CentOS7

Ste, John and Rob

University of Liverpool

April 2018

Introduction

- Part one of the talk is about the way that CentOS 7 has impinged on our site. We use ARC/Condor (VAC is still SL6). I'm going to talk a bit about:
  - Why and when we converted,what was available and who was involved.
  - Where did we draw the code from.
  - What systems did we migrate, how we did it, major component versions, and what issues we have seen.
  - How much take up there was by the VOs.
  - How busy has the C7 cluster been kept by the VOs.
- (Note: so much has come to light researching the talk that I'm practically overwhelmed. For this, I apologize and I'll try to only say things that have a decent chance of being useful.)

Introduction

- Part two is about general things related to adoption of CentOS 7 and configuration reuse (via Puppet Modules mainly):
  - An informal view of where T2 sites are at.
  - Prospects for a reusable configuration Puppet definition.
  - List of available resources.
  - Partial tests of reuse (TBD).
  - Some of the issues that cropped up.

Introduction

- Part three describes how to roll you own CentOS7 worker node using a very general config control tool called “bash”.
- I loosely specify what needs to be done using a document that exhibits high-cohesion and low-coupling (that’s a fancy way of saying it’s short and sweet.)
- I’ll show that setting up a worker node by hand is fairly trivial.
    - That’s how we used to always do it.
- Objective: show the inordinate disconnect between the size of the problem and the tiers of technology needed to manage it, while conceding that there is no silver bullet (yet.)

Part 1 - Liverpool C7

- Why adopt CentOS7: A year and half ago (or was it longer?) there was much less CentOS7 take up. ECDF had some nodes running due to local constraints which precluded SL6. The UMD4 release manager at that time (Andrea Manzi) wanted more sites for tests. ATLAS told us, via Alessandra, that they were getting keen to move.
- Yaim had left the building, so we rolled our own suite of puppet modules that grew over time. This may or may not have been a mistake. All our material can be got from git-hub.
  - git clone https://github.com/sjones-hep-ph-liv-ac-uk/arc_condor_c7.git
- There's a ./doc/ directory in there, that contains an full rundown, called C7ArcCondor.odt. It is what it is (we just run a site, not a project management office, so these are rough-cut but working sysadmin tools.)

Part 1 - Liverpool C7

- Here are the online repositories that we use to run the C7 cluster.

  Worker node:

  - CA
    http://repository.egi.eu/sw/production/cas/1/current
  - Condor
    http://research.cs.wisc.edu/htcondor/yum/stable/rhel7
  - Cvmfs
    http://cern.ch/cvmrepo/yum/cvmfs/EL/7/x86_64
  - Epel
    ftp://ftp.mirrorservice.org/sites/download.fedora.redhat.com/pub/epel/7/x86_64/
  - wlcg_repository
    http://linuxsoft.cern.ch/wlcg/sl6/x86_64
  - Umd_4
    http://repository.egi.eu/sw/production/umd/4/centos7/x86_64/base/
    http://repository.egi.eu/sw/production/umd/4/centos7/x86_64/updates/

Part 1 - Liverpool C7

- ARC/Condor Headnode; as per WN, plus the repository for Nordugrid ARC:
  - Nordugrid
    http://download.nordugrid.org/repos/15.03/centos/el7/x86_64/

- I'll also plug this document for SL6 that is still highly relevant but tending to get stale now. It lays out the ARC/Condor customisations that we applied.
  - https://www.gridpp.ac.uk/wiki/Example_Build_of_an_ARC/Condor_Cluster

Part 1 - Liverpool C7

- Major component versions.
  - CentOS: 17.4.1708
  - ARC: 5.4.1-1
  - HTCondor: 8.6.3-1
  - Argus:
    - argus-pep-server-1.7.3-1.el7.centos.noarch
    - argus-pap-1.7.1-1.el7.centos.noarch
    - argus-pdp-1.7.0-1.el7.centos.noarch
    - argus-authz-1.7.0-1.el7.centos.noarch
  - wn: 4.0.5-1
  - DPM: 1.9.2-1

Part 1 - Liverpool C7

- So, a year or so later, we have home-grown Hiera/Puppet 3 modules for :
  - HTCondor Workernodes
  - ARC/HTCondor CEs
  - ARGUS (Server and Clients)

- All of it is covered in the run down in Github.

- For DPM on CentOS7, John Bland set it up using the “vendor supplied” puppet modules, and the puppet apply command:
  - puppet apply dpm-config
- John’s DPM slides follow.

