Submitting locally and running globally – The GLOW and OSG Experience

Miron Livny
Computer Sciences Department
University of Wisconsin-Madison
miron@cs.wisc.edu

The Condor Project (Established ’85)

Distributed Computing research performed by a team of ~35 faculty, full time staff and students who
• face software/middleware engineering challenges in a UNIX/Linux/Windows/OS X environment,
• involved in national and international collaborations,
• interact with users in academia and industry,
• maintain and support a distributed production environment (more than 3900 CPUs at UW),
• and educate and train students.

Funding – DOE, NIH, NSF, INTEL, IBM
Micron, Microsoft and the UW Graduate School

Main Threads of Activities

› Distributed Computing Research – develop and evaluate new concepts, frameworks and technologies
› Keep Condor “flight worthy” and support our users
› The Open Science Grid (OSG) – build and operate a national distributed computing and storage infrastructure
› The Grid Laboratory Of Wisconsin (GLOW) – build, maintain and operate a distributed computing and storage infrastructure on the UW campus
› The NSF Middleware Initiative (NMI) – develop, build and operate a national Build and Test facility

1986-2006
Celebrating 20 years since we first installed Condor in our department
Claims for “benefits” provided by Distributed Processing Systems

- High Availability and Reliability
- High System Performance
- Ease of Modular and Incremental Growth
- Automatic Load and Resource Sharing
- Good Response to Temporary Overloads
- Easy Expansion in Capacity and/or Function

“What is a Distributed Data Processing System?” , P.H. Enslow, Computer, January 1978

Democratization of Computing:
You do not need to be a super-person to do super-computing

High Throughput Computing

We first introduced the distinction between High Performance Computing (HPC) and High Throughput Computing (HTC) in a seminar at the NASA Goddard Flight Center in July of 1996 and a month later at the European Laboratory for Particle Physics (CERN). In June of 1997 HPCWire published an interview on High Throughput Computing.
Why HTC?

For many experimental scientists, scientific progress and quality of research are strongly linked to computing throughput. In other words, they are less concerned about instantaneous computing power. Instead, what matters to them is the amount of computing they can harness over a month or a year --- they measure computing power in units of scenarios per day, wind patterns per week, instructions sets per month, or crystal configurations per year.

High Throughput Computing is a 24-7-365 activity

\[
\text{FLOPY} \neq (60 \times 60 \times 24 \times 7 \times 52) \times \text{FLOPS}
\]

Obstacles to HTC

› Ownership Distribution (Sociology)
› Customer Awareness (Education)
› Size and Uncertainties (Robustness)
› Technology Evolution (Portability)
› Physical Distribution (Technology)

Leads to a “bottom up” approach to building and operating distributed systems
My jobs should run …

› … on my laptop if it is not connected to the network
› … on my group resources if my certificate expired
› … on my campus resources if the meta scheduler is down
› … on my national resources if the trans-Atlantic link was cut by a submarine

Condor will run it …

› … on the machine where it was submitted,
› … on machines in your pool,
› … on machines in other pools via flocking,
› … on machines in other pools via Condor-C,
› … on machines managed by “other” resource management systems (RMS) via remote submissions and/or
› … on machines managed by other RMS via glide-ins

The search for SUSY*

› Sanjay Padhi is a UW Chancellor Fellow who is working at the group of Prof. Sau Lan Wu located at CERN (Geneva)
› Using Condor Technologies he established a “grid access point” in his office at CERN
› Through this access-point he managed to harness in 3 month (12/05-2/06) more that 500 CPU years from the LHC Computing Grid (LCG) the Open Science Grid (OSG) the Grid Laboratory Of Wisconsin (GLOW) resources and local group owned desktop resources.
The grid promises to fundamentally change the way we think about and use computing. This infrastructure will connect multiple regional and national computational grids, creating a universal source of pervasive and dependable computing power that supports dramatically new classes of applications. The Grid provides a clear vision of what computational grids are, why we need them, who will use them, and how they will be programmed.

