Liverpool accounting audit tools

- It’s nice to be sure all the work you do gets into APEL. It’s even nicer to find stray work that hasn’t been sent. Two ways to make sure(r) are:
  - Compare portal to theoretical site maximum (minus estimate of the “deadtime”).
  - Work out what should be there by looking at the sources of the data, and compare that.
- I’ll briefly discuss these ideas, and give examples* for ARC/HTCondor and VAC.

* Similar methods exist for CREAM/Torque also, but we don’t use that any more at Liverpool (so they are perhaps not current.)

Query the portal

- https://accounting.egi.eu/
- Research infrastructure/T2 grid accounting
- Drill down to country, region, site
- Set metric to Sum Wallclock Work HS06 Hours
- Set start and end time e.g. to August 2018
- VO Filtering and Job Submission “All”
- Row variable is submit host
- Column variable leave as Date

Result of query

First are the ARC CEs:

```
gsiftp://hepgrid2.ph.liv.ac.uk:2811/jobs 5,572,132  5,572,132    31.8%
gsiftp://hepgrid5.ph.liv.ac.uk:2811/jobs 7,642,634  7,642,634    43.62%
```

Then all the VAC nodes:

```
vac01.ph.liv.ac.uk/vac-comp500.bullx 28,970 28,970 0.17%
vac01.ph.liv.ac.uk/vac-comp501.bullx 46,393 46,393 0.26%
vac01.ph.liv.ac.uk/vac-comp503.bullx 29,073 29,073 0.17%
vac01.ph.liv.ac.uk/vac-comp504.bullx 34,389 34,389 0.2%
vac01.ph.liv.ac.uk/vac-comp506.bullx 39,502 39,502 0.23%
vac01.ph.liv.ac.uk/vac-comp507.bullx 51,395 51,395 0.29%
vac01.ph.liv.ac.uk/vac-comp508.bullx 54,264 54,264 0.31%
vac01.ph.liv.ac.uk/vac-comp510.bullx 44,797 44,797 0.26%
vac01.ph.liv.ac.uk/vac-comp511.bullx 45,942 45,942 0.26%
vac01.ph.liv.ac.uk/vac-comp512.bullx 46,574 46,574 0.27%
vac01.ph.liv.ac.uk/vac-comp514.bullx 46,555 46,555 0.27%
vac01.ph.liv.ac.uk/vac-comp515.bullx 46,131 46,131 0.26%
vachammer.ph.liv.ac.uk/vac-r23-n11.ph.liv.ac.uk 90,398    90,398 0.52%
vachammer.ph.liv.ac.uk/vac-r23-n12.ph.liv.ac.uk 89,454    89,454 0.51%
vachammer.ph.liv.ac.uk/vac-r23-n13.ph.liv.ac.uk 89,847    89,847 0.51%
etc.
```

Result of query

Submit Host	HS06	No. of Jobs
Hepgrid2 (ARC)	5,572,132	125,328
Hepgrid5 (ARC)	7,642,634	117,573
All VAC	4,307,221	33,570
Total	17,521,987	276,471

Is it all there?

- We’ve got three “clusters”. ARC Condor on SL6, ARC Condor on CentOS7, and VAC (spread over two sites.)

Cluster Type	Slots	HS06	Max Delivery (Aug) in HS06 Hours (Work)
Condor SL6	792	7,670	5,706,480
Condor CentOS7	1,024	10,796	8,032,224
VAC	682 (some flaky)	8,168	6,076,992
Total	2,498	26,634	19,815,696

Compare

- Max theoretical is 19,815,696 HS06 Hours. Portal says 17,521,987. So we got 88.42% of max, in August. 11.6% gone.
- Some of the VAC nodes are off/flaky; about 3.2% of total HS06. So gap is 8.6%.
- There is (est.) a 2% to 4% overhead for draining multicore due to job stream stoppystartyness. So let’s call the gap 5%.
- VAC on CentOS7 is flaky (kernel lockups), which caused some loss. Combined with “routine problems”, job droughts and general maintenance, that could easily soak up the rest of the gap.
- Thus it’s ***very reasonable to assume*** that “practically all” the August accounting got to the portal.

Still nervous

- If I am nervous, I can use the other method to check my accounting (most of this was done about a year ago and tested, but not used since AFAIK)
  - https://indico.cern.ch/event/673843
- So I got it all out and did some off the cuff testing over the same period of August to see if it still works.
  - git clone https://github.com/gridpp/audit
- And I found a horrible bug (double multiplying the cores!) After fixing that, this is what I got.

Test results (Aug)

Submit Host	Work (Portal)	Job count (Portal)	Work (Audit)	Job count (Audit)
Condor SL6 (hepgrid2)	5,572,132	125,328	5,615,327	126,328
Condor CentOS7 (hepgrid5)	7,642,634	117,573	7,474,923	115,360
VAC	4,307,221	33,570	4,260,292	32,997
Total	**17,521,987**	**276,471**	**17,350,542**	**274,685**

Test results

- The hepgrid5 (the CentOS7 cluster) audit showed -2% fewer records than the Portal holds. For me, that’s a good result: records from heaven!
- The hepgrid2 (the SL6 cluster) audit was closer, showing 0.7% more records than the Portal.
- The VAC Audit showed -1% fewer records than the Portal.
- I suspect I may have been careless preserving job log files. Or FS problems … or more bugs, boundary conditions, assumptions.
- The results are off further than I am comfortable about and more work in a “clean room” environment needed to narrow it down. In general, it’s just about close enough. The August “books” look OK.

condor_history

- The closer you can get to the original data, the better.
- The methods discussed so far use the ARC urs records, which are a side-effect of the Jura process. And not all sites use Jura.
- An alternative to Jura is to create accounting records directly from (say) HTCondor logged data.
- This would work, we assume, using any type of CE (CREAM, ARC, HTCondor-CE, whatever…)
- We’d just need a tool to create accounting (SSM) records from the condor_history files.
-

htcondorToApel.py

- Hence htcondorToApel.py
- It works from the /var/lib/condor/spool/history file.
- It goes through the condor history files, finding data on completed jobs, applying scaling factors according to the scheme in Ref 2.
- This is strictly work in progress. I have not used this to produce or verify a full month of data yet. But once it is fully tested, this will keep a full log of all accounting transactions and create a monthly report that can be directly compared to the APEL Accounting portal.

Last minute Audit July

- I’m suspicious about results for August. Something went wrong …. 2% errors?!?
- So perhaps the logs for July were better? Results below for only CEs – spot on!

Submit Host	Work (Portal)	Job count (Portal)	Work (Audit)	Job count (Audit)
Condor SL6 (hepgrid2)	5,392,464	134,495	53,924,63.7	134,495
Condor CentOS7 (hepgrid5)	6,758,537	116,958	6,758,536.5	116958

Last minute Audit July

- So what might have happened in August?
- According to my notes, lot’s of migration that month; new Puppet server; new Yum Repo server; new Kickstart builds for all worker nodes; tests for all the above;
- Suggest some carelessness with the way the logs were preserved…. impossible to know for sure.

Last minute Discussion

Here’s an entertaining question.

A job is said to “belong” to the month in which its end time falls.

Our batch system scales wall clock durations (transparently to outside world) to account for different node strengths - we scale to a unitary “reference node type”. Wall clock durations on a faster node are increased by scaling factor to account for the extra work done, and decreased on a slower node. Hence the REAL job end time differs from the start time + scaled wallclock duration.

On a faster node, if the REAL end time is in the last moments of (say) September, and the start time + scaled wall clock falls in the first moments of October, to which month does the record belong? A similar question exists for slower nodes.