Commissioning of US HPC GPU resources for CMS Data Reconstruction

Dirk Hufnagel, Fermilab

CHEP2023

11th May 2023

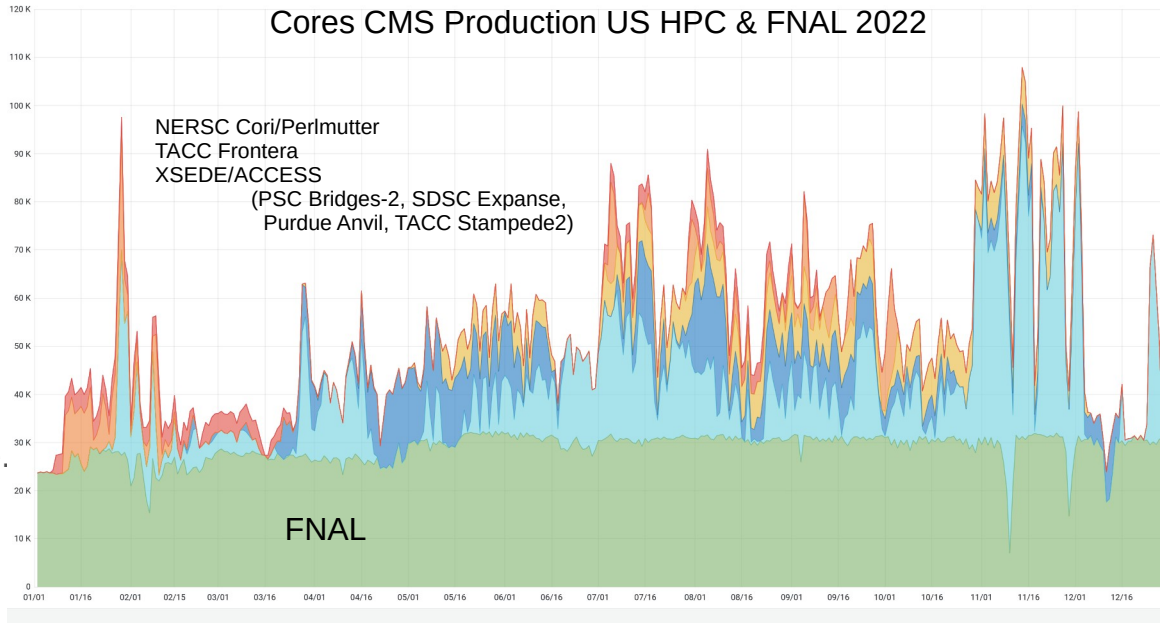
Overview

- Current US HPC usage and future outlook
- LCF integration
- GPU ReReco Workflows for US HPC
- Current Status and Outlook

Current CMS US HPC Usage

- Strong growth in US HPC use for CMS CompOps over last 5 years.

- Mostly used for various MC campaigns, with Pileup being read through the CMS AAA xrootd data federation.
- All these HPC can be categorized as ’user facility’ HPC and so far we are only targeting CPU production workflows at them.
- In this category (‘user facility’ US HPC suitable for CPU production workflows) there isn’t much room for growth anymore.



What's next?

- NERSC Perlmutter GPU (integration works, but we lack workflows)
- We aren't (regularly) using DOE Leadership Class Facilities (LCF)
  - Integration more difficult due to network restrictions on batch nodes
  - Provide most of their compute capacity via GPU (except ALCF Theta)
- The LCF are by far the largest US HPC
  - Pre-exascale: OLCF Summit, ALCF Polaris, ALCF Theta
  - Exascale: **OLCF Frontier**, ALCF Aurora (latter not yet available)
- Lots of potential for growth with LCF, but
  - Need to make integration work/usable
  - Need to be able to make use of GPU

LCF integration : overall approach and goals

- Use small ALCC (and discretionary) grants to work on integration
  - ALCC 2020/2021 (Joint US-ATLAS/US-CMS), ALCF Theta
    - Initial prototyping work, produced some MC samples for Snowmass studies
  - ALCC 2022/2023, OLCF Summit
    - Improve/generalize prototype, simplify operations, run/test GPU workflows
  - ALCC 2023/2024, ALCF Polaris/Aurora (applied, decision expected late May)

- Goals
  - Long term: able to utilize large LCF allocations for HL-LHC
  - How to get there: improve integration, larger allocation, iterate

LCF integration : software and conditions

- Both software (cvmfs via cvmfsexec) and conditions (frontier) access requires a local squid infrastructure (to act as a proxy for the batch nodes to bypass external network restrictions)
  - ALCC 2020/2021 (Joint US-ATLAS/US-CMS), ALCF Theta
    - Dedicated node provided by ANL, they managed the installation
  - ALCC 2022/2023, OLCF Summit
    - Deployed squid ourselves via OLCF Slate (OpenShift/Kubernetes)

- OLCF Slate setup intended to be template also for ALCF integration, ideally want to run the same Kubernetes setup (minimally the same container) to reduce support and maintenance costs long term
  - Currently no standardization of edge platforms, hopefully this will change

OLCF Slate Frontier-Squid

- Needed to get access to Slate Marble (Summit, Alpine)
  - Ticket with OLCF support to discuss needs like IO, file systems, network traffic etc