Part 1 - Liverpool C7

- Our DPM on CentOS7 is installed using standard kickstart and central puppet for local settings eg firewalls, ssh access, monitoring etc. All our DPM systems run C7.4 but mixing SL6/C7 and different versions of DPM has generally been ok.
- We then use DPM puppet modules to configure DPM using 'puppet apply'. Local puppet ensures httpd is running but otherwise doesn't touch it.
- We had to do some jiggery-pokery with systemd and system limits like open file numbers to get mysql to work properly with DPM. Not obvious at all.

Part 1 - Liverpool C7

- DPM by default has a cron job that gracefully restarts httpd. Not had any issues on pool nodes, occasionally this leaves httpd on the headnode in a broken state. Restarting fully fixes it. httpd stability is the only major problem we have with DPM.
- Other than getting to grips with systemd/firewalld not much difference between SL6 and C7 for servers (desktops were a much bigger challenge with lots of silly bugs, workarounds, broken backwards compatibility and general opaque and unpredictable behaviour.)
- In general I've found the transition from SL6 to C7 to be more time consuming, problematic and frustrating than any of the SL4>SL5>SL6 upgrades. Maybe I'm just old fashioned.
  - (My note: I had a hard time with it as well)

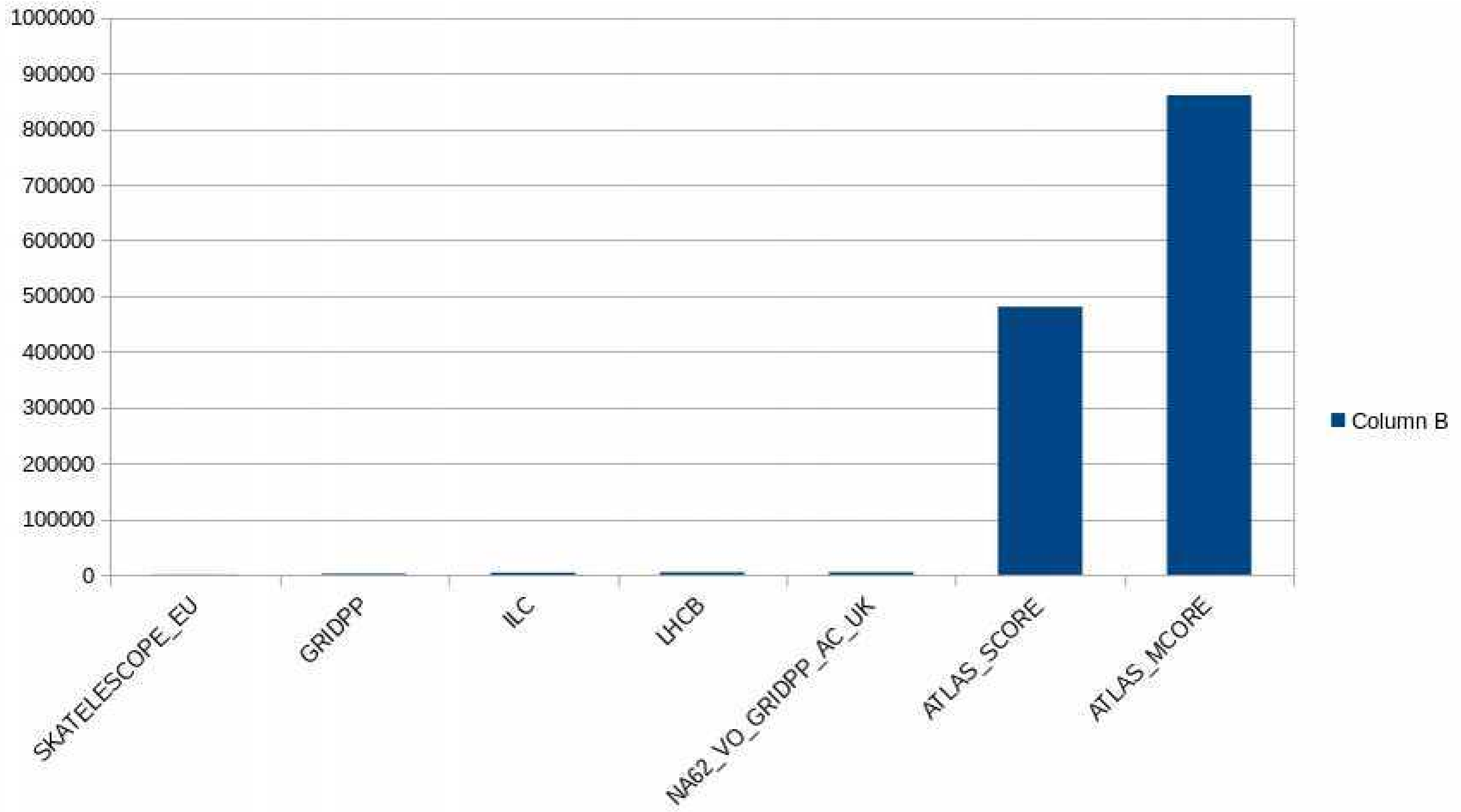
Operations

- On the compute side, we have had no showstoppers
