ARC-CE v6 at ScotGrid Glasgow

Dr. Emanuele Simili

29/08/2019 @ GridPP43

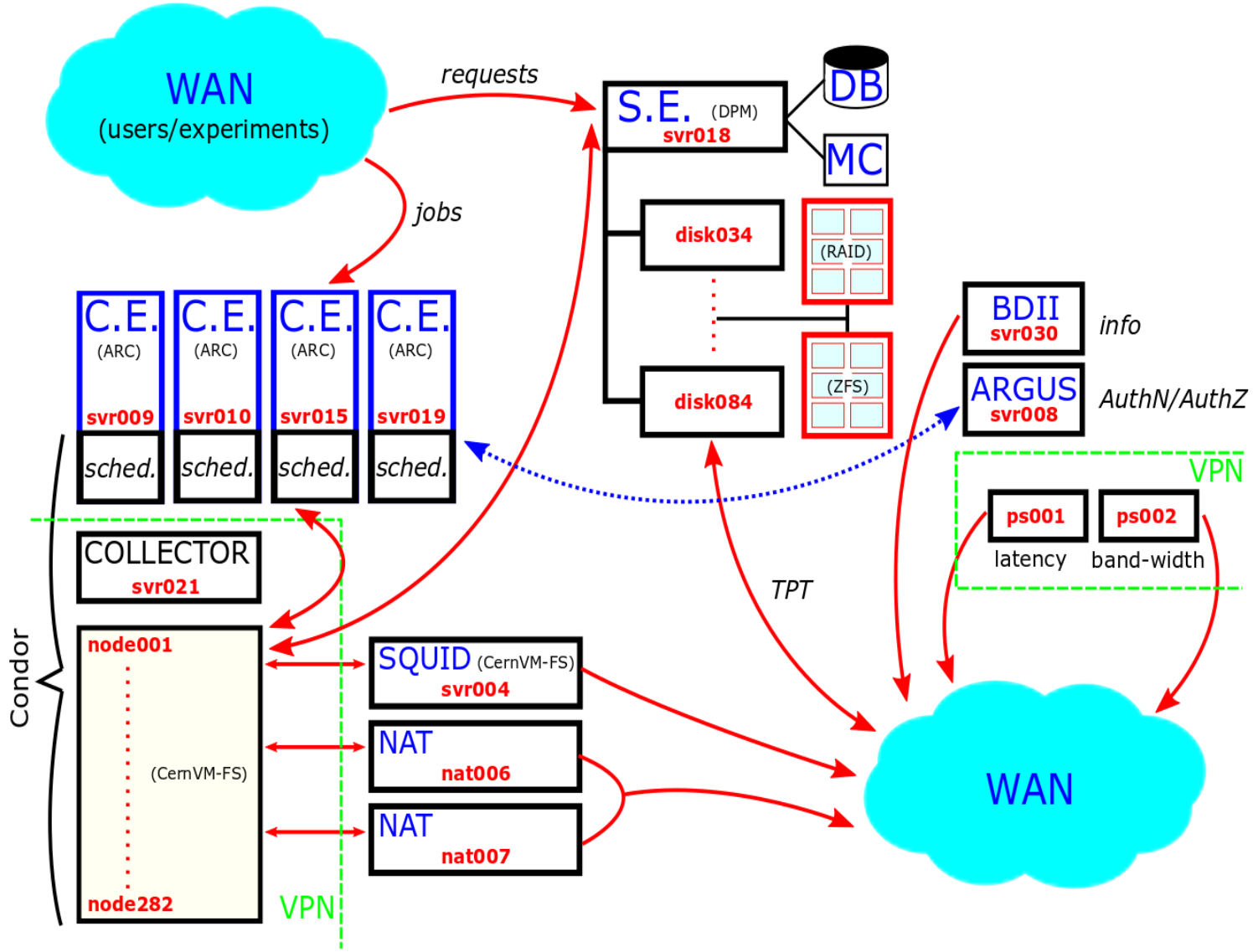
Outline

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

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

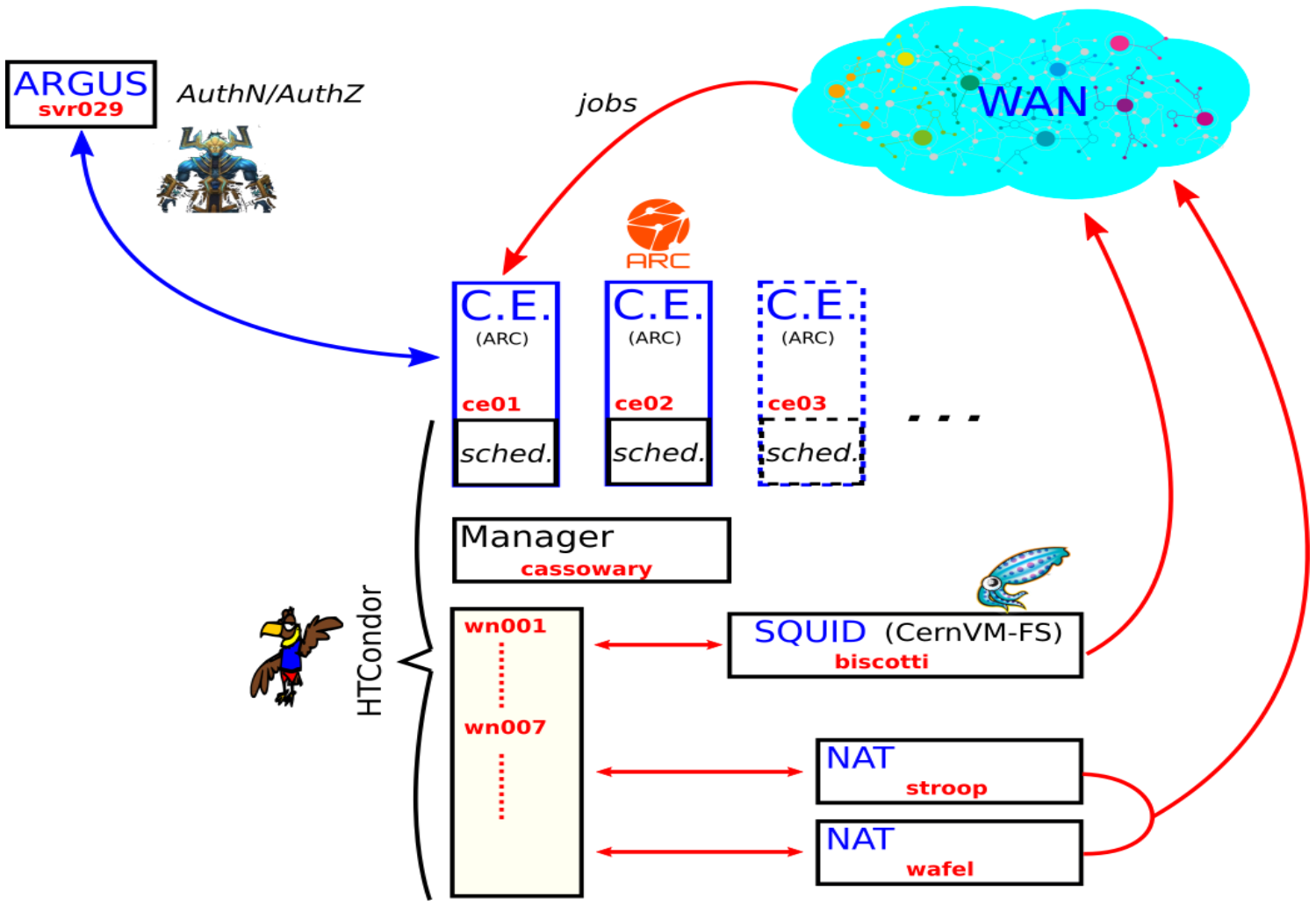
ARC-CE: The Advanced Resource Connector (ARC) Computing Element (CE) middleware, developed by the NorduGrid Collaboration, is an open source software solution enabling e-Science computing infrastructures with emphasis on processing of large data volumes.

