Science & Technology Facilities Council

UK Research and Innovation

Utilising Batch at the Tier 1

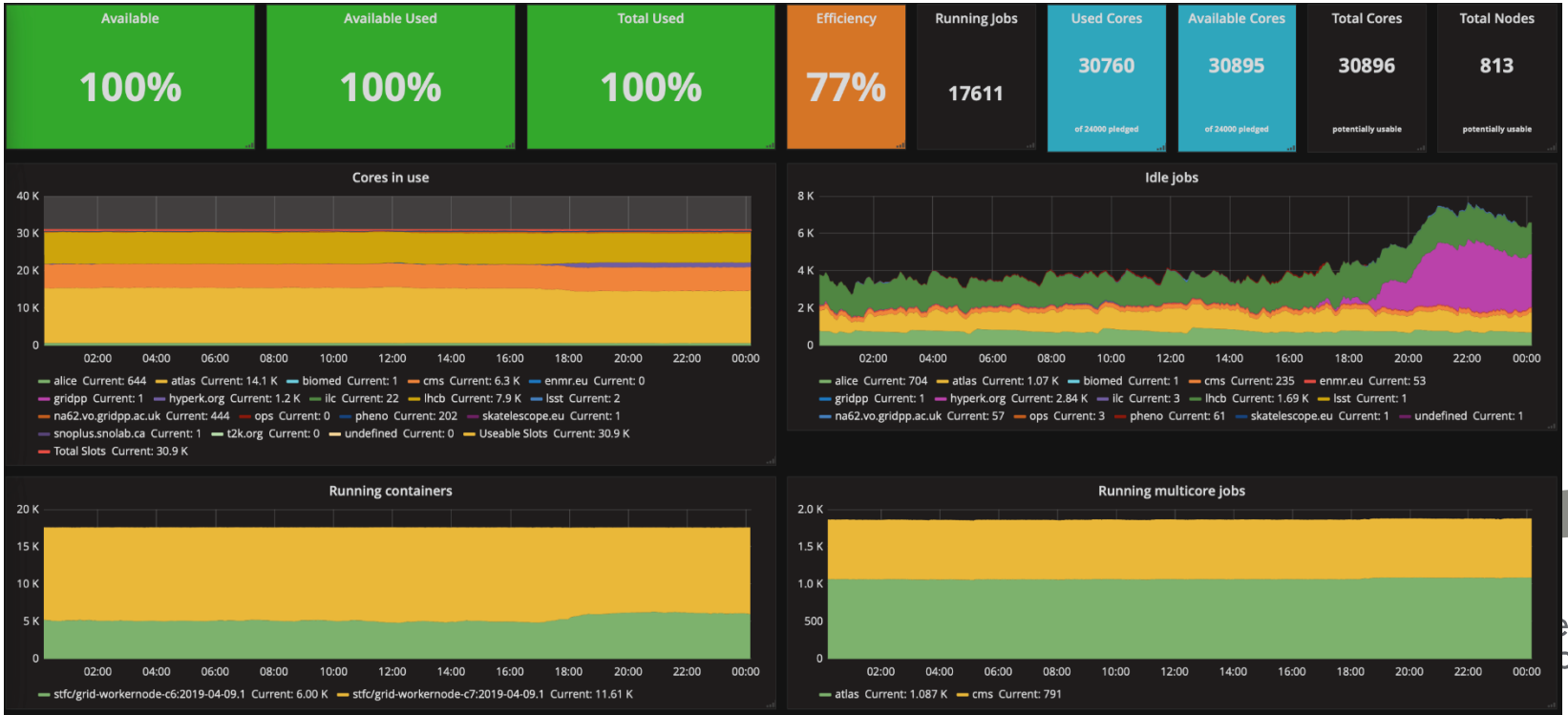
Alexander Dibbo

Summary

- Current Batch at the Tier 1
- The STFC Cloud
- Batch Bursting at the Tier 1
  - Approaches
  - Coyote
- Batch beyond the Tier 1

Current Batch at the Tier 1

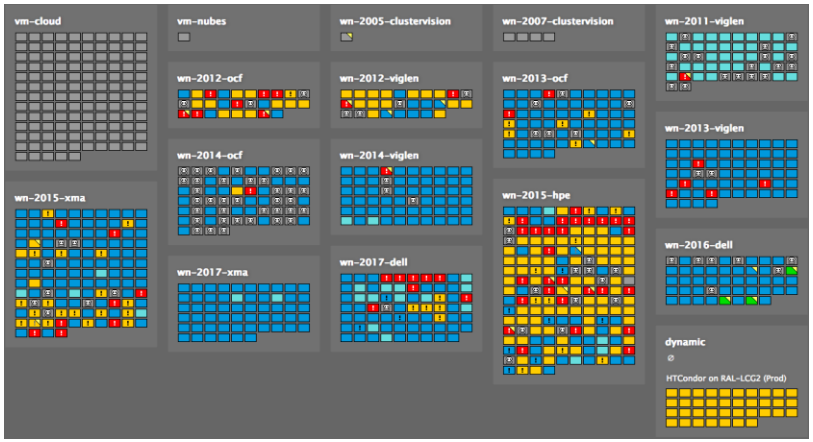
- 5 ARC CEs (version 5.4) + HTCondor (8.6.13)
- Jobs run in Docker containers
  - Support Centos 6 and 7 user spaces
  - Workernode software supplied by Matt D's tarball on CVMFS



