ARC-CE v6 at ScotGrid Glasgow
Outline

- New cluster prototype
- Installing ARC 6
- Authentication with ARGUS
- Adding HTCondor
- Testing
- Documentation & Code

**ScotGrid Glasgow**: Gareth, Gordon, Sam, me (Emanuele)

**ARC-CE**: The Advanced Resource Connector (ARC) Computing Element (CE) middleware, developed by the NorduGrid Collaboration, is an open source software solution enabling e-Science computing infrastructures with emphasis on processing of large data volumes.
What does it look like in reality?

The core of our new cluster built in two test-racks
(soon to be dismantled and moved to the DC)

Physical machines (compute & services):
- Server DELL 440 (3x), Proxmox (VMs)
- Server DELL 640 (1x), Proxmox
- HP ProLiant DL60 Gen9 (7x), WorkNode
- Server DELL 410 (1x), Squid
- Extreme switch 1 Gb (1x), IPMI net
- Lenovo switch 10 Gb (2x), internal net
- Lenovo switch 40 Gb (1x), uplink

Virtual machines (services):
- DNS (3x), DHCP (1x), NAT (2x) - ARGUS (1x)
- HTCondor Manager (1x) - ARC-CE (2x)

Ceph testing (storage):
- Server DELL 740 (8x) CEPH storage,
- Server DELL 440 (3x) CEPH servers
- Server DELL 440+ (2x) CEPH cache
With the new cluster the desire was to move key components to a virtualised environment to improve resiliency. Investigated Openstack, oVirt and others but they were complicated to operate with our desired level of resiliency.

Settled on ProxMox:

- Debian based appliance.
- Built from 3x R440 with SSDs for VM storage.
- Configured using CEPH for distributed fault tolerance.
- Can set High Availability for automatic VM migration on fault.
- Allows VMs and Containers to be centrally managed.
- Currently running key services for the new production environment.
Why ARC 6 and not HTCondor-CE?

• We investigated the option of deploying an HTCondor-CE, instead of an ARC-CE and started some initial experiments using details from Steve Jones (see GridPP & HepSysMan talks) along with the OSG documentation …

• We couldn’t find an exact match to replicate our production set-up (we wanted multiple CEs with a separate HTCondor Collector running on a different host) in the limited time we had we found the configuration of security between the CE and the multiple Collectors proved tricky …

So, to meet the CentOS7 upgrade timetable… **we have decided to go for ARC-CEs**, as we were more familiar with the configuration and had confidence in our understanding of the technology.

Plus, NorduGrid released a new ARC-CE v6 just at the right time …
We first installed the newly released ARC6 following the instructions on NorduGrid website:

http://www.nordugrid.org/arc/arc6/admins/arc6_install_guide.html

Pre-requisites: The ARC server needs a valid Host Certificate and an external IP

The installation steps are as follows:

- install the NorduGrid **GPG key** and add the NorduGrid **repository**:
  
rpm --import http://download.nordugrid.org/RPM-GPG-KEY-nordugrid-6

  wget https://download.nordugrid.org/packages/nordugrid-release/releases/6/centos/el7/x86_64/nordugrid-release-6-1.el7.noarch.rpm

  yum install nordugrid-release-6-1.el7.noarch.rpm

- install the ARC-6 package **nordugrid-arc-arex** with yum:
  
yum -y install nordugrid-arc-arex

- finally configure and start the ARC service(s):
  
  vi /etc/arc.conf

  arcctl service enable --as-configured --now

The default configuration enables to run fork jobs on the ARC server. And with **arcctl test-ca** we can set up a dummy C.A. and test job submission!
Proper Configuration

Configuring the full batch system needs few more configuration bits (/etc/arc.conf):

<table>
<thead>
<tr>
<th>Section</th>
<th>Value</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>[lrms]</td>
<td>lrms = condor</td>
<td>Defines the Local Resources Manager System as HTCondor</td>
</tr>
<tr>
<td>[authgroup: all-vos]</td>
<td>voms = * * * *</td>
<td>What VO's are allowed to do</td>
</tr>
<tr>
<td>[gridftp/jobs]</td>
<td>allowaccess = all-vos</td>
<td>Who is allowed to submit jobs via gridftp</td>
</tr>
<tr>
<td>[mapping]</td>
<td>map with plugin = all-vos 30 /usr/libexec/arc/arc-lcmaps %D %P liblcmaps.so /usr/lib64 /etc/lcmaps/lcmaps.db arc</td>
<td>External plug-in for mapping users 🌱</td>
</tr>
<tr>
<td>[infosys/cluster]</td>
<td>advertisedvo = vo.scotgrid.ac.uk</td>
<td>VO's that are allowed on the cluster</td>
</tr>
</tbody>
</table>

About authentication, we need to configure the lcmaps plug-in (/etc/lcmaps/lcmaps.db).