- The main problems have been file system deadlocks, and sporadic kernel panics; too infrequent to cause outrage or horror.
- These turn out to come from XFS. Depending when they happen, you may lose a job, all the jobs or (equivalently) the whole machine. Often locked in cleanup, “rf -rf <condor_pool/jobdir>”, but STARTD single-threaded, so it’s game over.
- It seemed more likely to happen with the out-of-band firmware release given by Intel (later retracted) after Spectre/Meltdown, but it is not limited to patched systems. I don’t know which has the most impact.
- Shotgun solution is to revert patch and use EXT4. It’s a shame because XFS is much faster than EXT4 for making the users (i.e. many small files).

VO Take up

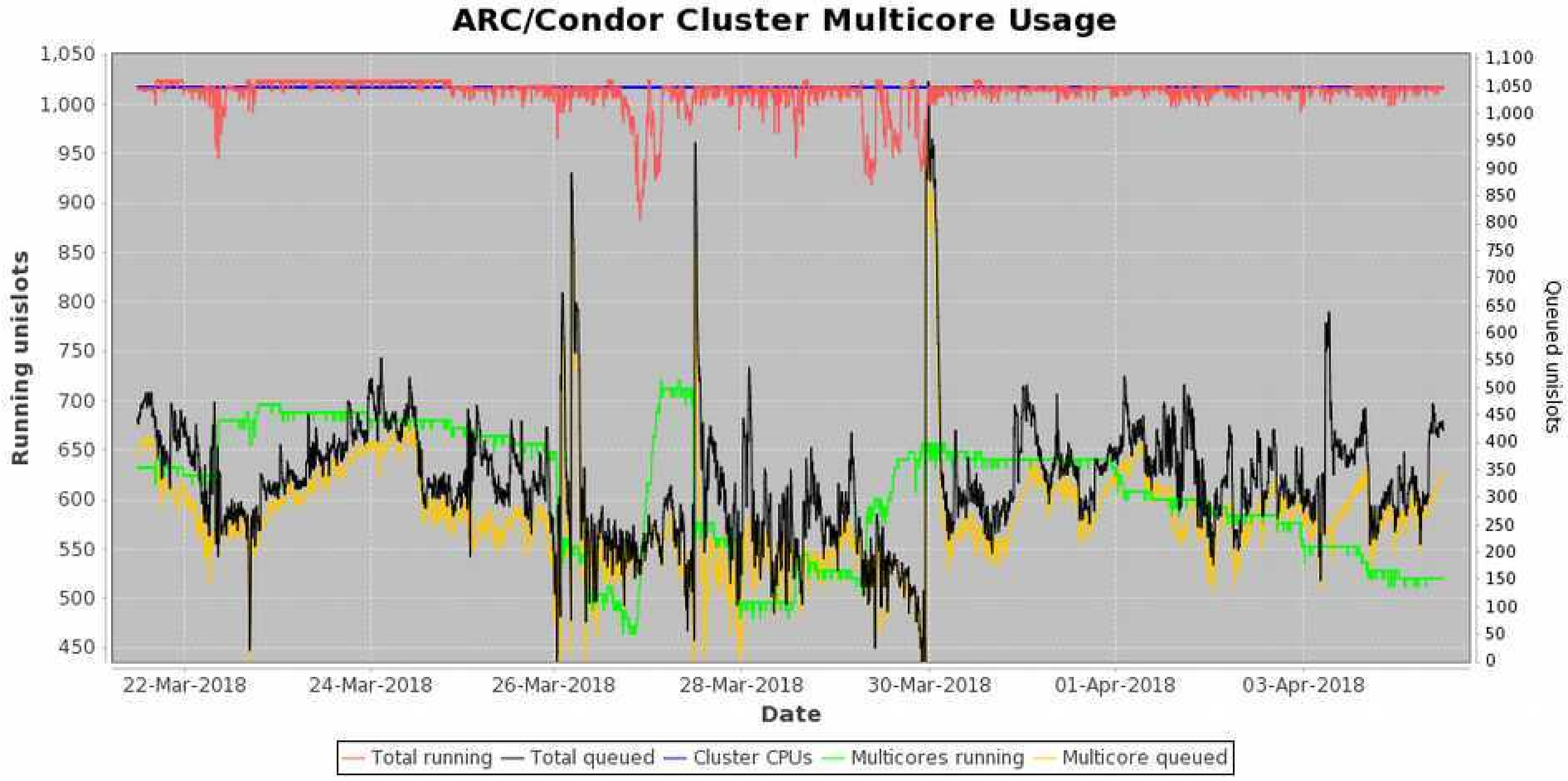
- There have been plenty of VOs who have taken a sniff of the CentOS7 cluster. Each of the following has run at least one job: atlas, lhcb, biomed, calice, cernatschool.org, dteam, gridpp, ilc, lsst, lz, na62.vo.gridpp.ac.uk, ops, pheno, skatelescope.eu, snoplus.snolab.ca, vo.northgrid.ac.uk.
- But, as the next slides show, the vast majority of the CentOS7 work is coming from ATLAS. Nonetheless, the work has consistently kept the C7 cluster full with a good mix of MCORE and SCORE jobs so we don’t see empty slots.
- We still run a rump SL6 cluster, which attracts LHCb and many small VOs, and we run 700 slots on VAC, mostly used by LHCb, so the job balance is not far off.