Production Cluster Map



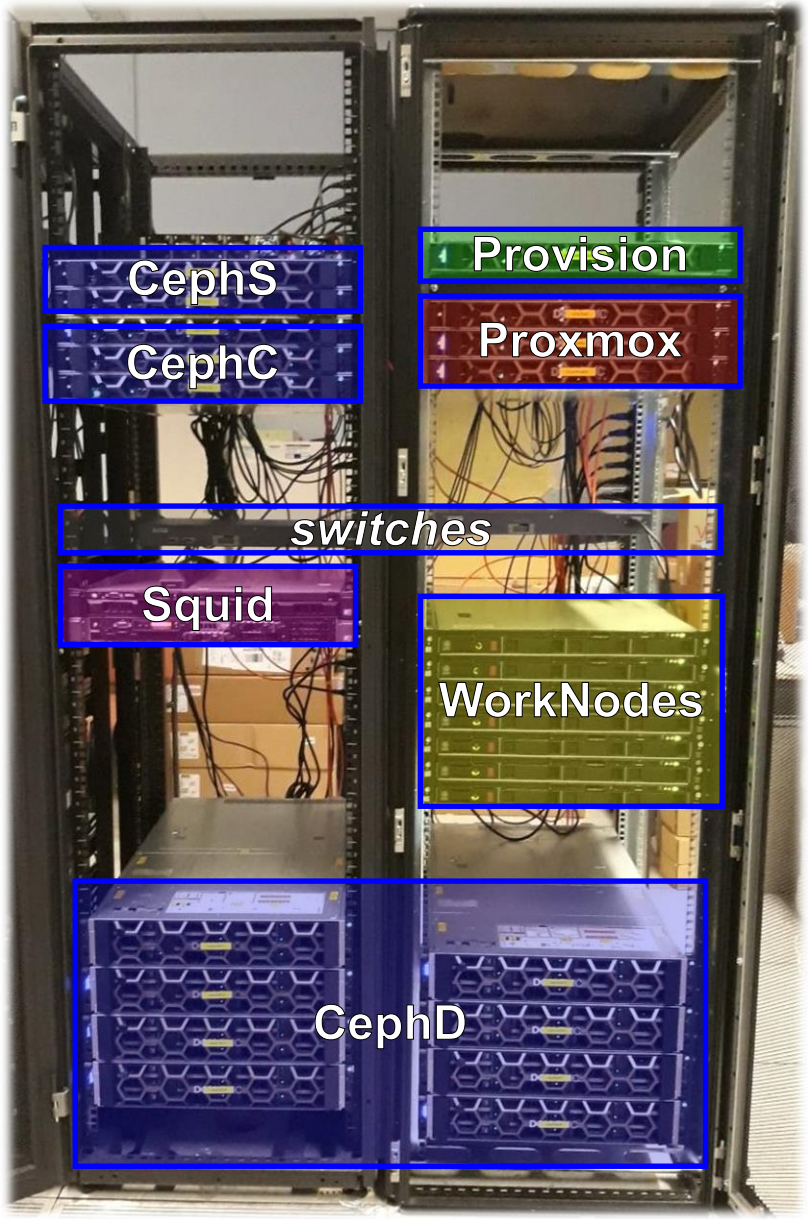
Service	Host	IPv4 (internal)	IPv4 (external)	VM
ARGUS	svr008	10.141.255.8	130.209.239.8	KVM
BDII-Site	svr030	10.141.255.30	130.209.239.30	
ARC-CE	svr009	10.141.255.9	130.209.239.9	
	svr010	10.141.255.10	130.209.239.10	
	svr015	10.141.255.15		
	svr019	10.141.255.19	130.209.239.19	
DPM	svr018	10.141.255.18		
Condor Collector	svr021	10.141.101.42		oVirt
Condor Master	condor-master	10.141.101.44		
Squid	svr004	10.141.255.24	130.209.239.24	
		10.141.201.1		
NAT	nat006	10.141.246.5	130.209.239.5	
	nat007	10.141.246.6	130.209.239.6	
perfSONAR	ps001	10.141.255.123	130.209.239.123	
	ps002	10.141.255.124	130.209.239.124	
Cobbler	provision	10.141.100.1		
		10.142.100.1		
VPN	svr024	10.141.255.24	130.209.239.24	
		10.141.201.1		
Node(s)	node001 - node282	10.141.0.1 - 10.141.1.29		
Disk(s)	disk034 - disk084	10.141.245.34 - 10.141.245.89	130.209.239.34 - 130.209.239.89	
Vac(s)	vac001 - vac044	10.141.213.1 - 10.141.213.44		

New Cluster Map (compute)



Hardware	Name	Model	IPv4 (int)	Service
40Gb Switch		Lenovo RackSwitch G8332	10.0.2.1	
10Gb Switch		Lenovo RackSwitch G8272	10.0.2.2	
Dell Server	Croquembouche	DELL R640	10.1.10.1	provision
Dell Server	Snicker	DELL R440	10.1.20.1	ProxMox
Dell Server	Doodle	DELL R440	10.1.20.2	
Dell Server	Krumkake	DELL R440	10.1.20.3	
1Gb Switch		Extreme Summit x440-48t-10G	10.142.101.241	*mngm*
10Gb Switch		Lenovo RackSwitch G8272	10.0.2.4	
10Gb Switch		Lenovo RackSwitch G8272	10.0.2.5	
HP Server	Wn001	HPE ProLiant DL60 Gen9	10.1.60.1	WorkNode
HP Server	Wn002	HPE ProLiant DL60 Gen9	10.1.60.2	WorkNode
HP Server	Wn003	HPE ProLiant DL60 Gen9	10.1.60.3	WorkNode
HP Server	Wn004	HPE ProLiant DL60 Gen9	10.1.60.4	WorkNode
HP Server	Wn005	HPE ProLiant DL60 Gen9	10.1.60.5	WorkNode
HP Server	Wn006	HPE ProLiant DL60 Gen9	10.1.60.6	WorkNode
HP Server	Wn007	HPE ProLiant DL60 Gen9	10.1.60.7	WorkNode
Dell Server	CephD5	DELL R740	10.1.50.25	CEPH disk
Dell Server	CephD6	DELL R740	10.1.50.26	CEPH disk
Dell Server	CephD7	DELL R740	10.1.50.27	CEPH disk
Dell Server	CephD8	DELL R740	10.1.50.28	CEPH disk
Dell Server	Biscotti	DELL R410	10.1.40.31	Squid Cache

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),	Provisioning
- 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):

