003 - Java Portlets - JSR 168
- ▶ 2003 - WSRP - Web Services for Remote Portlets (OASIS Technical Committee)

Web Evolution

Grid Technologies

- ▶ 1997 - Globus toolkit
- ▶ 2002 - Web Services
- ▶ OGSI - Open Grid Services Infrastructure
- ▶ OGSA - Open Grid Services Architecture*

* NOT the Olathe Girls Softball Association - the first hit on Google for OGSA

Standards Development

Grids

- ▶ Document standards - HTML into XML
- ▶ Grid Standards - Globus into OGSA
- ▶ Modular web standards - Servlets into Portlets
- ▶ Portlet standards - JSR 286

Definitions

- ▶ Portal - a web-based application that can be customized by the end-user both in terms of look/feel and content and applications
- ▶ Gateway - a portal dedicated to providing specific services to a specific user community
- ▶ Web container - something that runs servlets (Apache Tomcat)
- ▶ Portlet container - something that runs portlets (Apache Pluto)
- ▶ SOAP - simple object access protocol for exchanging XML messages over the Internet
- ▶ SAML - security assertion markup language - a framework for authentication and authorization

More Definitions

- ▶ Web Services (WS) - open standards-based (SOAP, HTTP, XML) web apps that are module, distributed and dynamic
 - ▶ allows easy connectivity between very software applications on different systems
 - ▶ easy reuse of applications
 - ▶ use of open standards
 - ▶ but... open standards are still immature
 - ▶ textual formats (XML) are more inefficient than other alternatives (CORBA, etc)
- ▶ WSDL - Web Services Definition Language- XML grammar to let a web service describe itself to clients
- ▶ UDDI - Universal Description, Discovery, Integration - XML format and API for searching through existing data using SOAP

Even More Definitions

- ▶ WSRP - Web Services for Remote Portlets - a presentation-orient spec aimed at portals/portlets
 - ▶ allows portal creation to be simply joining together elements with very little actual programming
 - ▶ encourages standard portlet and portlet reuse
- ▶ SOA - Service Oriented Architecture - a type of enterprise architecture that allows the creation of applications by combining loosely coupled and interoperable services

Servlets vs Portlets

- ▶ Portlets return fragments of web pages, servlets return complete web pages
- ▶ A single URL points to a set of Portlets
- ▶ Communication between the web client and portlets goes through the portal
- ▶ Portlets have scope of application and portlet, and are have persistent configurations
- ▶ Otherwise, very similar software— both run in Java and generate static or dynamic content

Portlets

Grids

- ▶ Defined by JSR 168/286
 - ▶ defines portlet lifecycle management
 - ▶ how to bundle portlets
- ▶ Standardized Java components that can be put together to make a portal page
- ▶ Portlets run inside JSR 168 compliant containers
 - ▶ Apache Pluto
 - ▶ Gridsphere
- ▶ Portlet containers run inside servlet containers
 - ▶ Apache Tomcat

Some tools

- ▶ Java community grids toolkit (COG kit)
 - ▶ <http://wiki.cogkit.org>
 - ▶ abstraction layer to hide Globus middle-ware
- ▶ Open Grid Computing Environment
 - ▶ <http://www.collab-ogce.org/>
 - ▶ Provides bundled portlets - for file transfer, collaboration, job submission, etc
- ▶ found under <http://www.gridisphere.org/>
- ▶ Runs under Gridsphere, but provides an abstraction layer for development

Resources Needed to Create a Gateway

- ▶ 0.3 to 2 FTE People
- ▶ Expertise in PHP, perl, SQL, Java, user interface
- ▶ 3 months to 2 man-years of time
- ▶ linux box/boxes
- ▶ Software
 - ▶ java, python, perl, ruby
 - ▶ DB program (MySQL etc)
 - ▶ Server - Apache, Apache Tomcat
 - ▶ Grid middle-ware - Globus, COG kit, etc
 - ▶ Portlet containers - Pluto, Gridsphere
 - ▶ Portlet building + toolkits - OGCE, GridPortlets
 - ▶ Webservices : WSRF

Servlets vs CGI

- ▶ Dynamically produced HTML like CGI
- ▶ Servlets do not run as a separate process from the server
- ▶ Servlets persist in memory between requests
- ▶ Part of Java API
- ▶ Require a web container - Tomcat
- ▶ Generally faster and “lighter” than CGI

Technology Summary

- ▶ There are MANY more layers of software on Grids
- ▶ This is more than just the web - user/software state is persistent
- ▶ Creating a grid is generally not useful for small user communities

The TeraGrid

<http://teragrid.org>

Grid infrastructure group at University of Chicago

Resources providers

Indiana University
Oak Ridge
NCSA
NCAR
Purdue
San Diego

Texas Advanced Computing Center
University of Chicago
Joint Institute for Computational
Sciences
Pittsburg Supercomputing Center