"... We claim that these mechanisms, although originally developed in the context of a cluster of workstations, are also applicable to computational grids. In addition to the required flexibility of services in these grids, a very important concern is that the system be robust enough to run in "production mode" continuously even in the face of component failures. ..."

Miron Livny & Rajesh Raman, "High Throughput Resource Management", in "The Grid: Blueprint for a New Computing Infrastructure".
“... Grid computing is a partnership between clients and servers. Grid clients have more responsibilities than traditional clients, and must be equipped with powerful mechanisms for dealing with and recovering from failures, whether they occur in the context of remote execution, work management, or data output. When clients are powerful, servers must accommodate them by using careful protocols...”


The OSG vision

Transform compute and data intensive science through a cross-domain self-managed national distributed cyber-infrastructure that brings together campus and community infrastructure and facilitating the needs of Virtual Organizations at all scales.

The Open Science Grid (OSG)

Miron Livny
OSG PI and Facility Coordinator
University of Wisconsin-Madison
Evolution of OSG


PPDGGriPhyNiVDGLTrilliumGrid3OSG (DOE) (DOE+NSF)(NSF)(NSF)

OSG Principles

- **Characteristics** -
  - Provide guaranteed and opportunistic access to shared resources.
  - Operate a heterogeneous environment both in services available at any site and for any VO, and multiple implementations behind common interfaces.
  - Interface to Campus and Regional Grids.
  - Federate with other national/international Grids.
  - Support multiple software releases at any one time.

- **Drivers** -
  - Delivery to the schedule, capacity and capability of LHC and LIGO:
    - Contributions to/from and collaboration with the US ATLAS, US CMS, LIGO software and computing programs.
    - Support for/collaboration with other physics/non-physics communities.
    - Partnerships with other Grids - especially EGEE and TeraGrid.
    - Evolution by deployment of externally developed new services and technologies.

Who are you?

- A resource can be accessed by a user via the campus, community or national grid.
- A user can access a resource with a campus, community or national grid identity.
Why do we need an OSG?

Sustained growth in the needs of traditional compute and data intensive science and the steady stream of scientific domains that add and expend the role of computing and data processing in their discovery process coupled with the administrative and physical distribution of compute and storage resources and increase in the size, diversity and scope of scientific collaborations.

OSG challenges

- Develop the organizational and management structure of a consortium that drives such a Cyber Infrastructure.
- Develop the organizational and management structure for the project that builds, operates and evolves such Cyber Infrastructure.
- Maintain and evolve a software stack capable of offering powerful and dependable capabilities that meet the science objectives of the NSF and DOE scientific communities.
- Operate and evolve a dependable and well managed distributed facility.

Part of the OSG Consortium

The OSG Project

- Co-funded by DOE and NSF at an annual rate of ~$6M for 5 years starting FY-07.
- 15 institutions involved – 4 DOE Labs and 11 universities.
- Currently main stakeholders are from physics - US LHC experiments, LIGO, STAR experiment, the Tevatron Run II and Astrophysics experiments.
- A mix of DOE-Lab and campus resources.
- Active “engagement” effort to add new domains and resource providers to the OSG consortium.
OSG PEP - Organization

OSG Project Execution Plan (PEP) - FTEs

<table>
<thead>
<tr>
<th>Activity</th>
<th>FTEs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Facility operations</td>
<td>6.0</td>
</tr>
<tr>
<td>Security and troubleshooting</td>
<td>4.5</td>
</tr>
<tr>
<td>Software release and support</td>
<td>6.5</td>
</tr>
<tr>
<td>Engagement</td>
<td>2.0</td>
</tr>
<tr>
<td>Education, outreach &amp; training</td>
<td>2.0</td>
</tr>
<tr>
<td>Facility management</td>
<td>1.0</td>
</tr>
<tr>
<td>Extensions in capability and scale.</td>
<td>8.0</td>
</tr>
<tr>
<td>Staff</td>
<td>3.0</td>
</tr>
<tr>
<td>Total FTEs</td>
<td>33</td>
</tr>
</tbody>
</table>