See next slide …
ARC-CE integrating with ARGUS

ARC-CE can do the authentication chain either with a static gridmap or using external plug-ins for user mapping (defined in the arc.conf). In particular, the lcmaps plug-in is used:

[mapping]
map_with_plugin = all-vos 30 /usr/libexec/arc/arc-lcmaps %D %P liblcmaps.so /usr/lib64/etc/lcmaps/lcmaps.db arc

The lcmaps plug-in points to the ARGUS server: (/etc/lcmaps/lcmaps.db)

Trust-Chain:
ARC CE needs IGTF CA certificates deployed to authenticate users and other services, such as storage elements.
To deploy IGTF CA certificates to ARC CE host, run:

  arcctl deploy igtf-ca classic

...then we can do job submission from real certificates!

path = /usr/lib64/lcmaps
verify_proxy = "lcmaps_verify_proxy.mod"
    "-certdir /etc/grid-security/certificates"
    "--discard_private_key_absence"
    "--allow-limited-proxy"

pepc = "lcmaps_c_pep.mod"
    "--pep-daemon-endpoint-url https://argus:8154/authz"
    "--resourceid http://gla.scotgrid.ac.uk/ce01"
    "--actionid http://glite.org/xacml/action/execute"
    "--capath /etc/grid-security/certificates/
    "--certificate /etc/grid-security/hostcert.pem"
    "--key /etc/grid-security/hostkey.pem"

# Policies:
arc:
verify_proxy -> pepc
UMD4 & Yum Priority

About installing lcmaps from the UMD4 repository …

When UMD4 is added from the rpm, by default it sets itself with priority=1 (the highest) and therefore it will take over any other repository … even if UMD4 packages are older versions. This is the case for ARC6 (new in NorduGrid) against ARC5.4 (old in UMD4).

The priority of NorduGrid repo must be adjusted before installing ARC6 ! (e.g., set it equal to the priority of UMD4, so the newest package is selected)

```
cd /etc/yum.repos.d
vi nordugrid*.repo
set: priority=1
yum clean all
```

Check what repository will be used before install:

```
yum info nordugrid-arc-arex
```

→ hopefully: nordugrid-

Alternatively, the UMD4 repo can be temporarily disabled or removed ...
If you have a HTCondor batch system you just need to add this line in `/etc/arc.conf` (... and the HTCondor Scheduler must be installed on the ARC-CE server)

```
[lrms]
lrms = condor
```

Our prototype batch system is composed of 7 physical machines as WorkNodes, and 2 virtual machines: the HTCondor Manager and the Scheduler (which is also the ARC-CE).