- Deployed directly from Github via Slate Marble web interface
  - OSG Frontier-Squid doesn’t work on OpenShift (no root allowed in container)
  - Modified version works
    - Bit hacky, good enough for now, long term needs some work (backport to OSG?)
    - https://github.com/hufnagel/images/tree/main/opensciencegrid/frontier-squid
  - Service with NodePort to squid port 3128 on pod
    - Allows CMS jobs on batch nodes to access squid pod through NodePort
  - CLI also available

Frontier-Squid Stress Testing

- LCF have batch queue policies that benefit/require large jobs
  - For example: larger wall times require many node batch jobs
- Optimally bad for squid, since many jobs start in parallel
  - Stress tests of central squid alone show problems with overloads/timeouts

- Solution: Extend squid hierarchy to the batch nodes
  - Run frontier-squid on each batch node, cascade-peer with central squid
  - Used successfully at Theta, Summit (and also NERSC, Frontera, Anvil)

LCF integration : job input data

- LCF batch nodes can’t access external network, rules out using AAA to stream data directory to jobs from storage at other CMS sites
- Running an edge xrootd proxy in principle possible, but edge platforms currently not suitable for that amount of traffic (might never be)
- Leaves reading job input data locally as only option, which requires active management of storage quotas at LCF and integration into the CMS transfer system (Rucio)
  - This currently means integration of Globus endpoints and transfers

Rucio/Globus for LCF data transfers

- Rucio supports Globus, allows transfers between Globus endpoints
- Transfer from storage that only supports gridftp or xrootd to storage that only supports Globus requires an intermediate that supports both

- Functionality tests within CMS Rucio integration instance
  - Rule 1 triggering FNAL – NERSC (gridftp) transfer
    & Rule 2 triggering NERSC – ALCF (Globus) transfer
  - Rule 3 triggering FNAL – NERSC – ALCF transfer (multihop)
- Adding Globus support to CMS Rucio production instance ongoing

- Globus access to LCF storage needs to be re-authenticated about once a week, concern for longterm/continuous data management operations

LCF integration : job output data

- Could stageout to local storage and transfer final output via Rucio
  - In the current WM system that would mean merge/cleanup/logcollect jobs would run at LCF, potentially a very bad fit with LCF batch policies

- Alternative: Any squid can be used for port-forwarding
  - We have our own squid already, small config change allows forwarding of xrootd traffic to FNAL dCache (even allows ACL restrictions)
  - Config change integrated in Github, active immediately after deployment
  - Requires wrapping xrootd commands to route tcp traffic through http proxy
    - https://github.com/rofl0r/proxychains-ng
  - Used at both ALCF Theta and OLCF Summit to stageout job output (both event data files and logs) to FNAL dCache

LCF integration : submission infrastructure

- Also need special integration into CMS submission infrastructure
  - GlideinWMS pilots (which we use at ‘user facility HPC’) don’t work due to network restrictions

- HTCondor Split-Starter setup connected to FNAL HEPCloud, similar to what’s used at Barcelona (J. Flix previous talk), except we don’t need to use communication through the shared file system (batch nodes can connect to login or edge nodes)

- Prototype developed/used with ALCC Theta grant, generalized and improved with ALCC Summit grant. Still a prototype, will need further improvements (with future ALCC grants or discretionary grants).

GPU Workflows targeting LCF (and NERSC GPU)

- Lots of efforts in last few years to port (reconstruction) algorithms to GPU
  - Already being used in HLT since 2022 (G.Parida Tuesday Track 2)
  - Moving towards production GPU ReReco workflow
    (C. Koraka Tuesday Track 2)
  - Also added support for GPU resources in CMS SI and WM
    (A. Pérez-Calero Yzquierdo Thursday Track 4)

- Caveat: unclear if that will really be the best/only use for GPU long term
  - Use of GPU for steps other than Reco (Gen for instance)?
  - Focus more on using GPU for ML/AI instead?

Current Status and Outlook

- NERSC mostly ready for more GPU ReReco workflows
  - Successfully ran preliminary GPU ReReco
    - Workflow assignment issues related to AAA in WM layers (have workaround, need final fix)
  - Waiting for workflows, current allocation ends Jan 2024, so have some time

- LCF still work in progress (but all needed functional pieces are there)
  - ALCC Summit grant expires end of June 2023
    - Just a few weeks ago ran first successful (CPU) test workflow
  - Will continue this work (discretionary grant on Summit and/or new ALCC grant for Polaris)
  - Also considering OLCF Frontier and ALCF Aurora, but support for non-Nvidia GPU depends on Alpaka adoption (A. Bocci Tuesday Track 2)

Related Contributions

- J. Flix: “Integration of the Barcelona Supercomputing Center for CMS computing: towards large scale production”, Track 7 Thursday 11:30
- G. Parida: “Run-3 Commissioning of CMS Online HLT reconstruction using GPUs”, Track X Tuesday 14:30
- C. Koraka: “Running GPU-enabled CMSSW workflows through the production system”, Track 2 Tuesday 14:00
- A. Pérez-Calero Yzquierdo: “The integration of heterogeneous resources in the CMS Submission Infrastructure for the LHC Run 3 and beyond”, Track 4 Thursday 12:15
- A. Bocci: “Adoption of the alpaka performance portability library in the CMS software”, Track 2 Tuesday 17:00