Extending Rocks Clusters into Amazon EC2 Using Condor

Philip Papadopoulos, Ph.D
University of California, San Diego
San Diego Supercomputer Center
California Institute for Telecommunications and Information Technology (Calit2)
Background:
So, You want to build a cluster?

Large Memory PSDAF
• 256 GB & 512 GB Nodes (32 core)
• 8TB Total
• 128 GB/sec
• ~ 9TF

Shared Resource Cluster
• 16 GB/Node
• 4 - 8TB Total
• 256 GB/sec
• ~ 20 TF

Large Scale Storage (Working on RFP)
• 2 – 4 PB
• 50-125 GB/sec
• 3000 – 6000 disks

UCSD Research Labs

Campus Research Network

© UC Regents 2010
The Modern “Cluster” Architecture is Not Just an MPI Cluster

Each logical Configuration is a Rocks **Appliance**

**CAMERA** Bioinformatics

Standard Compute Cluster
Rocks  www.rocksclusters.org

- Technology transfer of commodity clustering to application scientists
  - “make clusters easy”
- Rocks is a cluster on a CD
  - Clustering software (PBS, SGE, Ganglia, Condor, ... )
  - Highly programmatic software configuration management
  - Put CDs in Raw Hardware, Drink Coffee, Have Cluster.
- Extensible using “Rolls”
- Large user community
  - Over 1PFlop of known clusters
  - Active user / support list of 2000+ users
  - Estimate > 2000 installed cluster
- Active Development
  - 2 software releases per year
  - Code Development at SDSC
  - Other Developers (UCSD, Univ of Tromso, External Rolls)
- Supports Redhat Linux, Scientific Linux, Centos and Solaris
- Can build Real, Virtual, and Hybrid Combinations
Rocks Breaks Apart the Software Stack into **Rolls**

- **Parallel Code / WebFarm / Grid / Computer Lab**
- **Message Passing / Communication Layer**
  - **Job Scheduling and Launching**
  - **Cluster Software Management**
  - **Cluster State Management / Monitoring**
- **Linux Environment**
  - **Linux Kernel**
- **HPC Device Drivers**
  (e.g., Interconnect and Storage)

© UC Regents 2010
Rolls on a Simple Cluster

```
Connection to ec2-75-101-204-74.compute-1.amazonaws.com closed.
[root@landphil ~]# rocks list roll
NAME    VERSION  ARCH    ENABLED
sge:    5.2      x86_64  yes
ganglia: 5.2     x86_64  yes
kernel:  5.2     x86_64  yes
base:   5.2      x86_64  yes
java:   5.2      x86_64  yes
service-pack: 5.2.2 x86_64  yes
bio:    5.2      x86_64  no
area51: 5.2      x86_64  yes
xen:    5.2      x86_64  yes
hpc:    5.2      x86_64  yes
web-server: 5.2    x86_64  yes
CentOS:  5.3     x86_64  yes
CentOS-Updates: 5.3-2009-09-02 x86_64  yes
green:  5.2      x86_64  yes
condor: 5.2      x86_64  yes
ec2:    5.2      x86_64  yes
apbs:   5.3      x86_64  no
[root@landphil ~]#
```
Condor Roll

• Condor 7.4.1 (updating to 7.4.2)
• Integration with Rocks command line to do basic Condor configuration customization
• To build a Condor Cluster with Rocks
  – Base, OS, Kernel, Condor Roll
  – Gives you local collector, scheduler
• Basic, Working Configuration that can be customized as required.
Virtual Clusters in Rocks Today

Require:
1. Virtual Frontend
2. Nodes w/disk
3. Private Network
4. Power

Virtual Clusters:
• May overlap one another on physical HW
• Need network isolation
• May be larger or smaller than physical hosting cluster
How Rocks Treats Virtual Hardware

• It’s just another piece of HW.
  – If RedHat supports it, so does Rocks
• Allows mixture of real and virtual hardware in the same cluster
  – Because Rocks supports heterogeneous HW clusters
• Re-use of all of the software configuration mechanics
  – E.g., a compute appliance is compute appliance