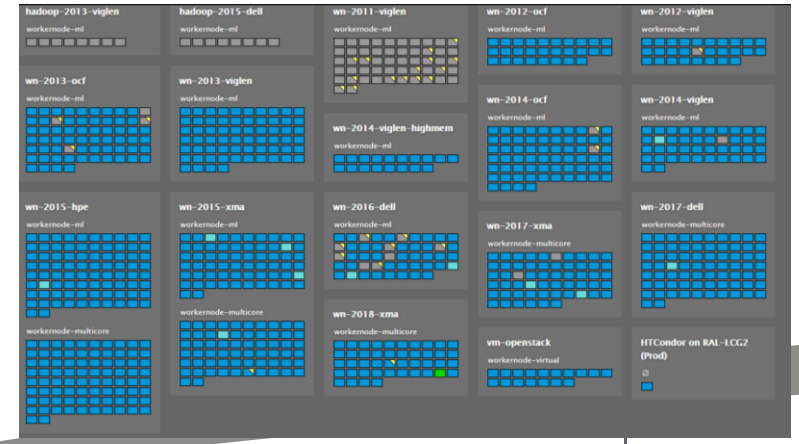
Recent improvements

- James Adams is batch farm manager and continues to fix problems and where sensible simplify setup.
- Since November 2018 we have had an apprentice (Dominic Banks) dedicated to fixing batch farm.
- 2019 pledge deployed into production in March!

