Estimating ARC/Condor Accounting

Steve Jones
University of Liverpool

- Sometimes it's useful to estimate the accounts for a system.
- When I see the values in APEL, I wonder if all my data got across. And I wonder whether all the data was properly scaled, multiplied up by cores, normalised, stored, whatever...
- Since we have several types of cluster (ARC/Condor, CREAM/Torque and VAC x 2), I can't work it out on the back of a fag packet.
- So I made some rough and ready tools to do it.
- This talk is about the tools I use to estimate the accounts APEL should have seen from ARC/Condor. I have similar things to Torque and VAC.

Preliminaries:

- May work on ARC with other batch systems.
- ARC writes job data to controldir (/var/spool/arc/jobstatus.) In here are ".diag" files, that hold the job durations etc.
- We scale those .diag durations to make them consistent with an abstract node type.

  Jura reads .diag files; creates UsageRecords (see /etc/arc.conf, jobreport_options). UsageRecords archived, sent to APEL (SSM).
- Thus the original logs are the .diag files, and the usage records show what Jura made out of them and sent over.

Getting the tools:

- You'll need to get some tools from my site.
- Point your browser to
- https://hep.ph.liv.ac.uk/~sjones/HepSysManNov16
- Get the two perl scripts there (accu.pl and parseUrs.pl)
- Install them on your ARC/Condor headnode in a directory called /root/audit/
- cd /root/audit; chmod 755 *pl

Using the tools:

- Be as root on ARC/Condor headnode (hg2)
- cd ~/audit
- ls /var/urs > allTheUrs # Loc varies, arc.conf
- for f in `cat allTheUrs`; do grep -l "EndTime.2016-10" /var/urs/$f; done > urs.oct
- for u in `cat urs.oct `; do
  ./parseUrs.pl $u;
  done > table.oct
- cat table.oct | ./accu.pl
- The HS06 Hours done in October should pop out.

Caveats, disclaimers:

- I've only tried to verify these tools at Liverpool site, so treat with caution.
- But they look portable to me.
- Let me know what you find and whether it tallies well with APEL.
- Let me know if it doesn't work, so I can fix it.
- And so on..

Results:

- When used with a similar set of tools for VAC and Torque (TBD), we compared the total results to the EGI APEL account portal for Sept and Oct. The Sept results were within ~ 0.4%.
- For October, the combined figure for VAC (local), VAC (remote), ARC/Condor and CREAM/Torque is 13,349,838 HS06 Hours. Figure in the EGI Portal (Normalised Elapsed time HEPSPEC06) is 13,390,992 HS06.
- The check scripts are within ~ 0.3%. The small discrepancy may down down to time zones, boundary/compare conditions, data that are "gone missing", or bad assumptions.

Further info:

- scaling_factors_plugin.py here:
  https://www.gridpp.ac.uk/wiki/
  Example_Build_of_an_ARC/Condor_Cluster)

- Also shows republishing script if you need to send data over again.

- Thanks, the end.