Virtual HW must meet minimum HW Specs
  – 1GB memory
  – 36GB Disk space*
  – Private-network Ethernet
  – + Public Network on Frontend

* Not strict – EC2 images are 10GB
Rocks Hybrid: Linux/Solaris/
Physical/Virtual

```
[root@vstorage ~]# uname -a
Linux vstorage.rocksclusters.org 2.6.18-128.1.6.el5xen #1 SMP Wed Apr 1 09:53:14
EDT 2009 x86_64 x86_64 x86_64 x86_64 GNU/Linux
[root@vstorage ~]# rocks list host
HOST MEMBERSHIP CPUS RACK RANK RUNACTION INSTALLACTION
vstorage: Frontend 1 0 0 0 os install
v20nas-sdsc-0-0: NAS Appliance 1 0 0 os install_sol
v20nas-sdsc-0-1: NAS Appliance 1 0 1 os install_sol
[root@vstorage ~]# ssh v20nas-sdsc-0-0
Last login: Fri Oct 2 07:51:02 2009 from vstorage.local
Sun Microsystems Inc. SunOS 5.10 Generic January 2005
Rocks 5.2 (Chimichanga)
Profile built 15:24 27-May-2009

Jumpstarted 15:30 27-May-2009
# uname -a
SunOS v20nas-sdsc-0-0.local 5.10 Generic_137138-09 i86pc i386 i86pc
# zfs list | grep datapool
datapool1 39.1M 8.89T 43.2K /datapool1
datapool1/arajendr 43.2K 8.89T 43.2K /datapool1/arajendr
datapool1/gbruno 47.3K 8.89T 47.3K /datapool1/gbruno
datapool1/mjkatz 47.3K 8.89T 47.3K /datapool1/mjkatz
datapool1/ppapadop 38.7M 8.89T 38.7M /datapool1/ppapadop
datapool1/pragma 29.9K 8.89T 29.9K /datapool1/pragma
```
Basic EC2

- AMIs are **copied** from S3 and booted in EC2 to create a “running instance”
- When instance is shutdown, all changes are lost
  - Can save as a new AMI

S3 – Simple Storage Service
EBS – Elastic Block Store

Amazon Machine Images (AMIs)
Basic EC2

• AMI (Amazon Machine Image) is copied from S3 to EC2 for booting
  – Can boot multiple copies of an AMI as a “group”
  – Not a cluster, all running instances are independent

• If you make changes to your AMI while running and want them saved
  – Must repack to make a new AMI
    • Or use Elastic Block Store (EBS) on a per-instance basis
Some Challenges in EC2

1. Defining the contents of your Virtual Machine (Software Stack)
2. Understanding limitations and execution model
3. Debugging when something goes wrong
4. Remembering to turn off your VM
   – Smallest 64-bit VM is ~$250/month running 7x24
What’s in the AMI?

- Tar file of a / file system
  - Cryptographically signed so that Amazon can open it, but other users cannot
  - Split into 10MB chunks, stored in S3
- Amazon boasts more than 2000 public machine images
  - What’s in a particular image?
  - How much work is it to get your software part of an existing image?
- There are tools for booting and monitoring instances.
- Defining the software contents is “an exercise left to the reader”
The EC2 Roll

• Take a Rocks appliance and make it compatible with EC2:
  – 10GB disk partition (single)
  – DHCP for network
  – ssh key management
  – Other small adjustments
• Create an AMI bundle on local cluster
  – `rocks create ec2 bundle`
• Upload a bundled image into EC2
  – `rocks upload ec2 bundle`
• Mini-tutorial on getting started with EC2 and Rocks
Putting all together: Virtual Cluster Experiment

Nimrod – Monash University
Rocks® – UC San Diego
Condor – U. Wisconsin
Amazon EC2 – Brought to you by Visa®
Virtual Clusters in Rocks Today

Physical Hosting Cluster
“Cloud Provider”

