Pros and cons of various CE alternatives

Ste Jones
sjones@hep.ph.liv.ac.uk
Liverpool University

2 May 2019

Introduction

- I’ve only got 10 mins so I’ll have to whizz along. I’ll talk about three candidates to replace CREAM that I’ve tried out at Liverpool.
  - ARC (with a HTCondor batch system)
  - HTCondor-CE (with a HTCondor batch system)
  - VAC (with no batch system)
- ARC and HTCondor-CE are traditional CEs that front various standard batch systems.
- VAC is a novel approach that uses the VACUUM model that pulls work from VOs without the need for “headnodes”.
- All of these systems are viable, but any particular site may favour one over the others, depending on local circumstances.
- I’ll talk about some of the properties of the various choices; whether they are “pros” or “cons” depends on the application, so I’ll say a little bit about that, too.

Timeline

- 2010
  - Replaced lcg-CE + PBS with CREAM-CE + PBS
- 2014
  - Replaced CREAM-CE + PBS with ARC + HTCondor.
- ~ 2015/2016
  - Set up around 70 VAC nodes, some of which still exist.
- 2019
  - Replaced ARC + HTCondor with HTCondor-CE + HTCondor.
- Note: As a batch system, I’ve only used HTCondor, PBS and SGE. Not LSF, SLURM, LoadLeveller, BOINC ...

HTCondor-CE

- A traditional batch system gateway, N x 100K LOCs.
- Supports SCORE and ATLAS (8 slot) MCORE + more.
- Strong, demonstrated long-term support from a university-based team at Madison, Wisconsin.
- Relatively rare at European sites, common at OSG.
- Official documentation:

  https://bbockelm.github.io/docs/compute-element/htcondor-ce-overview/
- Organisational website:

  https://research.cs.wisc.edu/htcondor/
- GridPP example install (with HTCondor backend, C7):

  https://www.gridpp.ac.uk/wiki/Example_Build_of_an_HTCondor-CE_Cluster

HTCondor-CE

- APEL support:
  - At present, APEL client DB support, but only tested in combination with HTCondor batch system (see prev. talk). More work ongoing to RPMise, but viable solution already fully worked out, tested and available.

    https://twiki.cern.ch/twiki/bin/view/LCG/HTCondorAccounting
- ARGUS Integration:
  - Supported, see GridPP example build on slide 4.
-

HTCondor-CE

- BDII Integration:
  - We’ve only discovered GLUE2 provider (see slide 4) but GLUE1 is (reputedly) available .
- Data cache mechanism:
  - None that I know of.
- Batch systems:
  - HTCondor (native), PBS, LSF, SGE, SLURM
- CERN Puppet module: yes

ARC-CE

- A traditional batch system gateway, N x 100K LOCs.
- Supports SCORE and ATLAS (8 slot) MCORE + more.
- New version, 6, on the cusp.
- Strong, demonstrated long-term support from Nordugrid collaboration.
- Relatively rare at OSG, common in Europe.
- Official documentation:

  http://www.nordugrid.org/arc/ce/
- Organisational website:

  http://www.nordugrid.org/
- GridPP example install (with HTCondor backend, sl6):

  https://www.gridpp.ac.uk/wiki/Example_Build_of_an_ARC/Condor_Cluster

ARC-CE

- APEL support:
  - Via ARC Tool called JURA, for all ARC supported batch systems.

    www.nordugrid.org/documents/jura-tech-doc.pdf
- ARGUS Integration:
  - Supported, see GridPP example build on slide 6.
- BDII Integration:
  - Nominally full BDII support, GLUE1 and GLUE2.
-

ARC-CE

- Data cache mechanism:
  - ARC Cache for pre-placing files.
- Batch systems:
  - HTCondor , LoadLeveller, PBS, LSF, SGE, SLURM, BOINC
- CERN Puppet module: I don’t think so.

