From 10K cores on EC2 to 60K cores on OSG

On the fly Condor deployment

ANL April 14, 2011

Condor Project
Computer Sciences Department
University of Wisconsin-Madison
"The interesting thing about cloud computing is that we’ve redefined cloud computing to include everything that we already do. I can't think of anything that isn't cloud computing with all of these announcements. The computer industry is the only industry that is more fashion-driven than women's fashion."

Oracle CEO
Larry Ellison
Perspectives on Grid Computing

Uwe Schwiegelshohn Rosa M. Badia Marian Bubak Marco Danelutto
Schahram Dustdar Fabrizio Gagliardi Alfred Geiger Ladislav Hluchy
Dieter Kranzlmüller Erwin Laure Thierry Priol Alexander Reinefeld
Michael Resch Andreas Reuter Otto Rienhoff Thomas Rüter Peter Sloot
Domenico Talia Klaus Ullmann Ramin Yahyapour Gabriele von Voigt

We should not waste our time in redefining terms or key technologies: clusters, Grids, Clouds... What is in a name? Ian Foster recently quoted Miron Livny saying: "I was doing Cloud computing way before people called it Grid computing", referring to the ground breaking Condor technology. It is the Grid scientific paradigm that counts!
Distributed Computing is very powerful
Claims for “benefits” provided by Distributed Processing Systems

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

• 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
Some enabling properties associated w/ Clouds and our approach

- Clouds run **Virtual Machines**, grids run processes
- Clouds focus on **remote resource allocation** (provisioning), while grids typically focus on job delegation (to gatekeepers/CEs)
- On-the-fly dynamic deployment
Condor Job Types
(Universes)

› Typical jobs
  • Serial jobs, Serial jobs w/ migration, Parallel job, Java jobs, ...

› “Grid” jobs
  • Job is delegated to another scheduler
  • Grid types supported: Globus, CREAM, Unicore, NorduGrid/ARC, Condor, PBS, LSF, ...
  • “Condor-G”
Virtual Machines in Condor

- VMs can be treated as jobs
  - VM image is executable
  - Instance data is input
  - Modified image is output
  - Launch, Monitor, Run VM until it shuts down, Cleanup
  - Scheduling, data transfer, authentication, logging, etc. all work the same
VM Universe

- Run VMs in a Condor pool
- Works with VMware, Xen, KVM
- Would like to work with everything libvirt supports
Wipro’s Cloud Platform

- CondorWeek 2009 Wipro presented “Cloud computing platform based on Condor”
- Who is Wipro?
  - World’s largest independent R&D Services Provider
  - Amongst the world’s largest global IT services, BPO and Product Engineering companies
  - 108,071 employees, $6 billion annual revenues, 72 development centers worldwide
How are we using Condor?

• Mostly standard configuration
• A few custom class ads in jobs and machines
• Schedd and Collector configured in HA mode
• Condor spool for VM persistence
• **Virtual machine provision request handled by Condor**
  – VM job to physical machine match-making, file transfer
• Partitionable slots for dynamic partitioning of physical machine resources
• Customized condor_vm_* files for configuring and starting VM's
  – VLAN control, Swap disk and additional storage creation, ...
• Lease management for limiting the number of running instances of a licensed image
Why Condor?

- **Trusty old features**
  - Flexibility – ClassAd mechanism, configurations and policies
  - Web Services API
  - High availability
  - Resource utilization of jobs

- **Newer features we like**
  - **VM Universe**
  - Partitionable Slots
  - Lease management
  - Integration with Amazon EC2 (public cloud)

- Proven in large scale deployments
- Condor-users and condor-admin support
- Open source
Access to Cloud Services via Condor

- Cloud service handled as a job scheduler type in grid universe
- Condor can speak two cloud protocols
  - EC2
  - Deltacloud
- These cover a broad spectrum of cloud services
Cloud Protocols

» Amazon’s EC2 becoming a lingua franca in cloud world

» Many cloud services speak EC2
  • Nimbus
  • Eucalyptus
  • OpenStack
  • OpenNebula
EC2 Dialects

- EC2 has two dialects: Soap and Query
  - Condor speaks Soap today
- Many cloud services only speak Query
  - So Condor will speak Query soon... currently being tested, should be released in summer
Deltacloud

» Project sponsored by Red Hat and Apache
» Has its own simple protocol
» Translates requests into protocol of target cloud service
  • Many protocols supported
  • Wider reach than EC2 protocol (e.g. GoGrid, Rackable, RHEV, …)
Red Hat Expands Messaging, Realtime and Grid Technology Capabilities to Advance Cloud Leadership