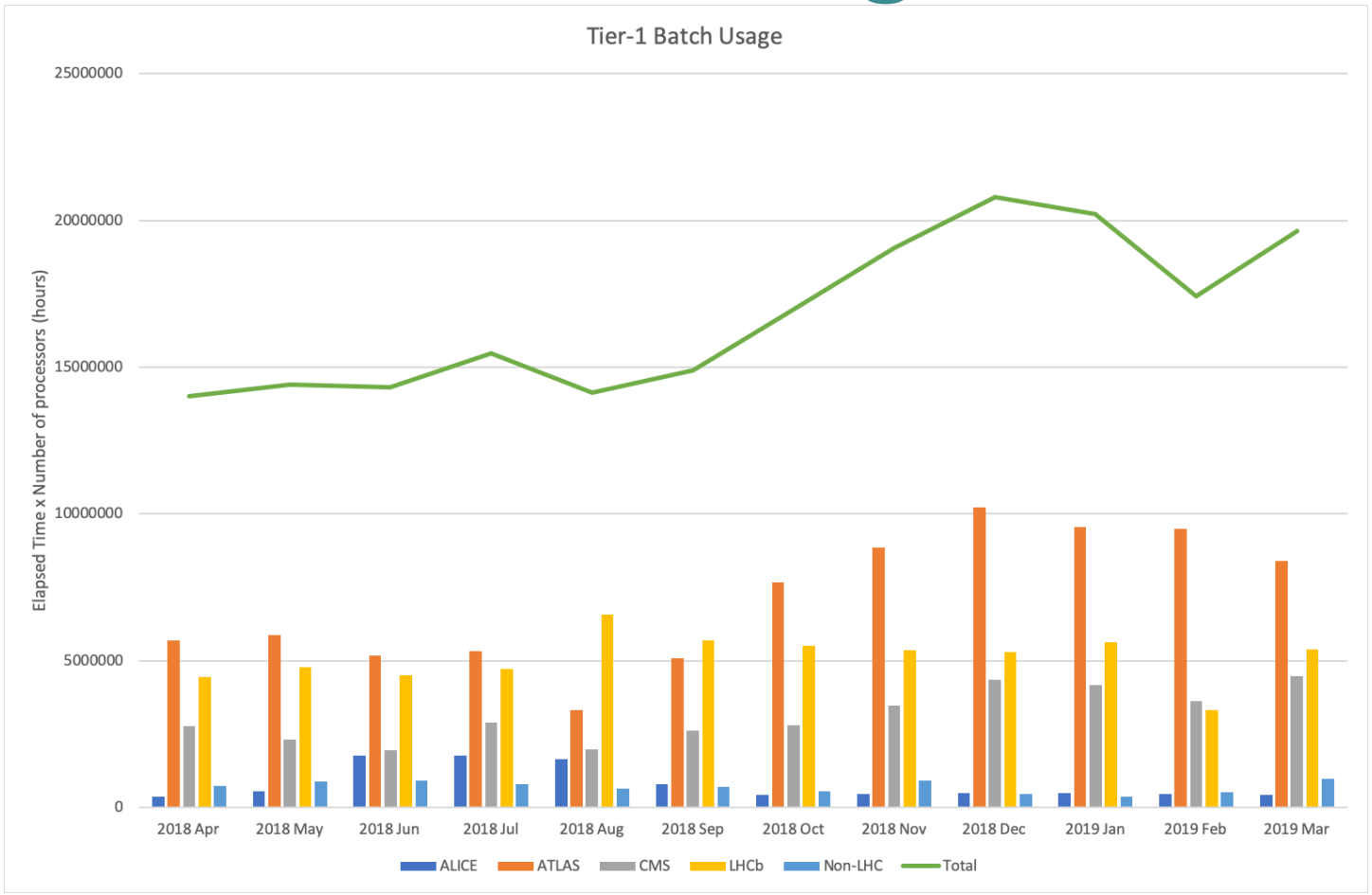
Aug 2018



Apr 2019



Batch Usage


Tier-1 Batch Usage
Elapsed Time x Number of processors (hours)
25000000
20000000
15000000
10000000
5000000
0
2018 Apr
2018 May
2018 Jun
2018 Jul
2018 Aug
2018 Sep
2018 Oct
2018 Nov
2018 Dec
2019 Jan
2019 Feb
2019 Mar
ALICE
ATLAS
CMS
LHCb
Non-LHC
Total

The STFC Cloud

- The Cloud is operated by a team separate from the Tier 1 since 2016
- The Cloud team has now moved to a different group within Scientific Computing
- The Cloud still works closely with the Tier 1
- The primary goal of the Cloud is to enable computing for STFC Science
  - STFC Facilities (ISIS, CLF and Diamond)
  - IRIS Communities (EUCLID, LSST, SKA etc)
  - STFC and SCD users

The STFC Cloud

- OpenStack based Infrastructure-as-a-service
- ~40 Controller nodes and currently 120 Hypervisors delivering over 3700 cores
  - 120 more hypervisors going in soon to bring this to ~8000 cores

Cloud Services

These are STFC Cloud capabilities intended to deployment this year which the Tier 1 will be able to make use of:

- Kubernetes-as-a-Service
  - powered by OpenStack Magnum
- Jupyter Hub service
  - (potentially with multiple backends)

Some users want Apache Spark or other “exotics”

- Limited support from us but if demand is significant the Cloud team may look at developing a recipe

Batch Bursting at the Tier 1

- Aim to backfill the STFC Cloud with worker nodes for the Tier-1
- Leave a buffer of capacity for use by Cloud users
  - The cloud prioritises on demand and interactive workloads and these need space to run.
- Could reinstall hypervisors as batch workers but this is intrusive and manual.
  - Having an automated process will mean that we don't have to spend time regularly balancing the capacity between cloud and batch

Batch Bursting at the Tier 1 - Approaches

- A cloud based HTCondor pool which the Tier 1 batch system submits a subset of jobs to.
  - Works best for remote/public cloud uses
- Get the Tier 1 HTCondor to create virtual workers based on the number of jobs waiting
  - Works well but Condor has no sense of how full the cloud is.
- Use OpenStack Heat to create a pool of workers that connect to the Tier 1 system.
  - Heat has no sense of how full the cloud is
- Use a custom script (coyote) to create and destroy workers based on the utilisation of the Cloud

Batch Bursting at the Tier 1 - Coyote

A python script which interacts with the OpenStack APIs and creates worker nodes and runs on a defined interval.

1. Delete virtual workers which are either shutoff or in error state
2. Query OpenStack for amount of available vCPUs
    1. Requires specific permissions for the coyote user
3. If available is greater than a buffer then create a worker
    1. Ensure capacity is available for Cloud users
4. Worker boots and connects to condor to start running jobs
5. After 1 week or no jobs condor stops and the worker shuts down.

Batch Bursting at the Tier 1 - Coyote

Next steps:

- Have coyote drain and stop workers when available is less than buffer
  - Choose workers which should drain fastest
- Have coyote kill workers when available is “much” less than the buffer
  - Hope is to that this will be an emergency action
- Dedicate some of the capacity to pre-emptible jobs to make quick reclamation of resources easier
- Publish the code to GitHub

Batch submission

3 options for batch for non LHC users:

1. ARC CE
   - We are working on HTCondor CEs
2. Local submission nodes (Coming soon)
   - Not designed to scale but for small user groups.
   - Not entirely clear on how accounting/tracebility works here
3. Build-your-own-batch
   - limited support from Tier 1/Cloud team (not recommended)

Any questions?