OSG Middleware Layering

- NSF Middleware Initiative (NMI): Condor, Globus, Myproxy
- Virtual Data Toolkit (VDT) Common Services NMI + VOMS, CEMon (common EGEE components), MonaLisa, Clarens, AuthZ
- OSG Release Cache: VDT + Configuration, Validation, VO management
- LIGO Data Grid
- CMS Services & Framework
- ATLAS Services & Framework
- CDF, D0 SamGrid & Framework

Resources and VOs activities

- Flexible Interfaces allow interoperation of previously disjoint resources.
- Sustaining through OSG submissions: 3,000-4,000 simultaneous jobs.
- ~10K jobs/day
- ~50K CPU hours/day.
- Peak test jobs of 15K a day.
- ~20,000 CPUs (from 30 to 4000)
- ~6 PB Tapes
- ~4 PB Shared Disk

- Using production & research networks
- 20 Virtual Organizations +6 operations Includes 25% non-physics.
FY 2006 Accomplishments

Open Science Grid

OSG impacts the Science of Distributed Computing
Supporting Computer Scientists, who study and develop distributed computing technologies, to use the OSG to experimentally evaluate novel frameworks and technologies.

OSG is stimulating Computational Science
Operating a large scale, heterogeneous distributed facility that supports innovative distributed algorithms. The computing and storage power of the facility enable at-scale evaluation of novel algorithms.

OSG is enhancing Scientific Discovery
Through a shared national facility increasing overall throughput and supporting “peak consumption” of computing and storage resources at critical times in the discovery process.

Inter-operability with Campus grids

At this point we have three operational campus grids – Fermi, Purdue and Wisconsin. We are working on adding Harvard (Crimson), Clemson, and Lehigh.

FermiGrid is an interesting example for the challenges we face when making the resources of a campus (in this case a DOE Laboratory) grid accessible to the OSG community.

What is FermiGrid?

- Integrates resources across most (soon all) owners at Fermilab.
- Supports jobs from Fermilab organizations to run on any/all accessible campus FermiGrid and national Open Science Grid resources.
- Supports jobs from OSG to be scheduled onto any/all Fermilab sites.
- Unified and reliable common interface and services for FermiGrid gateway - including security, job scheduling, user management, and storage.
- More information is available at http://fermigrid.fnal.gov

Job Forwarding and Resource Sharing

Gateway currently interfaces 5 Condor pools with diverse file systems and >1000 Job Slots. Plans to grow to 11 clusters (8 Condor, 2 PBS and 1 LSF)

Job scheduling policies and in place agreements for sharing allow fast response to changes in resource needs by Fermilab and OSG users.

Gateway provides single bridge between OSG wide area distributed infrastructure and FermiGrid local sites. Consists of a Globus gate-keeper and a Condor-G

Each cluster has its own Globus gate-keeper

Storage and Job execution policies applied through Site-wide managed security and authorization services.
The Crimson Grid is

- a Scalable collaborative computing environment for research at the interface of science and engineering
- a Gateway/Middleware release service to enable campus/community/national/global computing infrastructures for interdisciplinary research
- a Test bed for faculty & IT-industry affiliates within the framework of a production environment for integrating HPC solutions for higher education & research
- a Campus Resource for skills & knowledge sharing for advanced systems administration & management of switched architectures
UW Madison Campus Grid

- Condor pools in various departments (more than 3500 “cores”), made accessible via Condor ‘flocking’
  - Users submit jobs to their own private or department Condor scheduler.
  - Jobs are dynamically matched to available machines.
- Crosses multiple administrative domains.
  - No common uid-space across campus.
  - No cross-campus NFS for file access.
    - Users rely on Condor remote I/O, file-staging, AFS, SRM, gridftp, etc.

