The Evolution of Cloud Computing in ATLAS

Ryan Taylor

on behalf of the ATLAS collaboration
Outline

- Cloud Usage and IaaS Resource Management
- Software Services to facilitate cloud use
- Sim@P1
- Performance Studies
- Operational Integration
  - Monitoring, Accounting
The Clouds of ATLAS

ATLAS cloud jobs (Jan. 2014 – present)

61% Single-core production
33% Multi-core production
3% User analysis

CPU consumption All jobs in seconds
11210 Hours from Week 00 of 2014 to Week 15 of 2015 UTC

Total: 37,221,986,771, Average: 522.28 s
IaaS Resource Management

- HTCondor+Cloud Scheduler, VAC/VCycle, APF
- See talk 131 “HEP cloud production using the CloudScheduler/HTCondor Architecture” (C210, Tue. PM)
- Dynamic Condor slots to handle arbitrary job requirements
  - e.g. single-core, multi-core, high-mem
- uCernVM image
- Contextualization using cloud-init
- Using Glint Image Management System
  - see poster 304
Shoal
Proxy Cache “Federator”

• Build a fabric of proxy caches
  – configurationless topology
  – robust
  – scalable

• Needed to run uCernVM at scale
  – By default, DIRECT connection to closest Stratum 0/1
  – Contextualize instances to find proxy using Shoal
    
    [ucernvm-begin]
    CVMFS_PAC_URLS=http://shoal.heprc.uvic.ca/wpad.dat
    CVMFS_HTTP_PROXY=auto
    [ucernvm-end]

• Also use Shoal for Frontier access
  – Currently under investigation
Sim@P1

- Resource contribution similar to T1
  - 34M CPU hours, 1.1B MC events
- Used for LHC stops > 24h
- Fast automated switching via web GUI for shifters
  - TDAQ to Sim@P1: 20m (check Nova DB, start VMs)
  - Sim@P1 to TDAQ: 12m (graceful VM shutdown, update DB)
  - Emergency switch to TDAQ: 100s (immediate termination)

- See poster 169
HS06 Benchmarking Study

- Commercial clouds provide on-demand scalability
  - e.g. urgent need for beyond pledged resources
- But how cost-effective are they?
- Comparison to institutional clouds
Evolution of Cloud Computing in ATLAS

ATLAS Preliminary

Cloud Benchmarking

VM Type

GCE; Standard
- n1-standard-1
- n1-standard-2
- n1-standard-4
- n1-standard-8
- n1-standard-16

GCE; High CPU
- n1-highcpu-2
- n1-highcpu-4
- n1-highcpu-8
- n1-highcpu-16

Amazon EC2
- m3.large
- m3.xlarge
- m3.2xlarge

- cc-west
  - c2.1low
  - c4.1low
  - c8.1low
  - c16.1low

- cc-east
  - c4.1low
  - c8.1low
  - c16.1low

HS06 Score
Evolution of Cloud Computing in ATLAS

ATLAS Preliminary

Cloud Benchmarking

VM Type

GCE; Standard
- nl-standard-1
- nl-standard-2
- nl-standard-4
- nl-standard-8
- nl-standard-16

GCE; High CPU
- nl-highcpu-2
- nl-highcpu-4
- nl-highcpu-8
- nl-highcpu-16

Amazon EC2
- m3.large
- m3.xlarge
- m3.2xlarge

cc-west
- c2.1low
- c4.1low
- c8.1low
- c16.1low

cc-east
- c4.1low
- c8.1low
- c16.1low

HS06 Score / Number of vCPUs
T2 & Remote Cloud Performance Comparison

- Used Hammercloud stress tests (24 hour stream)
- Data and squid cache at grid site
  - Remote access for cloud site
    - like zero-storage processing site

Success rate similar

HC 20052434
MC12 AtlasG4_trf 17.2.2.2
• Software setup time
  – Relies on CVMFS cache and Squid proxy
  – VMs have to fill up empty cache

• Data stage-in time
  – Local vs. remote storage access

\[ (15 \pm 7) \text{ s} \]
\[ (45 \pm 15) \text{ s} \]
\[ (11 \pm 4) \text{ s} \]
\[ (54 \pm 20) \text{ s} \]
• Total running time
  - 1.5x longer on cloud
  - different CPUs
  - hyperthreading?
  - data & software access time not significant

• CPU efficiency equal!