VO Take up (C7)


1000000
900000
800000
700000
600000
500000
400000
300000
200000
100000
0
SKATELESCOPE_EU
GRIDPP
ILC
LHCB
NA62_VO_GRIDPP_AC_UK
ATLAS_SCORE
ATLAS_MCORE
■ Column B

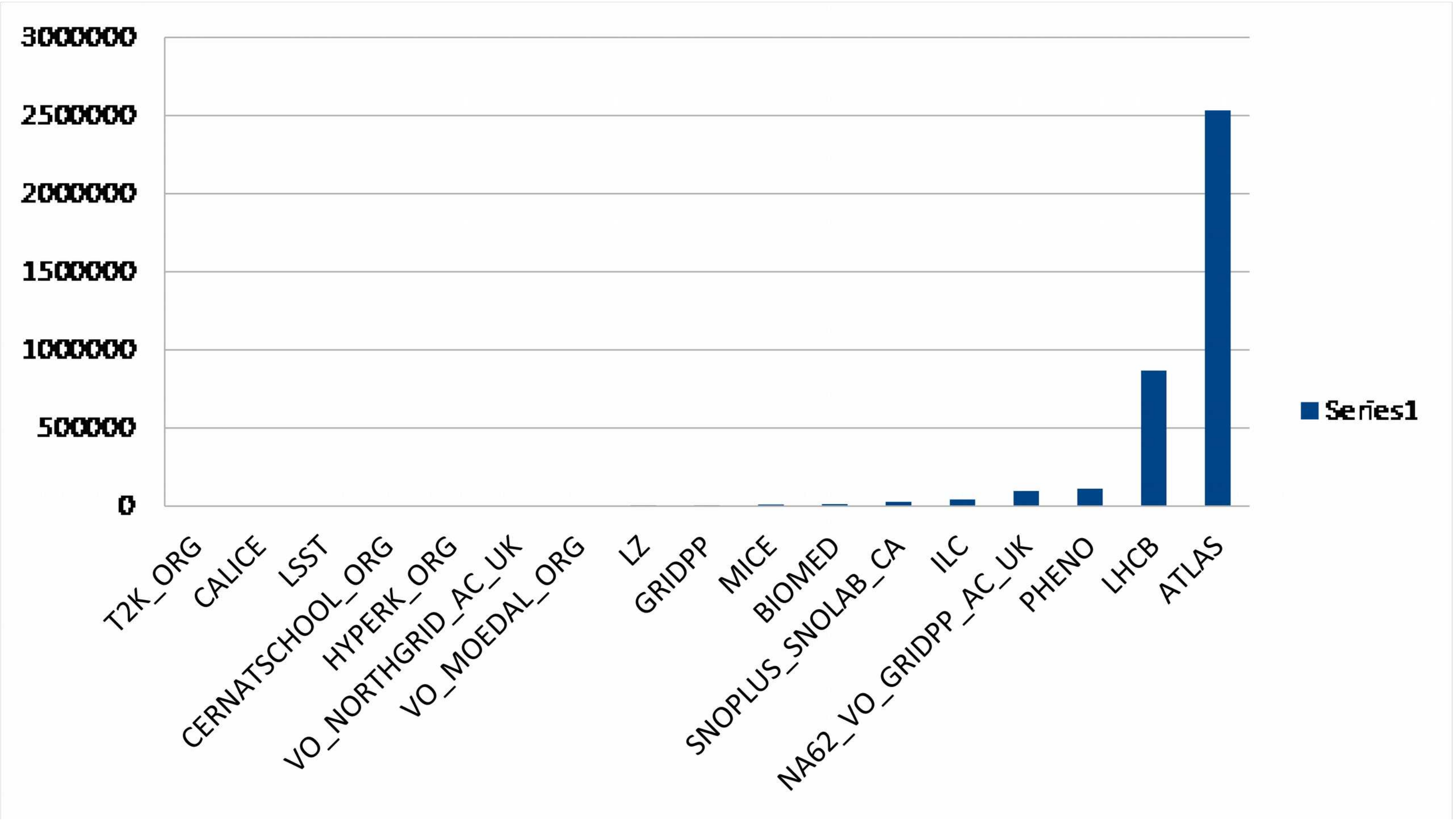
GridPP
UK Computing for Particle Physics