Grid Laboratory of Wisconsin

2003 Initiative funded by NSF(MIR)/UW at ~ $1.5M

Six Initial GLOW Sites

- Computational Genomics, Chemistry
- Amanda, Ice-cube, Physics/Space Science
- High Energy Physics/CMS, Physics
- Materials by Design, Chemical Engineering
- Radiation Therapy, Medical Physics
- Computer Science

Diverse users with different deadlines and usage patterns.

GLOW

- First machines arrived in 01/04
- First job completed 14 hours after machines arrived
- Today we have more than 1600 “cores”
- ~50% of these cores funded by non-GLOW money (mainly Capital Exercise and UW funds)
- One Group (UW-ATLAS) joined in 05

Submitting Jobs within UW Campus Grid

UW HEP User
Submitting Jobs within UW Campus Grid

UW HEP User

condor_submit

schedd (Job caretaker)

HEP matchmaker

CS matchmaker

GLOW matchmaker

flocking

4/5/07 46
Submitting Jobs within UW Campus Grid

UW HEP User

condor_submit

schedd
(Job caretaker)

GLOW
matchmaker

HEP matchmaker

CS matchmaker

startd
(Job Executor)

Example Uses

• Chemical Engineering
  – Students do not know where the computing cycles are coming from - they just do it - largest user group

• ATLAS
  – Over 15 Million proton collision events simulated at 10 minutes each

• CMS
  – Over 70 Million events simulated, reconstructed and analyzed (total ~10 minutes per event) in the past one year

• IceCube / Amanda
  – Data filtering used 12 CPU-years in one month

• Computational Genomics
  – Prof. Shwartz asserts that GLOW has opened up a new paradigm of work patterns in his group
  • They no longer think about how long a particular computational job will take - they just do it

Supports full feature-set of Condor:
• matchmaking
• remote system calls
• checkpointing
• MPI
• suspension VMs
• preemption policies
GLOW Usage 4/04-12/07

Over 23.4M CPU hours served!

Housing the Machines

- **Condominium Style**
  - centralized computing center
  - space, power, cooling, management
  - standardized packages

- **Neighborhood Association Style**
  - each group hosts its own machines
  - each contributes to administrative effort
  - base standards (e.g., Linux & Condor) to make easy sharing of resources

- **GLOW** has elements of both, but leans towards neighborhood style

Top active users by hours used on 01/16/2007:

dasu: 12305.4 (32.54%)
ice3simu: 11640.3 (30.92%)
szhou: 2641.8 (6.99%)
nenggu: 2168.1 (5.74%)
kirill: 2168.1 (5.74%)
sklya: 1535.6 (4.06%)
phuber: 1132.4 (2.99%)
tregre: 1043.8 (2.78%)
daing: 963.6 (2.55%)

Total hours: 37820.0

**CPU status last two years**

GLOW has elements of both, but leans towards neighborhood style.

- Neighborhood Association Style
- Condominium Style
- easy sharing of resources
- base standards (e.g. Linux & Condor) to make
- total hours: 37820.0
- Housing the Machines

- Condominium Style
  - centralized computing center
  - space, power, cooling, management
  - standardized packages

- Neighborhood Association Style
  - each group hosts its own machines
  - each contributes to administrative effort
  - base standards (e.g. Linux & Condor) to make easy sharing of resources

- GLOW has elements of both, but leans towards neighborhood style

Total hours: 37820.0
The Value of Campus Scale

- simplicity
  software stack is just Linux + Condor
- fluidity
  high common denominator makes sharing easier and provides richer feature-set
- collective buying power
  we speak to vendors with one voice
- standardized administration
  e.g. GLOW uses one centralized cfengine
- synergy
  face-to-face technical meetings
  mailing list scales well at campus level

Design highlights (HAD)

- Modified version of Bully algorithm
- One HAD leader + many backups
- HAD as a state machine
- “I am alive” messages from leader to backups
- Detection of leader failure
- Detection of multiple leaders (split-brain)
- “I am leader” messages from HAD to replication