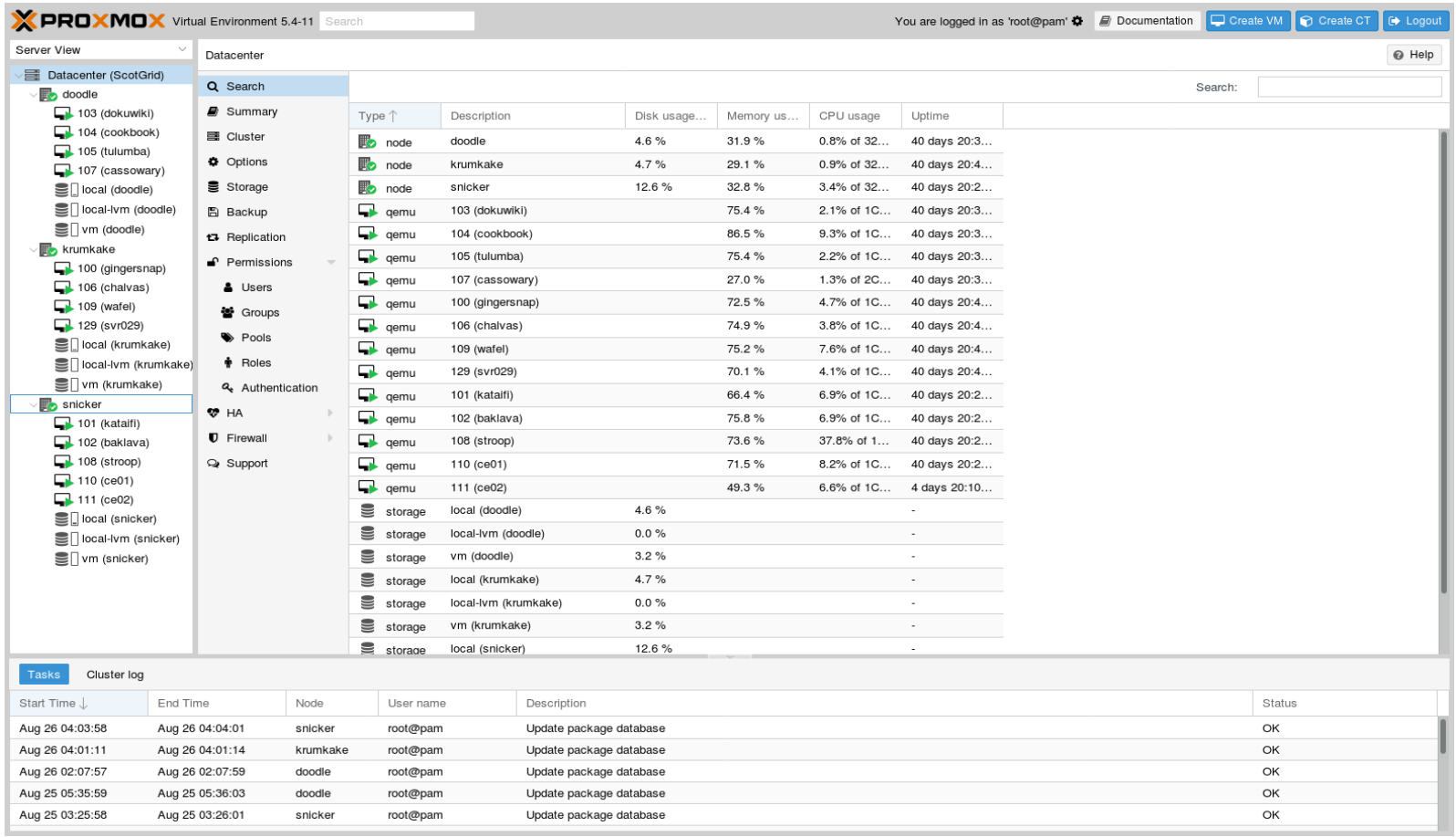
- Server DELL 740 (8x)	CEPH storage,
- Server DELL 440 (3x)	CEPH servers
- Server DELL 440+ (2x)	CEPH cache

ProxMox Hypervisor

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.

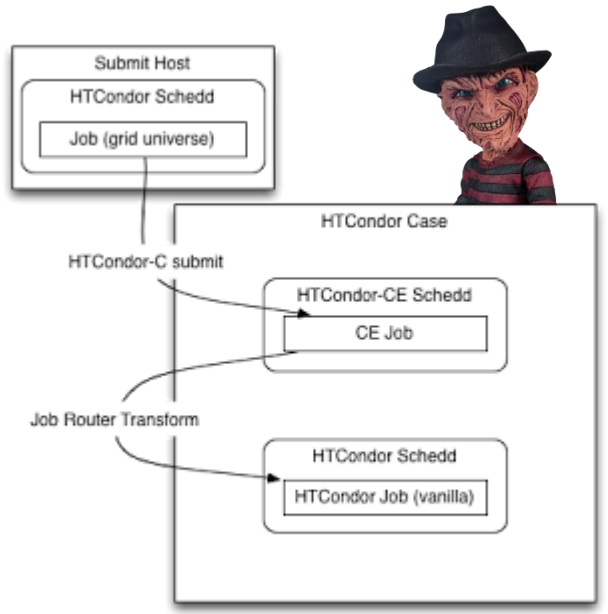
Settled on ProxMox:

- 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 ...