VO Take up



GridPP
UK Computing for Particle Physics

VO Take up (SL6)



Part 2 Prospects for reuse

- I wanted to do a survey to determine the current status of CentOS7 on the sites. This didn’t happen, but I did make informal estimates of the levels and types of CentOS7 take up.
- I think all sites are at least in the planning phase or further along, but a number sites (far fewer than half in my estimation) haven’t any operational CentOS7 system components at this point.
- The view I formed is that a significant number of sites are far along with the migration; some are practically complete, while other sites have a sizeable part of their infrastructure on the CentOS7 or some similar RedHat style system.

Part 2 Prospects for reuse

- There is no strong adoption of a common method for configuration control (build and on-going) of the CentOS7 operational baseline.
- Yaim is less relevant than ever (although components of Yaim can be used in “function” mode to perform familiar tasks such as making the pool of users or populating gridmapdir etc.)
- Ansible is starting to make ground but is still novel.
- Puppet is more usual but there is fragmentation nonetheless. Component providers (e.g. DPM, WN, HTCondor…) provide puppet modules but they have a learning curve.
- Hence there is a temptation to “roll your own”, with the usual “whipupability” versus “forking disadvantages” that are entailed.

Part 2 Prospects for reuse

- One driver of the trend to roll you own is that fact that it is much easier to write a single use Puppet module than it is to write a reusable module (whipupability.)
- Coupled with that is the fact that Puppet is (or has been) in a state of flux, where compatibility across versions is variable, making it harder to consolidate admins and devs on a single path.
- Whatever the truths of these assertions, the fact is that when Liverpool implemented its CentOS7 baseline, UMD4 was in test and there were fewer good Puppet resources than there are now… (PS: I'm being as fair as I can be here…. all of this is qualitative and hence open to question.)

Part 2 Prospects for reuse

- Anyway, there's no use just making assertions. We need evidence about the prospects for reuse.
- Several sites, Manchester, Bristol, perhaps others, have engaged in the reuse program with some success. Patches have been delivered to Puppet module authors to incrementally consolidate the software.
- This is commendable; we all know the time-honoured difficulties involved with forking and fragmentation.
- However, communication without central planning (e.g. Yaim) are problematic. Hence I decided to (try to) convert Liverpool to the "standard path" to test the viability.
- Unfortunately, the 'flu got in the way of this work, so the results are only fragmentary.

Part 2 Prospects for reuse

- So how did it go ….
- Lukaz (Bristol) gave a list of "plausible" reusable modules that I present in Appendix 1 of this talk.
- I started (and finished!) with the CentOS7 EGI wn (workernode) package. No time to get to HTCondor config.
- First thing I had to do was create a new Puppet server; I felt had to move from version 3 to 4, on seeing difficulties.
- After several days, I got the wn package up and going on a workernode. So it was a positive result, but I got no further.
- Only caveat; not possible with "standard" puppet module to make the accounts etc. in the exact same way as Yaim, hence parallel running SL6 and CentOS7 would be harder. Long discussion on this with AF.
- Many more test conclusions available but not yet written up.