• Cloud usage is efficient for this workload

• No significant performance penalty
Cloud Monitoring

- VM management becomes the responsibility of the VO
- Basic monitoring is required
  - Detect and restart problematic VMs
  - Identify “dark” resources (deployed but unusable)
  - Can identify inconsistencies in other systems through cross-checks
- Common framework for all VOs
- Implemented with Ganglia
- \url{http://agm.cern.ch}
Cloud Accounting

- Provider-side: commercial invoice for resources delivered
- Consumer-side: record resources consumed
- Need to cross-check invoice against recorded usage!

http://cloud-acc-dev.cern.ch/monitoring/ATLAS
Conclusion

- Increasing use of clouds in ATLAS Distributed Computing
- Performance characterization of commercial clouds
- More integration into operational model
  - accounting, monitoring, support
- Developing and deploying services to facilitate cloud use
Extra Material
Dynamic Federation

UGR

- Lightweight, scalable, stateless
- General-purpose, standard protocols and components
  - Could be adopted by multiple experiments
  - e.g. DataBridge, LHCb demo: http://federation.desy.de/fed/lhcb/
- Metadata plugin used to emulate Rucio directory structure
- No site action needed to join
  - HTTP endpoints extracted from AGIS with script
RACF/BNL Amazon Project

Enabled by $200k grant from Amazon to run all ATLAS workloads at large scale
Encompasses provisioning/compute, storage, networking, and ATLAS workflow.
  VMs via Imagefactory and templates/profiles.
  VM runtime config by cloud-init->Hiera->masterless Puppet.
Provisioning via AutoPyFactory, HTCondor-G. HTCondor batch pool.
  3 EC2 regions and 12 instance types to maximize capacity. Spot market.
SRM/GridFTP EC2 instance w/ S3FS back end. One per region.
Ultimately S3 native storage endpoint. Job stage-in/out via S3.
10/100Gb peering and 10Gb DirectConnect to 3 regions via ESNet.
Data egress fees waived as long as <15% of total cost.
Event service nearing completion w/ S3 objectstore, active deletion, and EC2 merge jobs.
  S3 storage support in Rucio/DDM.
2.5k node/20k core tested so far, 100k core final goal
List of Active Squids

5 active in the last 180 seconds

<table>
<thead>
<tr>
<th>#</th>
<th>Hostname</th>
<th>Public IP</th>
<th>Private IP</th>
<th>Bytes Out</th>
<th>City</th>
<th>Region</th>
<th>Country</th>
<th>Latitude</th>
<th>Longitude</th>
<th>Last Received</th>
<th>Alive</th>
<th>Verified</th>
<th>Access Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>squid-test01.gridpp.rl.ac.uk</td>
<td>130.246.183.249</td>
<td></td>
<td>0 kB/s</td>
<td>Appleton</td>
<td>United Kingdom</td>
<td></td>
<td>51.7</td>
<td>-1.35</td>
<td>7s</td>
<td>42h40m43s</td>
<td>✓</td>
<td>Global</td>
</tr>
<tr>
<td>2</td>
<td>kraken01-westgrid.ca</td>
<td>206.12.46.240</td>
<td>172.22.22.25</td>
<td>809 kB/s</td>
<td>Vancouver</td>
<td>Canada</td>
<td></td>
<td>40.2838</td>
<td>-123.1041</td>
<td>10s</td>
<td>107h49m9s</td>
<td>✓</td>
<td>Global</td>
</tr>
<tr>
<td>3</td>
<td>atlascaq3.triumf.ca</td>
<td>142.90.110.88</td>
<td></td>
<td>0 kB/s</td>
<td>Vancouver</td>
<td>Canada</td>
<td></td>
<td>49.2755</td>
<td>-123.2177</td>
<td>20s</td>
<td>166h20m3s</td>
<td>✓</td>
<td>Global</td>
</tr>
<tr>
<td>4</td>
<td>atlas-squid.cern.ch</td>
<td>128.142.220.105</td>
<td></td>
<td>0 kB/s</td>
<td>Geneva</td>
<td>Switzerland</td>
<td></td>
<td>48.1958</td>
<td>6.1481</td>
<td>22s</td>
<td>166h19m59s</td>
<td>X</td>
<td>Global</td>
</tr>
<tr>
<td>5</td>
<td>t2software3.physics.ox.ac.uk</td>
<td>163.1.5.176</td>
<td></td>
<td>35 kB/s</td>
<td>Oxford</td>
<td>United Kingdom</td>
<td></td>
<td>51.7</td>
<td>-1.25</td>
<td>26s</td>
<td>166h18m56s</td>
<td>✓</td>
<td>Global</td>
</tr>
</tbody>
</table>

PAC Interface

```javascript
function FindProxyForURL(url, host) {
}
```

JSON REST Interface

```json
{
    "load": 0,
    "domain_access": true,
    "equid_port": 3128,
    "global_access": true,
    "verified": true,
    "last_active": 1249903448.149289,
    "created": 1246836678.111659,
    "external_ip": null,
    "geo_data": {
        "city": "Vancouver",
        "region_name": "BC",
        "area_code": 0,
        "timezone": "America/Vancouver",
        "dma_code": 0,
        "metro_code": null,
        "country": "CA",
        "country_name": "Canada",
        "continent": "NA"
    },
    "hostname": "atlascaq3.triumf.ca",
    "public_ip": "142.90.110.88",
    "private_ip": null,
    "net_load": 1289903448.149289,
    "distance": 6.002394311931146886
}
```

• [github.com/hep-gc/shoal](http://github.com/hep-gc/shoal)
• [CHEP 2013 Poster](http://shoal.heprc.uvic.ca)
http://cern.ch/go/d8Qj
http://cern.ch/go/HB9m