ARC-CE Installation

We first installed the newly released ARC6 following the instructions on NorduGrid website: http://www.nordugrid.org/arc/arc6/admins/arc6_install_guide.html
http://www.nordugrid.org/arc/arc6/admins/try_arc6.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:

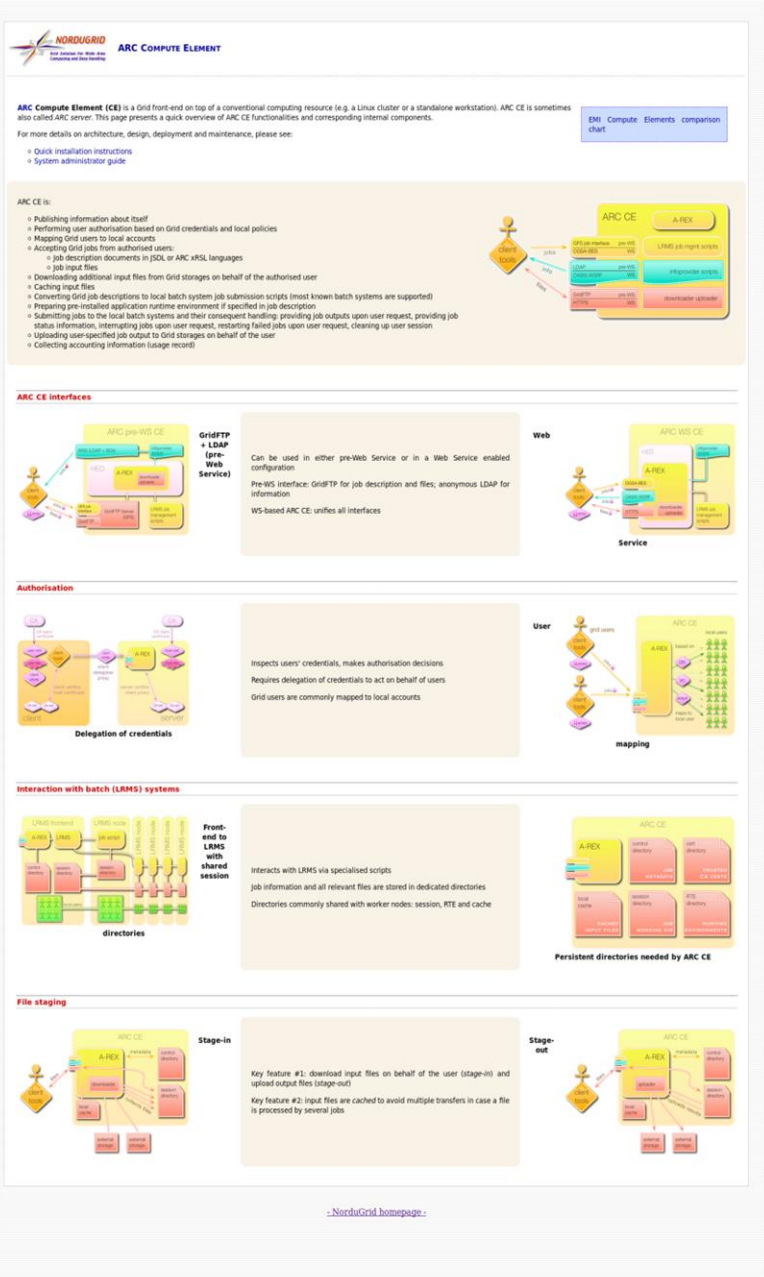
  yum -y install nordugrid-arc-arex

- 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*):