Part 2 Prospects for reuse

- Further actions:
  - Complete these tests, convert Liverpool to use “standard puppet modules”, report on output of this task. Feed patches back if necessary.
  - Action outstanding from a technical meeting to set up document infrastructure that lays out the way each site has approached CentOS7 migration.

Part 3 Rolling your own

- This is my little party piece.
- If I specify a system too tightly, the rules soon become too unwieldy to read, understand, apply or keep up to date. People ignore it (the manual is only the final option.)
- And if I specify the system too loosely, the rules leave too many loopholes and everything goes to pot in a different, (sloppy) sort of way.
- Theory: if the rules are terse and easy to change, then can it be tightly specific enough and still kept up to date enough? Andrew McNab touched on baseline driven change. It’s used a lot in aerospace. It works but it’s expensive.
- Anyway, this is a a real problem for systems engineers and there is no silver bullet AFAIK. But I want to try it out…

Notation

- So let’s just try something simple.
  - Objective: I’ll find out exactly what these mountains of config control technology actually do.
  - Take a plain vanilla worker node – no middleware.
  - Compare it ‘en masse’ to a built Puppetised workernode and grab the deltas.
  - Copy the config from the Puppetised worker node to PVWN by hand. Document how to do it.
  - Write it up and call it a “minimal spec”
- The next slide shows the boiled down requirements that I got from this exercise. It’s two (mostly white) sheets of A4.

Minimal

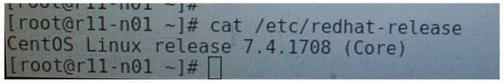
```
CentOS7 Worker-node

Prerequisites:

 OS is CentOS 7.4
 Enabled yumrepos are base, updates, extras
 Kernel is 3.10.0-693.17.1
 Network and firewall setup, with dns or /etc/hosts names for all systems

Yumrepos:

 Prerequisites plus CA, umd/4/centos7 (base, update), wlcg, epel,
   cvmfs (maybe -config and -testing), htcondor

RPMS:
 ca-policy-egi-core, ca-policy-lcg, lcg-CA
 wn, HEP_OSlibs, condor, mjf-htcondor,
 lfc-python, yum-plugin-priorities, yum-plugin-protectbase, voms-clients-cpp

Configuration:

 wn:
   make the users and groups [mug]
   optionally populate the /etc/grid-security/gridmapdir [gmd]
   optionally populate the /etc/grid-security/vomsdir [vd]
   optionally populate the /etc/vomses [vs]
   $ fetchcrl

 cvmfs:
   Configure cvmfs[cvmfs]

 arc:
   Set up arc runtime dirs[rt]

 condor:
   Copy in /etc/condor/pool_password
   Copy in /etc/condor/config.d/condor_config.local
   Copy in /etc/condor/config.d/00-node_parameters
   Copy in /etc/profile.d/liv-lcg-env.*
   $ touch /usr/etc/globus-user-env.sh
   $ mkdir /etc/condor/ral   # liv std
   $ mkdir /data/condor_pool # liv std
   $ chown condor:condor /data/condor_pool

Finally:
 Reboot
 Start condor services
```

```
Routines:

 mug():
   Make the users and groups
   Copy in makeNewUsers.pl, users.conf
   $ ./makeNewUsers.pl users.conf

 gmd():
   Make the gridmapdir
   Copy in createGridMapDirUsers.pl
   $ ./createGridMapDirUsers.pl users.conf

 vd():
   Populate the /etc/grid-security/vomsdir
   TBD

 vs():
   Populate the /etc/vomses
   TBD

 cvmfs():
   Configure cvmfs
   $ yum install cvmfs # maybe gpgcheck=0
   Copy in default.local
   $ vi /etc/fuse.conf # user_all_other option
   $ cvmfs_config setup
   $ cvmfs_config chksetup
   $ cvmfs_config probe

   Maybe reboot needed now ...
   $ systemctl status autofs.service
   $ systemctl restart autofs.service

   $ ls /cvmfs/atlas.cern.ch

  rt():
   Set up arc runtime dirs
   $ mkdir  /etc/arc
   $ mkdir  /etc/arc/runtime
   $ cd /etc/arc/runtime
   $ mkdir ENV
   $ touch ./ENV/GLITE
   $ touch ./ENV/PROXY
   $ touch ./GLITE-3_1_0
   $ touch ./LCG-2_1_1
   $ touch ./LCG-2_3_0
   $ touch ./VO-t2k.org-ND280-v10r11p25
   $ touch ./LCG-2_7_0
   $ touch ./VO-lhcb-pilot
   $ touch ./VO-t2k.org-ND280-v10r11p29
   $ touch ./LCG-2_2_0
   $ touch ./R-GMA
   $ touch ./LCG-2_5_0
   $ touch ./LCG-2_6_0
   $ touch ./VO-t2k.org-ND280-v11r21
   $ touch ./LCG-2_1_0
   $ touch ./VO-snoplus.snolab.ca-cvmfs
   $ touch ./LCG-2
   $ touch ./GLITE-3_0_0
   $ touch ./VO-biomed-CVMFS
   $ touch ./LCG-2_3_1
   $ touch ./LCG-2_4_0
   $ touch ./HYDRA-CLIENT-3_1

Resources:
```