October 14th, 2010 by Enterprise MRG Team

Red Hat today announced the availability of Red Hat Enterprise MRG 1.3, including updates to the product’s Messaging, Realtime and Grid technologies, which provide a key technology base for Red Hat Cloud Foundations, a solution set that offers a comprehensive set of tools to build and manage a private cloud. Red Hat Enterprise MRG provides an integrated platform for high-performance distributing computing. First released in June 2008, Enterprise MRG has since enabled customers around the world to meet their messaging, realtime and grid computing needs, offering: …
Enterprise MRG’s Grid functionality, based on the Condor Project created and hosted by the University of Wisconsin, Madison, brings the advantages of flexible deployment to a wide range of applications and workloads.

With Grid, customers can build cloud infrastructures to aggregate multiple clouds. It provides integrated support for virtualization and public clouds and easier aggregation of multiple cloud resources into one compute pool. In addition, it provides more streamlined and flexible computing across remote grids with servers, clusters and cycle-harvesting from desktop PCs as well as across private, public and hybrid clouds. MRG Grid is a key base component of Red Hat Cloud Foundations.
VM and Cloud provisioning with Deltacloud

Job request

MRG Grid

Deltacloud driver

Private Cloud with VMs

External Cloud

Execute Node(s) on Physical HW

Execute Node Running VMs

VM Execute Node

VM Execute Node

Execute Node

Execute Node
Case Study: Scale Testing

- Goal is to hit \( \geq 95\% \) 24x365 (i.e. AY)
- That meant 28,000 jobs running concurrently
- The 7 day test – real data of the worst 7 days
- Ramp up time – 30 minutes
- Dynamic deployment and reconfiguration
  - Machines, Data and Policies

Under NDA
Open Science Grid
HTC at the National Level
Overlay Resource Managers

Ten years ago we introduced the concept of Condor glide-ins as a tool to support ‘just in time scheduling’ in a distributed computing infrastructure that consists of resources that are managed by (heterogeneous) autonomous resource managers. By dynamically deploying a distributed resource manager on resources allocated (provisioned) by the local resource managers, the overlay resource manager can implement a unified resource allocation policy.

In other words, we use remote job invocation to get resources allocated.
Submit local, run global!

GlideinWMS

- Frontend node
- Monitor Condor
- Central manager
- Submit node
- Factory node
- Condor
- Factory
- CREAM
- Startd
- Glidein
- Submit glideins
- Execute node glidein
- Execute node glidein
- Execute node glidein

Submit glideins
ExTENCI Project

> ExTENCI - Automate and streamline...
  > Authoring of application-specific VM images
  > Distribution of images to many cloud services
  > Launching and management of VM instances as part of a glidein pool

> Also working w/ ROCKS project
  > GLEON computing
Current OSG glideinWMS factory stats (174 logical sites) : 10k cores
# of Cores delivered by glideinWMS and managed by one Condor submit front end: 60k
10,000-Core EC2 Pool

- Done by Cycle Computing (www.cyclecomputing.com) for Genentech
- Approximates #114 in Top 500 Supercomputers list
- Ran life science workflow for 8 hours
- Total cost $1060/hr
- 2k cores in 15min, 10k cores in 45min
# 10,000-Core EC2 Pool

<table>
<thead>
<tr>
<th>Cluster Role</th>
<th>Count</th>
<th>AWS Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execution Machines</td>
<td>1250</td>
<td>c1.xlarge: 8 core, 7GB RAM each</td>
</tr>
<tr>
<td>Condor Collector/Negotiator and CycleServer</td>
<td>1</td>
<td>m1.xlarge: 4 core, 17.1GB RAM</td>
</tr>
<tr>
<td>Primary Scheduler and Disk Filer</td>
<td>1</td>
<td>c1.xlarge: 8 core, 7GB RAM</td>
</tr>
<tr>
<td>Auxiliary Schedulers</td>
<td>2</td>
<td>m1.large: 2 core, 7.5GB RAM</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>1254</strong></td>
<td>10014 core, 8.6TB RAM, 2PB disk</td>
</tr>
</tbody>
</table>
We are working so Condor can Manage the resources of my Cloud and/or Provision resources from a Cloud and/or Schedule my resources in a Cloud … along w/ the software engineering challenge of doing so currently at production scales of tens of thousands of cores.
Thank you for the opportunity to present!

Questions?