Experience Running an Analysis Cluster in an Academic Cloud

Peter Onyisi, Crystal Riley

DPF, 15 August 2013
Cloud Computing

What is a Cloud?

• Let's go with “computing services with uniform interface made available to a broad community’’
  – internal components abstracted

• Different types often referred to as “X as a service”
  – Storage as a Service: Dropbox, Amazon S3
  – Software as a Service: Gmail, Microsoft Office 365
  – Infrastructure as a Service (virtual machines on abstract hardware): Amazon EC2, Openstack, Eucalyptus, Nimbus, ...
  – Platform as a Service: LHC Computing Grid
Why are (IaaS) Clouds Interesting?

- User can create any base image (OS, loaded software, network configurations...) they want
  - replicated across multiple virtual machines
  - trivial rollback to known good version
  - easy deployment of changes

- Less concern about hardware management and lifecycles
  - (of course assuming others are running your cloud for you)

- Potential sharing of resources with others: run VMs only when needed
  - no contention over software configuration!
Our Use Case

- UT-Austin has no centralized HEP computing; sysadmin resources very stretched
  - workstations somewhat heterogeneous (even with Puppet, etc.)
- Investment in hardware will be obsolete (and fall out of warranty) within a few years
- Choice of being too small to handle peak loads, or so large that resources usually idle

Can we have CPU on demand with completely controlled and homogeneous software and configuration?

**Yes**, with Infrastructure as a Service.
Clouds at UT-Austin

- “Enterprise” cloud (VMWare); not intended for dynamic loads
  - also, $$$$  

- Research cloud (FutureGrid) is part of NSF XSEDE
  - testbed for high performance cloud research
  - UT site (Alamo) is administered by the Texas Advanced Computing Center (TACC)
  - can compare the performance of VMs and bare metal on identical physical nodes
  - also, free
Other Work

- CERN's interactive linux cluster recently switched to all virtual machines!
- Large research effort within ATLAS, other experiments to use clouds to top up capacity
  - access (and pay for) resources only when needed
  - in ATLAS, effort focused on commercial clouds and CERN infrastructure, some work with FutureGrid
  - More focused on VMs as part of ATLAS production system rather than generic batch system nodes
Overall Architecture

Firewalls are a complication...

- Workstation NAS
- User @ Workstation
- TACC Alamo login node
- Compute instance
- Compute instance
- Storage instance (not yet)
- NFS
- Service node (Condor, HTTP proxy, xrootd for local files) *Hosted in UT enterprise cloud*

- ATLAS Grid data (via FAX)

- Job submission via Bosco

- cluster startup over ssh

- firewalls

**Condor**
Alamo IaaS Stacks

- Alamo offers **OpenStack** and **Nimbus** stacks
- Both use KVM to actually run the VM, images incompatible with other FutureGrid sites (which use Xen)
- Most of our effort is on OpenStack: tools more transparent than Nimbus

<table>
<thead>
<tr>
<th>Feature</th>
<th>OpenStack</th>
<th>Nimbus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon EC2 API</td>
<td>Yes</td>
<td>“Yes” (optional)</td>
</tr>
<tr>
<td>Block persistent storage</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Object persistent storage</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Contextualization</td>
<td>Amazon-like user-data</td>
<td>Nimbus-specific broker</td>
</tr>
<tr>
<td>Default public IPs</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
VM Images

- CentOS 5.9 images made with Boxgrinder: http://boxgrinder.org/
  - in fact, built with Boxgrinder virtual appliance (build VM image in VM); build is < 10 min
- Key ingredient: CVMFS to provide access to ATLAS software stack
  - images are configured to use our local (always on) HTTP cache, so we're not re-downloading everything from CERN all the time
- ATLAS DB access through Frontier, again through local cache
- Other software as required
Starting the Batch System

- We use Condor as the batch system
- We constantly run a Condor scheduler outside the cloud
- When VMs boot, they start local Condor daemons and register job slots with the main scheduler
  - dynamic handling is automatic; clean shutdown of VMs means slots are properly removed from scheduler
- We find our OpenStack instances boot much faster than Nimbus ones (< 1 min vs minutes)
  - probably because the Nimbus installation does not support the QCOW2 image format, so we use gzipped RAW images
Submitting a Job

• We use Bosco: http://bosco.opensciencegrid.org/
  – Bosco creates a local Condor cluster that will submit jobs to other batch systems on your behalf (Condor, SGE, PBS, LSF...)
  – We submit a Condor job to the Bosco queue (along with necessary inputs); job is sent to remote worker node
  – Outputs are copied back via Condor (heavy reliance on Condor's file transfer mechanisms)
  – No significant latency from Bosco seen
Data

- So far we are discussing a “diskless” Tier-3
  - clouds generally do not have huge block storage available (Alamo limits us to 1TB)
  - they can have large object storage (like Amazon S3) but typically this doesn’t match well to direct use in ROOT

- Access to experiment data planned through wide area network xrootd
  - to work with Snowmass Energy Frontier Delphes ROOT files we used xrootd to Nebraska-Lincoln
  - Plan to use ATLAS Federated xrootd for ATLAS data

- Local storage (shared with user workstations): access through xrootd bridge
Tuning

- Virtual machines add extra tuning complications: *host*, *hypervisor*, and *guest* all need to be tuned

- e.g. networking:
  - host needs good performance
  - hypervisor/host kernel need to have paravirtualization enabled
  - guest needs to use drivers for paravirtual device

The following network comparison is based on the current Alamo configuration. Compute nodes have 1 Gbps links, I/O speeds tested to other UT machines:

<table>
<thead>
<tr>
<th></th>
<th>Bare metal</th>
<th>Nimbus</th>
<th>OpenStack</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paravirtualized?</td>
<td>N/A</td>
<td>No (emulated NIC)</td>
<td>Yes (virtio)</td>
</tr>
<tr>
<td>Traffic Shaping?</td>
<td>No</td>
<td>No</td>
<td>Yes?</td>
</tr>
<tr>
<td>Peak network I/O per node</td>
<td>113 MB/s</td>
<td>~ 40 MB/s</td>
<td>11.2 MB/s (!)</td>
</tr>
</tbody>
</table>
Exercising the System

- We used the cloud cluster for analysis of fast simulation for Snowmass
  - used OpenStack configuration as it was ready
- Biggest limitation was networking on our side
  - working with cloud sysadmins to understand this
- Some instability in the VMs seen (random reboots)
  - proved impossible to reproduce
- Condor and Bosco worked well
Future Directions

- Explore most efficient configuration for cluster
  - can tune number of cores/memory/local disk per instance
- Explore best interface to data
  - will federated xrootd be sufficient?
- Automated start/stop of cluster
  - Cloud Scheduler?
- Explore possibility of long term production system as a resource for UT