<table>
<thead>
<tr>
<th>Node type</th>
<th>Daemons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Worker Node</td>
<td>MASTER, STARTD</td>
</tr>
<tr>
<td>Manager Node</td>
<td>MASTER, SCHEDD, COLLECTOR, NEGOTIATOR</td>
</tr>
<tr>
<td>CE Node</td>
<td>MASTER, SCHEDD</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Service</th>
<th>IPv4 (int)</th>
<th>IPv4 (ext)</th>
<th>Hardware</th>
</tr>
</thead>
<tbody>
<tr>
<td>wn001</td>
<td>WorkNode</td>
<td>10.1.60.1</td>
<td>/ /</td>
<td>HP ProLiant</td>
</tr>
<tr>
<td>wn002</td>
<td>WorkNode</td>
<td>10.0.60.2</td>
<td>/ /</td>
<td>HP ProLiant</td>
</tr>
<tr>
<td>wn003</td>
<td>WorkNode</td>
<td>10.0.60.3</td>
<td>/ /</td>
<td>HP ProLiant</td>
</tr>
<tr>
<td>wn004</td>
<td>WorkNode</td>
<td>10.1.60.4</td>
<td>/ /</td>
<td>HP ProLiant</td>
</tr>
<tr>
<td>wn005</td>
<td>WorkNode</td>
<td>10.0.60.5</td>
<td>/ /</td>
<td>HP ProLiant</td>
</tr>
<tr>
<td>wn006</td>
<td>WorkNode</td>
<td>10.0.60.6</td>
<td>/ /</td>
<td>HP ProLiant</td>
</tr>
<tr>
<td>wn007</td>
<td>WorkNode</td>
<td>10.0.60.7</td>
<td>/ /</td>
<td>HP ProLiant</td>
</tr>
<tr>
<td>Cassowary</td>
<td>Manager</td>
<td>10.1.43.1</td>
<td>/ /</td>
<td>ProxMox VM</td>
</tr>
<tr>
<td>Ce01</td>
<td>ARC-CE</td>
<td>10.1.44.1</td>
<td>130.209.239.25</td>
<td>ProxMox VM</td>
</tr>
<tr>
<td>svr029</td>
<td>ARGUS</td>
<td>10.1.45.1</td>
<td>130.209.239.29</td>
<td>ProxMox VM</td>
</tr>
<tr>
<td>Biscotti</td>
<td>Squid</td>
<td>10.1.40.31</td>
<td>...</td>
<td>DELL R410</td>
</tr>
</tbody>
</table>

**Note:**

The HTCondor batch system doesn’t ‘know’ about the ARC-CE (no reference in condor config). So ... configure the HTCondor cluster first, then install ARC-CE on the Scheduler node!
ARC-CE integration test

The full chain (ARC-CE + ARGUS + HTCondor) has been extensively tested by submitting jobs with our ScotGrid certificate from a remote ARC-Client. Example:

- generate an arc-proxy:
  
  `arcproxy -S vo.scotgrid.ac.uk`

- check info about the ARC-CE and then submit the job:
  
  `arcinfo -c ce01.gla.scotgrid.ac.uk`
  
  `arcsub -c ce01.gla.scotgrid.ac.uk sieve.rsl`

If the ‘submit’ is successful, it returns a string with the Job-id (e.g., `gsiftp://ce01.gla.scotgrid.ac.uk:2811/jobs/blabla`). Use this string to check the status of the job (`arcstat`) and retrieve the output when job is finished (`arcget`):

  `arcstat gsiftp://ce01.gla.scotgrid.ac.uk:2811/jobs/blabla`
  
  `arcget gsiftp://ce01.gla.scotgrid.ac.uk:2811/jobs/blabla`

The command `arcget` clears the job and copies the output to a local folder, named after the job-id:

  `ls blabla/`
  
  `cat blabla/stdout`

For this test job (Eratostene's sieve), the output should be a list of prime numbers from 1 to 1000.

For testing the Squid integration, it is sufficient to run a job which does `ls` on CVMFS folders.
Turning on accounting in ARC6 is simple, only takes 2 steps:

1) Register to GOC,
2) Add this bit to `/etc/arc.conf`:

```bash
[arex/jura]
loglevel=DEBUG

[arex/jura/archiving]

[arex/jura/apel:egi]
targeturl=https://mq.cro-ngi.hr:6162
topic=/queue/global.accounting.cpu.central
gocdb_name=UKI-SCOTGRID-GLASGOW
benchmark_type=HEPSPEC
benchmark_value=8.74
use_ssl=yes
```
During our tests, we have found few **bugs** in ARC6:

We wanted to run ops-test. To do this we had to enable the BDII & register in GOC …

**Problem:** the update of BDII failed if the provided information is not complete. But it failed silently: no error message was given! (in particular: job submission failed if the user specify a minimum memory requirement but the site did not publish one!)

We think these are the minimal settings it can work with …

We also discover a race condition:

**systemd** starts the update and then terminates before the update is finished, causing the update to fail!

This was reported by Gareth, and NorduGrid fixed it in ARC6.1!
Add a VO

New VOs are added to the cluster via an Ansible role, which takes care of the following:

# Create pool accounts and home directories on ARC-CE and worker nodes
  - include_tasks: pool-accounts.yml
# Create vomsdir configuration everywhere: create VOMSDIR & sub-folders, generates LSC files
  - include: vomsdir.yml
# Create vomses configuration on worker nodes: VOMSES folder and content
  - include: vomses.yml
# Create mapfiles on ARGUS
  - include: mapfiles.yml
# Perform additional configuration on ARGUS: touch files for user mappings & add VO entry to Argus policy
  - include: argus-config.yml
# Perform additional configuration on ARC-CE: add advertisedvo entry to arc.conf
  - include: arc-ce-config.yml