Require:
1. Virtual Frontend
2. Nodes w/disk
3. Private Network
4. Power

Virtual Clusters:
• May overlap one another on physical HW
• Need network isolation
• May be larger or smaller than physical hosting cluster
Extended Cluster Experiment in PRAGMA

NIMROD – Parameter Sweep/Optimization

MeSSAGE Lab
Monash eScience and Grid Engineering Laboratory

Nimrod.rockscluster.org
fiji.rocksclusters.org
Hosting Cluster

Amazon EC2 Cloud
Rocks – Created VM

Condor
High Throughput Computing

for Job Management
Extended Cluster Using Condor

```
[ppapadop@landphil ~]$ rm *err *log *out
[ppapadop@landphil ~]$ condor_submit hello.sub
Submitting job(s)........
Logging submit event(s).........
8 job(s) submitted to cluster 11.
[ppapadop@landphil ~]$ condor_status
```

```
<table>
<thead>
<tr>
<th>Name</th>
<th>OpSys</th>
<th>Arch</th>
<th>State</th>
<th>Activity</th>
<th>LoadAv</th>
<th>Mem</th>
<th>ActvtyTime</th>
</tr>
</thead>
<tbody>
<tr>
<td>compute-0-0-0.locan</td>
<td>LINUX</td>
<td>X86_64</td>
<td>Unclaimed</td>
<td>Idle</td>
<td>0.000</td>
<td>1024</td>
<td>+01:30:04</td>
</tr>
<tr>
<td>devel-server-0-1-0</td>
<td>LINUX</td>
<td>X86_64</td>
<td>Unclaimed</td>
<td>Idle</td>
<td>0.000</td>
<td>1024</td>
<td>00:00:04</td>
</tr>
<tr>
<td>slot1@ec2-75-101-2</td>
<td>LINUX</td>
<td>X86_64</td>
<td>Claimed</td>
<td>Busy</td>
<td>0.000</td>
<td>2560</td>
<td>00:00:02</td>
</tr>
<tr>
<td>slot1@landphil-0-0</td>
<td>LINUX</td>
<td>X86_64</td>
<td>Unclaimed</td>
<td>Idle</td>
<td>0.000</td>
<td>658</td>
<td>00:00:04</td>
</tr>
<tr>
<td>slot2@ec2-75-101-2</td>
<td>LINUX</td>
<td>X86_64</td>
<td>Claimed</td>
<td>Busy</td>
<td>0.000</td>
<td>2560</td>
<td>00:00:03</td>
</tr>
<tr>
<td>slot2@landphil-0-0</td>
<td>LINUX</td>
<td>X86_64</td>
<td>Unclaimed</td>
<td>Idle</td>
<td>0.000</td>
<td>658</td>
<td>00:00:05</td>
</tr>
<tr>
<td>slot3@ec2-75-101-2</td>
<td>LINUX</td>
<td>X86_64</td>
<td>Claimed</td>
<td>Busy</td>
<td>0.000</td>
<td>2560</td>
<td>00:00:03</td>
</tr>
<tr>
<td>slot3@landphil-0-0</td>
<td>LINUX</td>
<td>X86_64</td>
<td>Claimed</td>
<td>Busy</td>
<td>0.000</td>
<td>658</td>
<td>00:00:06</td>
</tr>
<tr>
<td>slot4@landphil-0-0</td>
<td>LINUX</td>
<td>X86_64</td>
<td>Claimed</td>
<td>Busy</td>
<td>0.000</td>
<td>658</td>
<td>00:00:07</td>
</tr>
</tbody>
</table>

Total