VAC

- A novel workload management system, that pulls work as VMs and requires no headnode, ~ 3K LOCs.
- Supports SCORE and various MCORE concepts.
- Responsive support, sole developer in GridPP/UK.
- Used at ~ 5 or 6 UK sites (see VAC evaluation, next.)
- Official and organisational website:

  https://www.gridpp.ac.uk/vac/

VAC

- APEL support:
  - Via supplied, dedicated python APEL client, makes use of standard SMM libs.
- ARGUS Integration:
  - Not needed, since VAC pulls work from secured sites.
- BDII Integration:
  - Not needed, since VAC works on “opportunistic” principles.
- Data cache mechanism:
  - None.
- Batch systems:
  - Not relevant. VAC needs no batch system. Each VAC factory is its own workernode.

Evaluations

- ARC and HTCondor-CE are are “generic CE systems”; no further evaluation is needed to explain their basics.
- But VAC is novel, so I used a questionnaire gauge opinion on functionality, installation/maintenance, testability/debugging and clarity.
- Findings at:

  http://hep.ph.liv.ac.uk/~sjones/vacEvaluation/vac.odt

- “Summary of summaries” follows:

VAC Evaluation

VAC is effective for running grid jobs for a couple of large LHC VOs and a number of smaller ones. It is much simpler than a CE & batch farm. It's low-maintenance. VAC clusters are usually full. It is easy to maintain and it scales well. VOs are satisfied. A Puppet module is available. The Vacpipe feature makes it trivial to support (say) GridPP assigned VOs. The whole VAC system comprises only ~ 3500 lines of Python. The developer of VAC, is highly responsive, but with all the risks on "just one developer". The documentation is usually up to date. VAC has decent logging.

Some issues keeping the ATLAS multicore full... I believe work is happening to fix this in the new harvester system. Some extra memory needed. Getting the iptables virtual network configuration correct for the VMs was tricky. Running jobs, in a VM, are somewhat opaque, although you can remote ssh into any of the running VMs to check for issues.

Which to use?

- General statements, to start with...
- ARC is the longest established; and is best known to the VOs. They know how to submit to this CE just as well as CREAM.
- Our HTCondor-CE is mostly utilised by ATLAS and LHCb and (very recently) GridPP “assigned VOs” that come from DIRAC.
- VAC is mostly used by the same set, i.e. not all VOs can use it.
-

Which to use?

- Whatever you chose to do, it might be right to try VAC on (e.g.) one node, since, if you get it going, you have at least some fallback position once CREAM goes (whatever else happens.) And it's trivial to scale up.
- If you find VAC great, stick with it (why not?)
- And if your batch system is not supported by either ARC or HTCondor, use VAC or change batch system.
- If not using all VAC, then let's talk about how to chose between ARC and HTCondor-CE.

Which to use?

- If your batch system is supported by only one of ARC or HTCondor, end of discussion...
- Assuming your batch system is supported by both ARC and HTCondor, then it’s game on …
- If your batch system is not HTCondor, then you would have to wait for APEL accounting in HTCondor-CE, since only HTCondor-CE/HTCondor has fully tested APEL support at present. If this is your position, then ARC might be the best choice if you want to start soon, since it already has APEL accounting (via JURA) for all its supported batch systems.

Which to use?

- If you get this far, your batch system is HTCondor, because HTCondor-CE presently has APEL support only for that (or you can write your own APEL code, or afford to wait for someone else to do so.)
- So you’ll have to explore some of the finer details to make the choice.
- If low-load and low use of file-space is your priority, then HTCondor-CE might be the better choice.
- If you’d like to use the ARC caching system to pre-place files, or some other ARC feature, then ARC is your choice.
- If you like a system where the same code is used for both the CE and the batch system (same dev, same processes), then HTCondor-CE is your choice.
- If you absolutely must have GLUE1, then only ARC definitely supports that.

Conclusions

- I’ve given links to official documents and example builds of 3 alternatives to CREAM-CE.
- For each, I’ve listed what I believe to be the salient features, in particular what batch systems are supported, how they hook up to external services and methods to install and maintain the products.
- Since VAC is novel, I’ve summarised the views of various site admins wrt usability as a CE replacement.
- And I’ve given criteria by which to chose between the various applications.