So far, we have added (and made Ansible 'roles' for) the following VOs:

- vo-atlas
- vo-dteam
- vo-gluex
- vo-ops
- vo-scotgrid

Beta testing:
the batch system is currently running pilot jobs from ATLAS

Results:
jobs can are running and everything seems to work fine!
ARC-CE has a new `arcctl`

New in ARC6 is `arcctl`, which gives an easy way to configure services, access accounting and get info ….

(see some examples of `arcctl` commands)

```
[root@ce01 ~]# arcctl service list
arc-acix-index     (Not installed, Disabled, Stopped)
arc-acix-scanner   (Not installed, Disabled, Stopped)
arc-arex           (Installed, Enabled, Running)
arc-datadelivery-service (Not installed, Disabled, Stopped)
arc-gridftp        (Installed, Enabled, Running)
arc-infosys-ldap   (Installed, Enabled, Running)
[root@ce01 ~]#
```

```
[root@ce01 ~]# arcctl rte list
ENV/CANDYPOND      (system, disabled)
ENV/CONDOR/DOCKER  (system, disabled)
ENV/LRMS-SCRATCH   (system, disabled)
ENV/PROXY          (system, enabled)
ENV/PROXY-backup   (system, disabled)
ENV/RTE            (system, disabled)
ENV/SINGULARITY    (system, disabled)
APPS/HEP/ATLAS-SITE-LCG (dummy, enabled)
ENV/GLITE          (dummy, enabled)
[root@ce01 ~]#
```

```
[root@ce01 ~]# arcctl accounting stats -t apel
 Number of jobs:     62197
 Total WallTime: 2909 days, 22:31:54
 Total CPUBTime: 2242 days, 17:00:58
[root@ce01 ~]#
```
Code is Growing

We have collected all configuration/installation procedures in a clean Ansible environment, safely stored in our GitLab repository. As the code has been growing and becoming more consistent, we thought is about time to share it.

→ see Gordon’s talk!

What we have built is a collection of fully automated procedures to build a Tier2 site from bare metal 😊

But, whatever automated tools are out there, we all know that site admins will still prefer to go the hard way ...
Documentation is Growing too

Cluster information and administrative procedures are safely organized in our internal Wiki, which is linked to the relevant Ansible roles in GitLab.

As both documentation and code grow, things start getting hard to manage and pages keep getting out of synch ... hopefully things will converge.

However, from my un-expert standpoint, I have learned to make documentation part of the procedures: Learn → Do → Write (... and repeat the cycle a few times, as we normally do).

These pages are currently available only to us (Glasgow), but we are ready to share our documentation as we are sharing our Ansible roles.

http://dokuwiki.beowulf.cluster/
## Physical Machines

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Name</th>
<th>Model</th>
<th>IPv4 (int)</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>40Gb Switch</td>
<td>Lenovo RackSwitch G8332</td>
<td>10.0.2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10Gb Switch</td>
<td>Lenovo RackSwitch G8272</td>
<td>10.0.2.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dell Server</td>
<td>Croquembouche</td>
<td>DELL R640</td>
<td>10.1.10.1</td>
<td>provision</td>
</tr>
<tr>
<td>Dell Server</td>
<td>Snicker</td>
<td>DELL R440</td>
<td>10.1.20.1</td>
<td>ProxMox</td>
</tr>
<tr>
<td>Dell Server</td>
<td>Doodle</td>
<td>DELL R440</td>
<td>10.1.20.2</td>
<td></td>
</tr>
<tr>
<td>Dell Server</td>
<td>Krumkake</td>
<td>DELL R440</td>
<td>10.1.20.3</td>
<td></td>
</tr>
</tbody>
</table>

### 1Gb Switch

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Name</th>
<th>Model</th>
<th>IPv4 (int)</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>1Gb Switch</td>
<td>Extreme Summit x440-48t-10G</td>
<td>10.14.2.101.241</td>
<td>mngm</td>
<td></td>
</tr>
<tr>
<td>1Gb Switch</td>
<td>Lenovo RackSwitch G8272</td>
<td>10.0.2.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1Gb Switch</td>
<td>Lenovo RackSwitch G8272</td>
<td>10.0.2.5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### HP Server