```

© UC Regents 2010
Can Log into the Running VM

```
[root@landphil ~]# ssh ec2-75-101-204-74.compute-1.amazonaws.com
Last login: Thu Apr 15 07:45:08 2010 from rocks-154.sdsc.edu
Rocks 5.2 (Chimichanga)
Profile built 11:23 03-Mar-2010

Kickstarted 11:43 03-Mar-2010
EC2-enabled Client
Rocks 5.2 Development Server
[root@ec2-75-101-204-74 ~]# date
Thu Apr 15 07:46:10 PDT 2010
[root@ec2-75-101-204-74 ~]# condor_status
```

```
<table>
<thead>
<tr>
<th>Name</th>
<th>OpSys</th>
<th>Arch</th>
<th>State</th>
<th>Activity</th>
<th>LoadAv</th>
<th>Mem</th>
<th>ActvtyTime</th>
</tr>
</thead>
<tbody>
<tr>
<td>compute-0-0-0.loca</td>
<td>LINUX</td>
<td>X86_64</td>
<td>Unclaimed Idle</td>
<td>0.000</td>
<td>1024</td>
<td>0+01:40:04</td>
<td></td>
</tr>
<tr>
<td>devel-server-0-1-0</td>
<td>LINUX</td>
<td>X86_64</td>
<td>Unclaimed Idle</td>
<td>0.000</td>
<td>1024</td>
<td>0+00:11:08</td>
<td></td>
</tr>
<tr>
<td>slot1@ec2-75-101-2</td>
<td>LINUX</td>
<td>X86_64</td>
<td>Unclaimed Idle</td>
<td>0.000</td>
<td>2560</td>
<td>0+00:09:28</td>
<td></td>
</tr>
<tr>
<td>slot1@landphil-0-0</td>
<td>LINUX</td>
<td>X86_64</td>
<td>Unclaimed Idle</td>
<td>0.000</td>
<td>658</td>
<td>0+00:11:06</td>
<td></td>
</tr>
<tr>
<td>slot2@ec2-75-101-2</td>
<td>LINUX</td>
<td>X86_64</td>
<td>Unclaimed Idle</td>
<td>0.000</td>
<td>2560</td>
<td>0+00:09:29</td>
<td></td>
</tr>
<tr>
<td>slot2@landphil-0-0</td>
<td>LINUX</td>
<td>X86_64</td>
<td>Unclaimed Idle</td>
<td>0.000</td>
<td>658</td>
<td>0+00:11:07</td>
<td></td>
</tr>
<tr>
<td>slot3@ec2-75-101-2</td>
<td>LINUX</td>
<td>X86_64</td>
<td>Unclaimed Idle</td>
<td>0.000</td>
<td>2560</td>
<td>0+00:09:30</td>
<td></td>
</tr>
<tr>
<td>slot3@landphil-0-0</td>
<td>LINUX</td>
<td>X86_64</td>
<td>Unclaimed Idle</td>
<td>0.000</td>
<td>658</td>
<td>0+00:11:08</td>
<td></td>
</tr>
<tr>
<td>slot4@landphil-0-0</td>
<td>LINUX</td>
<td>X86_64</td>
<td>Unclaimed Idle</td>
<td>0.000</td>
<td>658</td>
<td>0+00:11:09</td>
<td></td>
</tr>
</tbody>
</table>
```

© UC Regents 2010
Steps to Make this Work

**PREPARATION**

- Build Local Cluster with appropriate rolls
  - Rocks + Xen Roll + EC2 Roll + Condor Roll (+ NIMROD + ... )
- Create local appliance as VM using standard Rocks tools
  - Set ec2_enable attribute to build it as an EC2-Compatible VM
  - Build and test locally
- Bundle, Upload, Register as an EC2 AMI
  - Rocks command line tools

**RUN**

- Boot with appropriate meta data to register automatically with your local collector.
  - ec2-run-instances -t m1.large ami-219d7248 -d "condor:landphil.rocksclusters.org:40000:40050"
  - Requires one-time EC2 firewall settings
- Use your extended Condor Pool
Summary

• Easily Extend your Condor pool into EC2
  – Others can do this as well
  – Condor supports the public/private network duality of EC2

• Have your software on both local cluster and remote VM in EC2

• Mix and match
  – Local Physical, Local Virtual, Remote Virtual

• If you use Rocks, does not take extra effort