The TeraGrid

<http://teragrid.org>

Current science gateways include

- ▶ National Virtual Observatory (NVO)
- ▶ Special Priority and Urgent Computing Environment (SPRUCE)
- ▶ Linked Environments for Atmospheric Discovery (LEAD)
- ▶ Computing Chemistry Grid (GridChem)
- ▶ Computational Science And Engineering On-line (quantum chem)
- ▶ Network for Earthquake Engineering Simulation
- ▶ Biology and Biomedicine Science Gateway
- ▶ Neutron Science Teragrid Gateway

Special Priority and Urgent Computing Environment (SPRUCE)

<http://teragrid.org>

- ▶ provides fast, immediate access to resources to support large scale models
- ▶ gives massive resources on short notice to applications
- ▶ targeted for urgent decisions involving public health, safety, and security
- ▶ this is still a work in-progress

CSE-Online

A example Teragrid Portal

- ▶ GUI interfaces for quantum chemistry, kinetics, combustion chemistry, and bio simulations
- ▶ Knowledge management system for quantum chemistry
- ▶ Visualization/analysis tools

Creating a Cluster

At some point, you might need to configure a Beowulf cluster.
Buy the book “Beowulf Cluster Computing with Linux” by Gropp,
Lusk, and Sterling

Cluster Configuration

- ▶ install the main node
- ▶ configure cluster services on main node
- ▶ define configuration of a compute node
- ▶ for all the compute nodes
 - ▶ detect Ethernet address
 - ▶ install OS
 - ▶ complete configuration
- ▶ set up services (PBS,etc) that are now aware of the cluster

OSCAR

A management system for clusters

- ▶ Single image to install cluster management software
- ▶ contains
 - ▶ LAM and MPICH MPI distributions
 - ▶ A “switcher” to make it easy to change compute environments
 - ▶ Torque/Maui Scheduler
 - ▶ Password/user management system
 - ▶ SIS - a system installation suite to easily add nodes
 - ▶ C3 Cluster Command Control tools (cexec, cget, ckill, etc)

Queuing System

- ▶ qsub - submits a job
- ▶ qdel - deletes a job
- ▶ qstat - displays current job status
- ▶ pbsnodes - node status

Sample Script Using Torque

From OSCAR documentation

```
% qsub -N my_jobname -e my_stderr.txt \  
    -o my_stdout.txt -q workq \  
    -l nodes=X:ppn=Y:all,walltime=1:00:00 \  
    my_script.sh
```

my_script.sh

```
#!/bin/sh  
echo Launchnode is 'hostname'  
pbsdsh /path/to/my_executable  
# All done
```

Updating Software

- ▶ yume command - for yum updates over many clients

```
% yume update
```

```
% yume --installroot /var/lib/systemimager/oscarimage update
```

```
% cexec yume -y update
```

local update, local repository update, update other nodes

Switcher

- ▶ allows easy management of system packages by users

```
% switcher list
% switcher mpi --list
% switcher mpi = name
% switcher mpi = name --system
```

sets system level defaults

Basic Cluster Management

- ▶ queuing
- ▶ scheduling
- ▶ monitoring
- ▶ resource management
- ▶ accounting

Queuing

- ▶ users must be able to submit their jobs to a queue for processing
- ▶ work is collected into a batch system
- ▶ the batch system matches resources to jobs
- ▶ there is an art to creating the write kind of queues
 - ▶ maximum length of run
 - ▶ maximum size of the memory
 - ▶ maximum number of nodes

Scheduling

- ▶ choosing the “best” job to run
- ▶ best depends on
 - ▶ workload
 - ▶ usage policy
 - ▶ cluster resources
 - ▶ types of applications
- ▶ two basic parts
 - ▶ policy enforcement
 - ▶ resource optimization

Monitoring

- ▶ verify that idle nodes are ready to receive more work
- ▶ check utilization of in-use resources
- ▶ ensure processes are released and the node is working when a job completes

Resource Management

- ▶ starting, stopping, and cleaning up after jobs
- ▶ removing/adding additional computing resources

Accounting

- ▶ collecting resource usage data for jobs on the cluster
- ▶ allows a time allocation committee to budget cycles for projects
- ▶ helps justify the cost of the hardware to managers
- ▶ helps plan for future expansion

Summary

- ▶ policy drives the use of the cluster
- ▶ queuing, scheduling, and management are done based on these policies
- ▶ resource updates are essential as the system evolves
- ▶ accounting is essential for planning

Final thoughts

- ▶ Parallel computing is here to stay
 - ▶ graphics cards
 - ▶ cores
 - ▶ clusters
 - ▶ grids
- ▶ Use high level tools, if possible
 - ▶ OSCAR
 - ▶ OpenMP
 - ▶ CUDA
 - ▶ Libraries
 - ▶ MPI – if needed
- ▶ This is a rapidly changing field
 - ▶ keep up with technological changes!