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Name</th>
<th>Model</th>
<th>IPv4 (int)</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>HP Server</td>
<td>Wn001</td>
<td>HPE ProLiant DL60 Gen9</td>
<td>10.1.60.1</td>
<td>WorkNode</td>
</tr>
<tr>
<td>HP Server</td>
<td>Wn002</td>
<td>HPE ProLiant DL60 Gen9</td>
<td>10.1.60.2</td>
<td>WorkNode</td>
</tr>
<tr>
<td>HP Server</td>
<td>Wn003</td>
<td>HPE ProLiant DL60 Gen9</td>
<td>10.1.60.3</td>
<td>WorkNode</td>
</tr>
<tr>
<td>HP Server</td>
<td>Wn004</td>
<td>HPE ProLiant DL60 Gen9</td>
<td>10.1.60.4</td>
<td>WorkNode</td>
</tr>
<tr>
<td>HP Server</td>
<td>Wn005</td>
<td>HPE ProLiant DL60 Gen9</td>
<td>10.1.60.5</td>
<td>WorkNode</td>
</tr>
<tr>
<td>HP Server</td>
<td>Wn006</td>
<td>HPE ProLiant DL60 Gen9</td>
<td>10.1.60.6</td>
<td>WorkNode</td>
</tr>
<tr>
<td>HP Server</td>
<td>Wn007</td>
<td>HPE ProLiant DL60 Gen9</td>
<td>10.1.60.7</td>
<td>WorkNode</td>
</tr>
</tbody>
</table>

### Dell Server

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Name</th>
<th>Model</th>
<th>IPv4 (int)</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dell Server</td>
<td>CephS01</td>
<td>DELL R440</td>
<td>10.1.50.1</td>
<td>CEPH server</td>
</tr>
<tr>
<td>Dell Server</td>
<td>CephS02</td>
<td>DELL R440</td>
<td>10.1.50.2</td>
<td>CEPH server</td>
</tr>
<tr>
<td>Dell Server</td>
<td>CephS03</td>
<td>DELL R440</td>
<td>10.1.50.3</td>
<td>CEPH server</td>
</tr>
<tr>
<td>Dell Server</td>
<td>CephC01</td>
<td>DELL R440</td>
<td>10.1.50.11</td>
<td>CEPH cache</td>
</tr>
<tr>
<td>Dell Server</td>
<td>CephC02</td>
<td>DELL R440</td>
<td>10.1.50.12</td>
<td>CEPH cache</td>
</tr>
<tr>
<td>Dell Server</td>
<td>Biscotti</td>
<td>DELL R410</td>
<td>10.1.40.31</td>
<td>Squid Cache</td>
</tr>
<tr>
<td>Dell Server</td>
<td>Canucci</td>
<td>DELL R510</td>
<td>10.1.40.33</td>
<td>YAC's Squid</td>
</tr>
</tbody>
</table>

### Lenovo Server

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Name</th>
<th>Model</th>
<th>IPv4 (int)</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lenovo 40 Gb</td>
<td>Lenovo RackSwitch G8332</td>
<td>10.0.2.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lenovo 40 Gb</td>
<td>Lenovo RackSwitch G8272</td>
<td>10.0.2.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lenovo 10 Gb</td>
<td>Lenovo RackSwitch G8272</td>
<td>10.1.50.2</td>
<td>CEPH server</td>
<td></td>
</tr>
<tr>
<td>Lenovo 10 Gb</td>
<td>Lenovo RackSwitch G8272</td>
<td>10.1.50.3</td>
<td>CEPH server</td>
<td></td>
</tr>
<tr>
<td>Extreme 1 Gb</td>
<td>Extreme Summit x440-48t-10G</td>
<td>10.14.2.101.241</td>
<td>mngm</td>
<td></td>
</tr>
</tbody>
</table>

### Other Rack

- Lenovo 10 Gb
- Lenovo 40 Gb
- Extreme 1 Gb

### Test Rack

- Lenovo 10 Gb
- Lenovo 40 Gb

### Diagram:

- Lenovo 40 Gb: 10.0.2.1, 10.0.2.2, 10.1.50.1, 10.1.50.2, 10.1.50.3
- Lenovo 10 Gb: 10.0.2.3, 10.1.50.2, 10.1.50.3
- Dell Server: CephS01, CephS02, CephS03, CephC01, CephC02
- HP Server: Wn001, Wn002, Wn003, Wn004, Wn005, Wn006, Wn007
- Dell Server: Biscotti, Canucci
- Lenovo Server: Lenovo RackSwitch G8332, Lenovo RackSwitch G8272, Lenovo RackSwitch G8272
These services are running as VMs on ProxMox (see ProxMox VM Cluster).