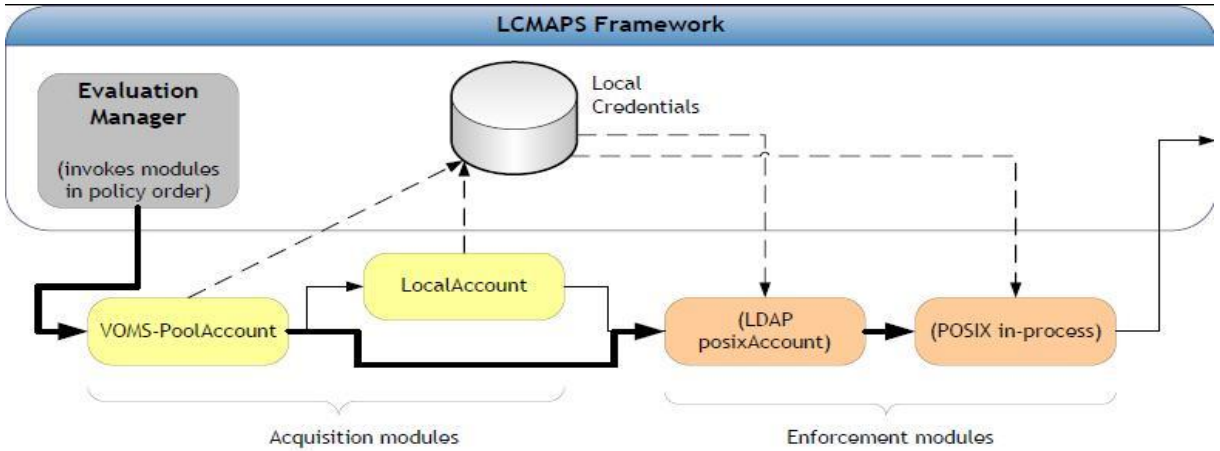
Section	Value	Purpose
`[lrms]`	`lrms = condor`	Defines the Local Resources Manager System as HTCondor
`[authgroup: all-vos]`	`voms = * * * *`	What VOs are allowed to do
`[gridftpd/jobs]`	`allowaccess = all-vos`	Who is allowed to submit jobs via gridftp
`[mapping]`	`map_with_plugin = all-vos 30 /usr/libexec /arc/arc-lcmaps %D %P liblcmaps.so /usr/lib64 /etc/lcmaps/lcmaps.db arc`	External plug-in for mapping users *
`[infosys/cluster]`	`advertisedvo = vo.scotgrid.ac.uk`	VOs that are allowed on the cluster
`[arex/jura/apel: EGI]`	`targeturl = https://mq.cro-ngi.hr:6162`	Target for Apel accounting data

Integration with **HTCondor**

lcmaps plug-in

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

See next slide …



Integration with **ARGUS**

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*)

```
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
```

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 !

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 ...

ARC-CE adding HTCondor

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).

Node type	Daemons
Worker Node	MASTER, STARTD
Manager Node	MASTER, SCHEDD, COLLECTOR, NEGOTIATOR
CE Node	MASTER, SCHEDD

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

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.

ARC-CE adding accounting

Turning on accounting in ARC6 is simple, only takes 2 steps:

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

```
[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
```

ARC-CE required BDII information

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 !

```
[infosys]
loglevel = INFO

[infosys/ldap]
bdii_debug_level= INFO

[infosys/nordugrid]

[infosys/glue2]
admindomain_name = UKI-SCOTGRID-GLASGOW

[infosys/glue2/ldap]

[infosys/cluster]
advertisedvo = atlas
advertisedvo = vo.scotgrid.ac.uk
advertisedvo = ops
advertisedvo = dteam
nodememory = 6000
defaultmemory = 2048
nodeaccess = outbound
alias = ScotGrid Glasgow
cluster_location = UK-G128QQ
cluster_owner = University of Glasgow
#homogeneity = False
```