GLOW Architecture in a Nutshell

One big Condor pool
- But backup central manager runs at each site (Condor HAD service)
- Users submit jobs as members of a group (e.g. “CMS” or “MedPhysics”)
- Computers at each site give highest priority to jobs from same group (via machine RANK)
- Jobs run preferentially at the “home” site, but may run anywhere when machines are available

HAD state diagram

http://www.cs.technion.ac.il/Labs/dsl/projects/gozal/
**HAD-enabled pool**

- Multiple Collectors run simultaneously on each Central Manager (CM) machine
- All submission and execution machines must be configured to report to all CMs
- High Availability
  - HAD runs on each CM
  - Replication daemon runs on each CM (if enabled)
- HAD makes sure a single Negotiator runs on one of the CMs
- Replication daemon makes sure the up-to-date accountant file is available

http://www.cs.technion.ac.il/Labs/dsl/projects/gozal/

**The value of the big G**

- Our users want to collaborate outside the bounds of the campus (e.g. Atlas and CMS are international).
- We also don’t want to be limited to sharing resources with people who have made identical technological choices.
- The Open Science Grid (OSG) gives us the opportunity to operate at both scales, which is ideal.

**Moving Forward**

- Submitted a proposal for the second phase of GLOW seeking $2M in NSF funding
- 3 new departments (Biostatistics and Medical Information, Genetics, Physical Engineering)
- 17 faculty involved
- Focused Education and Outreach activity
- Strong campus support
- Integration with national Cyberinfrastructure through OSG

**Submitting jobs through OSG to UW Campus Grid**

Open Science Grid User

4/5/07
Submitting jobs through OSG to UW Campus Grid

Open Science Grid User

condor_submit

schedd (Job caretaker)

condor gridmanager

Globus gatekeeper

condor_submit

schedd (Job caretaker)
Submitting jobs through OSG to UW Campus Grid

Open Science Grid User

condor_submit

schedd (Job caretaker)

condor gridmanager

condor_submit

 schedd (Job caretaker)

HEP matchmaker

CS matchmaker

GLOW matchmaker

flocking

schedd (Job caretaker)

condor_submit

condor gridmanager

Globus gatekeeper

HEP matchmaker

CS matchmaker

GLOW matchmaker

Elevating from GLOW to OSG

Schedd

On The Side

Schedd
Elevating from GLOW to OSG

The Grid Universe
The Grid Universe

The Grid Universe

The Grid Universe

The Grid Universe

Schedd

Random Seed

Schedd

vanilla
The Grid Universe

- Dynamic Routing Jobs

- • easier to live with private networks
  • may use non-Condor resources
  • restricted Condor feature set (e.g. no std universe over grid)
  • must pre-allocate jobs between vanilla and grid universe
Dynamic Routing Jobs

Local Starts

Negotiator

Schedd

Gatekeeper

Random Seed Random Seed

vanilla site X

Schedd On The Side

Random Seed Random Seed

vanilla site Y

Schedd On The Side

Random Seed Random Seed

vanilla site Z

Schedd On The Side
Dynamic Routing Jobs

- dynamic allocation of jobs between vanilla and grid universes.
- not every job is appropriate for transformation into a grid job.

What Types of Grids?

- Routing table may contain any combination of grid types supported by the grid universe.
- Example: Condor-C

What Types of Grids?

- Routing table may contain any combination of grid types supported by the grid universe.
- Example: Condor-C

- for two Condor sites, schedd-to-schedd submission requires no additional software
- however, still not as trivial to use as flocking
Routing Jobs from UW Campus Grid to OSG

condor_submit

schedd (Job caretaker)

HEP matchmaker
CS matchmaker
GLOW matchmaker

4/5/07 64
Routing Jobs from UW Campus Grid to OSG

- condor_submit
- Grid JobRouter
- HEP matchmaker
- CS matchmaker
- GLOW matchmaker
- schedd (Job caretaker)
- gridmanager
- globus gatekeeper

Combining both worlds:
- simple, feature-rich local mode
- when possible, transform to grid job for traveling globally

From a grid of one to a grid of many