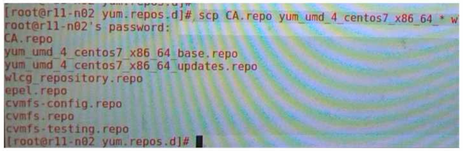
Notation

- Once I knew what I was doing, it took about 25 minutes to configure the PVWN node and get a job running on it.
- The boiled down steps resolve to - make PVWN, add yum repos, add rpms, make users and groups, run fetchcrl, configure cvfms, configure condor, make a few files (mostly empty) and dirs , reboot and start the services.
- It’s not rocket science, and it’s hard to understand why (say) Puppet needs so much TLC. Note to self: must look at Ansible…
- Perhaps we must stick with the “hands off” approach (i.e. use a Puppet/Ansible config definition system) though, since it ostensibly makes ongoing maintenance easier. But I’m starting to question whether the set-up is commensurate with the task.
- So … still no silver bullet, as ever.

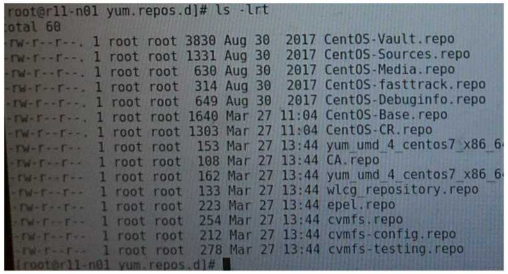
Results



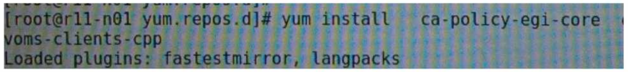
```
[root@r11-n01 ~]# cat /etc/redhat-release
CentOS Linux release 7.4.1708 (Core)
[root@r11-n01 ~]# 
```



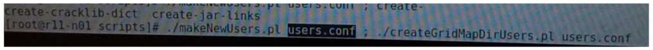
```
[root@r11-n02 yum.repos.d]# scp CA.repo yum_umd_4_centos7_x86_64_* w
root@r11-n02's password:
CA.repo
yum_umd_4_centos7_x86_64_base.repo
yum_umd_4_centos7_x86_64_updates.repo
wlcg_repository.repo
epel.repo
cvmfs-config.repo
cvmfs.repo
cvmfs-testing.repo
[root@r11-n02 yum.repos.d]# 
```