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-gridftpd                    (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 ~]#
```

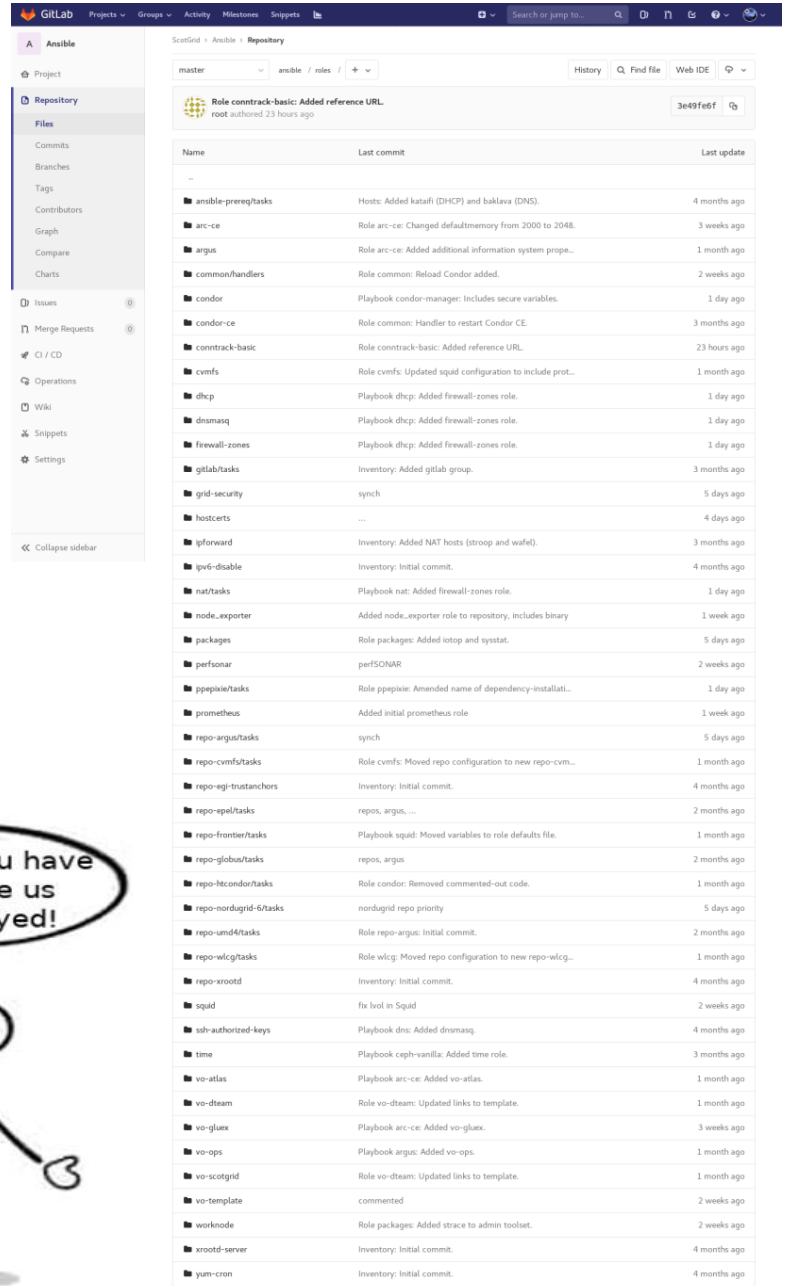
```
[root@ce01 ~]# arcctl accounting stats -t apel
Statistics for APEL jobs from 2019-07-18 10:42:33 till 2019-08-28 10:0
  Number of jobs:            62197
  Total WallTime: 2909 days, 22:31:54
  Total CPUTime:  2242 days, 17:00:58
[root@ce01 ~]#
```

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

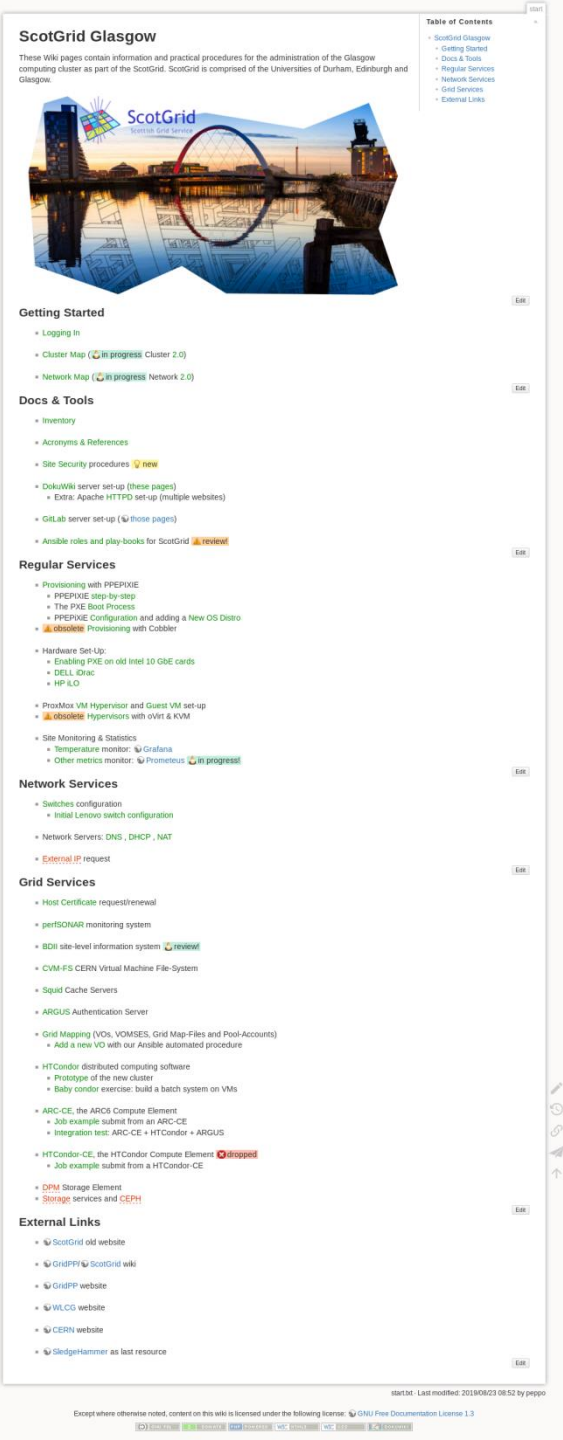
Cluster information and administrative procedures are safely organized in our internal Wiki, which is linked to the relevant Ansible roles in GitLab.

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.

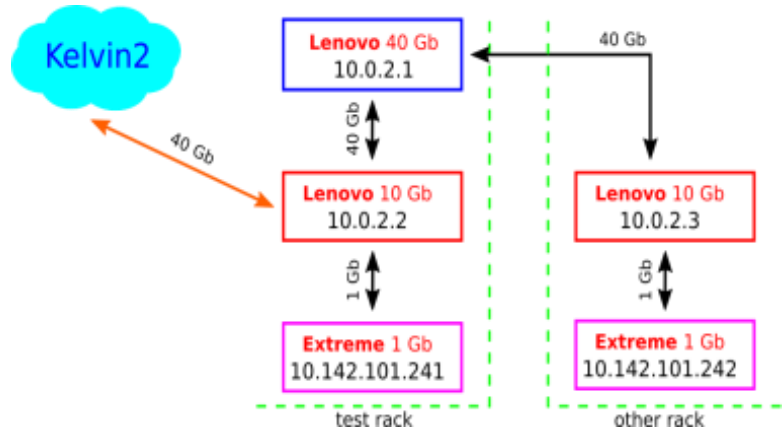
http://dokuwiki.beowulf.cluster/




The End

Physical Machines

Hardware	Name	Model	IPv4 (int)	Service
40Gb Switch		Lenovo RackSwitch G8332	10.0.2.1	
10Gb Switch		Lenovo RackSwitch G8272	10.0.2.2	
Dell Server	Croquembouche	DELL R640	10.1.10.1	provision
Dell Server	Snicker	DELL R440	10.1.20.1	ProxMox
Dell Server	Doodle	DELL R440	10.1.20.2	
Dell Server	Krumkake	DELL R440	10.1.20.3	
1Gb Switch		Extreme Summit x440-48t-10G	10.142.101.241	*mngm*
10Gb Switch		Lenovo RackSwitch G8272	10.0.2.4	
10Gb Switch		Lenovo RackSwitch G8272	10.0.2.5	
HP Server	Wn001	HPE ProLiant DL60 Gen9	10.1.60.1	WorkNode
HP Server	Wn002	HPE ProLiant DL60 Gen9	10.1.60.2	WorkNode
HP Server	Wn003	HPE ProLiant DL60 Gen9	10.1.60.3	WorkNode
HP Server	Wn004	HPE ProLiant DL60 Gen9	10.1.60.4	WorkNode
HP Server	Wn005	HPE ProLiant DL60 Gen9	10.1.60.5	WorkNode
HP Server	Wn006	HPE ProLiant DL60 Gen9	10.1.60.6	WorkNode
HP Server	Wn007	HPE ProLiant DL60 Gen9	10.1.60.7	WorkNode
Dell Server	CephD5	DELL R740	10.1.50.25	CEPH disk
Dell Server	CephD6	DELL R740	10.1.50.26	CEPH disk
Dell Server	CephD7	DELL R740	10.1.50.27	CEPH disk