### Network Services (10.1.40.4)

<table>
<thead>
<tr>
<th>Name</th>
<th>Service</th>
<th>IPv4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kataifi</td>
<td>DHCP</td>
<td>10.1.40.1</td>
</tr>
<tr>
<td>Balkava</td>
<td>DNS</td>
<td>10.1.40.11</td>
</tr>
<tr>
<td>Tulumba</td>
<td>DNS</td>
<td>10.1.40.12</td>
</tr>
<tr>
<td>Chalvax</td>
<td>DNS</td>
<td>10.1.40.13</td>
</tr>
<tr>
<td>Stroop</td>
<td>NAT</td>
<td>10.1.40.21</td>
</tr>
<tr>
<td>Wafel</td>
<td>NAT</td>
<td>10.1.40.22</td>
</tr>
</tbody>
</table>

### Miscellaneous Services (10.1.41.4)

<table>
<thead>
<tr>
<th>Name</th>
<th>Service</th>
<th>IPv4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dokuwiki</td>
<td>Wiki</td>
<td>10.1.41.1</td>
</tr>
<tr>
<td>Cookbook</td>
<td>Gitlab</td>
<td>10.1.41.2</td>
</tr>
</tbody>
</table>

### Monitoring Services (10.1.42.4)

<table>
<thead>
<tr>
<th>Name</th>
<th>Service</th>
<th>IPv4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gingersnap</td>
<td>Prometheus</td>
<td>10.1.42.1</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>

### HTCondor batch system (10.1.43-45.4)

<table>
<thead>
<tr>
<th>Name</th>
<th>Service</th>
<th>IPv4 (int)</th>
<th>IPv4 (ext)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cassowary</td>
<td>Manager</td>
<td>10.1.43.1</td>
<td></td>
</tr>
<tr>
<td>Ce01</td>
<td>ARC-CE</td>
<td>10.1.44.1</td>
<td>130.209.239.25</td>
</tr>
<tr>
<td>Ce002</td>
<td>ARC-CE</td>
<td>10.1.44.2</td>
<td>130.209.239.122</td>
</tr>
<tr>
<td>Sw029</td>
<td>ARGUS</td>
<td>10.1.45.1</td>
<td>130.209.239.29</td>
</tr>
</tbody>
</table>
HTCondor Cluster

The prototype batch system is composed of 7 physical machines as CentOS7 WorkNodes & 2 virtual machines: the HTCondor Manager and the Scheduler (which is also CE).

Installation steps:

• Install HTCondor with yum from University of Wisconsin–Madison repo
  

• Set the important configuration files (/etc/condor/config.d):
  
  01-daemons.config → Daemons settings (see table)
  network.config → internal IP address
  security.config → local settings ...

<table>
<thead>
<tr>
<th>Name</th>
<th>Service</th>
<th>IPv4 (int)</th>
<th>IPv4 (ext)</th>
<th>Hardware</th>
</tr>
</thead>
<tbody>
<tr>
<td>wn001</td>
<td>WorkNode</td>
<td>10.1.60.1</td>
<td>/</td>
<td>HP ProLiant</td>
</tr>
<tr>
<td>wn002</td>
<td>WorkNode</td>
<td>10.0.60.2</td>
<td>/</td>
<td>HP ProLiant</td>
</tr>
<tr>
<td>wn003</td>
<td>WorkNode</td>
<td>10.0.60.3</td>
<td>/</td>
<td>HP ProLiant</td>
</tr>
<tr>
<td>wn004</td>
<td>WorkNode</td>
<td>10.1.60.4</td>
<td>/</td>
<td>HP ProLiant</td>
</tr>
<tr>
<td>wn005</td>
<td>WorkNode</td>
<td>10.0.60.5</td>
<td>/</td>
<td>HP ProLiant</td>
</tr>
<tr>
<td>wn006</td>
<td>WorkNode</td>
<td>10.0.60.5</td>
<td>/</td>
<td>HP ProLiant</td>
</tr>
<tr>
<td>wn007</td>
<td>WorkNode</td>
<td>10.0.60.7</td>
<td>/</td>
<td>HP ProLiant</td>
</tr>
<tr>
<td>Cassowary</td>
<td>Manager</td>
<td>10.1.43.1</td>
<td>/</td>
<td>ProxMox VM</td>
</tr>
<tr>
<td>Ce01</td>
<td>ARC-CE</td>
<td>10.1.44.1</td>
<td>130.209.239.25</td>
<td>ProxMox VM</td>
</tr>
<tr>
<td>svr029</td>
<td>ARGUS</td>
<td>10.1.45.1</td>
<td>130.209.239.29</td>
<td>ProxMox VM</td>
</tr>
<tr>
<td>Biscotti</td>
<td>Squid</td>
<td>10.1.40.31</td>
<td>...</td>
<td>DELL R41.0</td>
</tr>
</tbody>
</table>