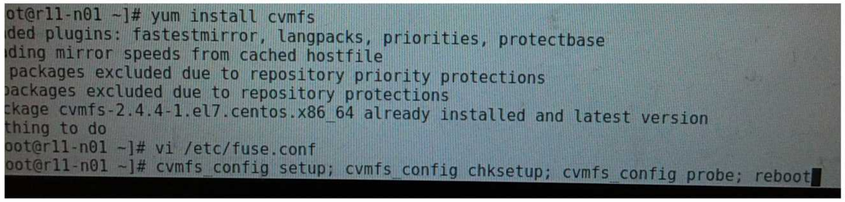
```
root@r11-n01 yum.repos.d]# ls -lrt
otal 60
-rw-r--r--. 1 root root 3830 Aug 30  2017 CentOS-Vault.repo
-rw-r--r--. 1 root root 1331 Aug 30  2017 CentOS-Sources.repo
-rw-r--r--. 1 root root  630 Aug 30  2017 CentOS-Media.repo
-rw-r--r--. 1 root root  314 Aug 30  2017 CentOS-fasttrack.repo
-rw-r--r--. 1 root root  649 Aug 30  2017 CentOS-Debuginfo.repo
-rw-r--r--. 1 root root 1640 Mar 27 11:04 CentOS-Base.repo
-rw-r--r--. 1 root root 1303 Mar 27 11:04 CentOS-CR.repo
-rw-r--r--  1 root root  153 Mar 27 13:44 yum_umd_4_centos7_x86_6
-rw-r--r--  1 root root  108 Mar 27 13:44 CA.repo
-rw-r--r--  1 root root  162 Mar 27 13:44 yum_umd_4_centos7_x86_6
-rw-r--r--  1 root root  133 Mar 27 13:44 wlcg_repository.repo
-rw-r--r--  1 root root  223 Mar 27 13:44 epel.repo
-rw-r--r--  1 root root  254 Mar 27 13:44 cvmfs.repo
-rw-r--r--  1 root root  212 Mar 27 13:44 cvmfs-config.repo
-rw-r--r--  1 root root  278 Mar 27 13:44 cvmfs-testing.repo
[root@r11-n01 yum.repos.d]# 
```



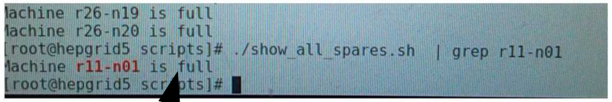
```
[root@r11-n01 yum.repos.d]# yum install   ca-policy-egi-core
voms-clients-cpp
Loaded plugins: fastestmirror, langpacks
```



```
create-cracklib-dict  create-jar-links
[root@r11-n01 scripts]# ./makeNewUsers.pl users.conf ; ./createGridMapDirUsers.pl users.conf
```



```
ot@r11-n01 ~]# yum install cvmfs
ded plugins: fastestmirror, langpacks, priorities, protectbase
ding mirror speeds from cached hostfile
 packages excluded due to repository priority protections
packages excluded due to repository protections
ckage cvmfs-2.4.4-1.el7.centos.x86_64 already installed and latest version
thing to do
oot@r11-n01 ~]# vi /etc/fuse.conf
oot@r11-n01 ~]# cvmfs_config setup; cvmfs_config chksetup; cvmfs_config probe; reboot
```



```
achine r26-n19 is full
achine r26-n20 is full
[root@hepgrid5 scripts]# ./show_all_spares.sh  | grep r11-n01
achine r11-n01 is full
[root@hepgrid5 scripts]# 
```

It’s hard to see, but the test machine’s single slot is full. The job ran to the end.

Wrapup

- I’ve talked about why we upgraded, where the software is that we used, what services we upgraded, how we configured it.
- I’ve talked about trade-offs between home-grown modules and some “standard” reusable modules.
- We’ve developed, used and described our own home-grown modules and found them to be “satisfactory”.
- We’ve used the DPM standard Puppet modules and found them to be satisfactory.
- We’ve tested the WN standard module, and found that to be satisfactory with some caveats.
- I’ve measured VO takeup and busi-ness of the CentOS7 cluster and compared it to an SL6 cluster and found them comparable.
- We’ve touched on operational issues that came up with CentOS7, and our general experiences.
- We’ve provided a list of resources that we hope will help.
- We’ve got some outstanding actions that we could not complete due to “external issues”.

Appendix 1

Reusable Puppet modules (thanks to Luke Krezko, Bristol)

- https://github.com/HEP-puppet (HEP-Puppet gives you the right entry point for the bare modules; when checking out, omit puppet- prefixes… This is where I obtained the wn module.)

Other useful puppet module locations:

- https://github.com/cern-it-sdc-id (albeit handle with care due to CERN specific bits)
- https://github.com/cvmfs/puppet-cvmfs
- https://github.com/cernops/puppet-htcondor_ce (BDII, but no APEL)

Luke has also made the non-secret parts of our Puppet setup public. Which gives an idea how these modules can be tied together.

- https://github.com/uobdic/UKI-SOUTHGRID-BRIS-HEP

End

Questions:

sjones@hep.ph.liv.ac.

jbland@hep.ph.lib.ac.uk

fay@hep.ph.liv.ac.uk