Dell Server	CephD8	DELL R740	10.1.50.28	CEPH disk



Hardware	Name	Model	IPv4 (int)	Service
10Gb Switch		Lenovo RackSwitch G8272	10.0.2.3	
Dell Server	CephS01	DELL R440	10.1.50.1	CEPH server
Dell Server	CephS02	DELL R440	10.1.50.2	CEPH server
Dell Server	CephS03	DELL R440	10.1.50.3	CEPH server
Dell Server	CephC01	DELL R440 +	10.1.50.11	CEPH cache
Dell Server	CephC02	DELL R440 +	10.1.50.12	CEPH cache
1Gb Switch		Extreme Summit x440-48t	10.142.101.242	*mngm*
Dell Server	Biscotti	DELL R410	10.1.40.31	Squid Cache
Dell Server	Cantucci	DELL R610	10.1.40.33	VAC's Squid
Dell Server	CephD1	DELL R740	10.1.50.21	CEPH disk
Dell Server	CephD2	DELL R740	10.1.50.22	CEPH disk
Dell Server	CephD3	DELL R740	10.1.50.23	CEPH disk
Dell Server	ChepD4	DELL R740	10.1.50.24	CEPH disk

Virtual Machines

These services are running as VMs on ProxMox (see ProxMox VM Cluster).

Network Services (10.1.40.*)

Name	Service	IPv4
Kataifi	DHCP	10.1.40.1
Baklava	DNS	10.1.40.11
Tulumba	DNS	10.1.40.12
Chalvas	DNS	10.1.40.13
Stroop	NAT	10.1.40.21
Wafel	NAT	10.1.40.22

Miscellaneous Services (10.1.41.*)

Name	Service	IPv4
Dokuwiki	Wiki	10.1.41.1
Cookbook	Gitlab	10.1.41.2

Monitoring Services (10.1.42.*)

Name	Service	IPv4
Gingersnap	Prometheus	10.1.42.1
…		

HTCondor batch system (10.1.43-45.*)

Name	Service	IPv4 (int)	IPv4 (ext)
Cassowary	Manager	10.1.43.1	
Ce01	ARC-CE	10.1.44.1	130.209.239.25
Ce002	ARC-CE	10.1.44.2	130.209.239.122
Svr029	ARGUS	10.1.45.1	130.209.239.29

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

  https://research.cs.wisc.edu/htcondor/yum/repo.d/htcondor-stable-rhel7.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 ...

Node type	Daemons
Worker Node	MASTER, STARTD
Manager Node	MASTER, SCHEDD, COLLECTOR, NEGOTIATOR
CE Node	MASTER, SCHEDD

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

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

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)