We have Ansible roles for the complete set-up of WorkNodes and Services (takes care of repositories, packages installation and configuration).

Note: The HTCondor batch system does no need to know about the ARC-CE (no reference in the config). So ... install HTCondor first, then the ARC-CE on the Scheduler node.
ARGUS

The ARGUS server is installed on a CentOS7 Virtual Machine (svr029):

Installation of the ARGUS meta-package (via Ansible):

- Repositories are added (Epel, EGI, UMD4, Globus, ARGUS),
- Default TCP ports are enabled for ARGUS (8150, 8152, 8154) and LDAP (2170),
- Globus and gLite packages are installed (with default config),
- The BDII resource package is installed and configured (custom LDAP),
- The ARGUS package is installed and its components/services are configured,
- ARGUS services are started in the correct sequence: PAP, PDP, PEPD.

Ingredients to make it work (what must be present on the ARGUS server):

- Map-files must be present on ARGUS: grid-mapfile, groupmapfile, voms-grid-mapfile;
- Empty ‘pool-mapping’ files on ARGUS (/etc/grid-security/gridmapdir/*) must match the pool accounts on the HTCondor WorkNodes;
- VOMSDIR directory on ARGUS must contain LSC (=List of Certificates) files for each supported VO (/etc/grid-security/vomsdir/*//*.lsc);
- Policy files on ARGUS must be updated with supported VOs: /etc/policies/svr029.policy;
- Up-to-date host certificate and external IP must exist.

Note: the link between ARC-CE and ARGUS is made on the ARC server via LCMAPS. So ... no special steps are needed when installing ARGUS.
Squid

As the WorkNodes use CVMFS, we built a Squid server to act as cache for the cluster: the new Squid is installed on a physical DELL 410 machine placed in the same test-racks.

Hardware specs as follows:

- **Hardware:** DELL R410
- **CPU:** Xeon 4core @ 2.40 GHz
- **RAM:** 16 Gb
- **HD(s):** 3 * 250 Gb
- **Name:** biscotti
- **IPv4 (int):** 10.x.x.xx
- **IPv4 (ext):** not yet

**Note:** the Ansible role we use to installs CVMFS also sets the URL of this Squid on all worknodes

```yaml
infrastructure.yml
squid_server_host: biscotti.beowulf.cluster
squid_server: "{"squid_server_host }":3128"
```

The Squid server was also configured remotely with Ansible, steps are as follows:

- The root logical volume / is extended proportionally to the number of WORKERS and CACHE_MEM,
- The CERN Frontier yum repository is added,
- Frontier Squid is installed with yum,
- The main configuration file (/etc/squid/customize.sh) is generated with the given parameters,
- The Squid service is started.

Network ports: Frontier squid communicates on ports 3128 (TCP) and 3401 (UDP).
ARC-CE (notes)

Slides on initial install

1) Simple submission with built in CA
2) Add prod CA with fixed user mappings
3) Add central auth with new ARGUS
4) Add batch farm
5) Add required RTE
6) Full integration testing

New tooling, should highlight arcctl

Slides on encountered issues

1) Nagios Probes
2) BDII info, silent failures
3) BDII not updating with system (reported & fixed in 6.1)
4) Empty RTE causing failuers (reported & fixed in 6.1)