Available on CMS information server

CMS CR -2018/283

The Compact Muon Solenoid Experiment

Conference Report

Mailing address: CMS CERN, CH-1211 GENEVA 23, Switzerland

16 October 2018 (v3, 13 November 2018)

HPC resource integration into CMS Computing via HEPCloud

Dirk Hufnagel for the CMS Collaboration

Abstract

The higher energy and luminosity from the LHC in Run2 have put increased pressure on CMS computing resources. Extrapolating to even higher luminosities (and thus higher event complexities and trigger rates) beyond Run3, it becomes clear that simply scaling up the the current model of CMS computing alone will become economically unfeasible. High Performance Computing (HPC) facilities, widely used in scientific computing outside of HEP, have the potential to help fill the gap. Here we describe the USCMS efforts to integrate US HPC resources into CMS Computing via the HEPCloud project at Fermilab. We present advancements in our ability to use NERSC resources at scale and efforts to integrate other HPC sites as well. We present experience in the elastic use of HPC resources, quickly scaling up use when so required by CMS workflows. We also present performance studies of the CMS multi-threaded framework on both Haswell and KNL HPC resources.

Presented at *CHEP 2018 Computing in High-Energy Physics 2018*

HPC resource integration into CMS Computing via HEPCloud

Dirk Hufnagel[1,*], *Burt* Holzman[1], *David* Mason[1], *Parag* Mhashilkar[1], *Steven* Timm[1], *Anthony* Tiradani[1], *Farrukh Aftab* Khan[1], *Oliver* Gutsche[1], and *Kenneth* Bloom[2]

[1]Fermilab, Batavia, IL, USA
[2]University of Nebraska, Lincoln, NE, USA

Abstract. The higher energy and luminosity from the LHC in Run 2 have put increased pressure on CMS computing resources. Extrapolating to even higher luminosities (and thus higher event complexities and trigger rates) beyond Run 3, it becomes clear that simply scaling up the the current model of CMS computing alone will become economically unfeasible. High Performance Computing (HPC) facilities, widely used in scientific computing outside of HEP, have the potential to help fill the gap. Here we describe the U.S.CMS efforts to integrate US HPC resources into CMS Computing via the HEPCloud project at Fermilab. We present advancements in our ability to use NERSC resources at scale and efforts to integrate other HPC sites as well. We present experience in the elastic use of HPC resources, quickly scaling up use when so required by CMS workflows. We also present performance studies of the CMS multi-threaded framework on both Haswell and KNL HPC resources.

1 Introduction

The LHC experiments have their own computing infrastructures which have been successfully used during Run 1 and Run 2 of LHC. Why are we looking at using HPC sites to complement our own resources now? The reason is resource extrapolations for the planned LHC upgrade to the HL-LHC. Figure 1 shows the expected increase in trigger rates and luminosity for HL-LHC. These increases result in more CMS collission events collected and these events have a higher internal complexity, making them harder to reconstruct. Not only will we have more events to process, the processing time per event will also go up. All of this leads to a very large increase in resource demands for HL-LHC, as shown in figure 2.

2 What is HEPCloud?

HEPCloud is envisioned as a portal to an ecosystem of diverse computing resources, commercial or academic. It will provide "complete solutions" to users, with agreed-upon levels of service. It will route jobs to local or remote resources based on workflow requirements, cost, and efficiency of accessing various resources. U.S.CMS is planning to use HEPCloud to provide access for CMS to US HPC. Figure 3 shows a diagram with the architecture of the system.

*e-mail: hufnagel@fnal.gov

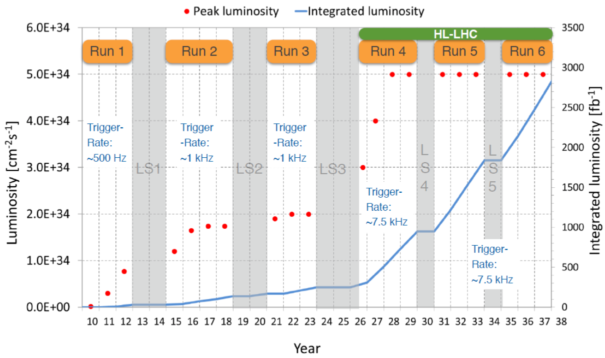

Figure 1. Trigger rates and Luminosity for HL-LHC

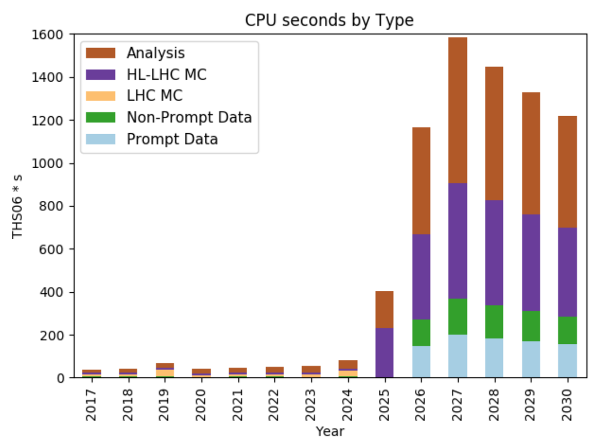

Figure 2. Extrapolating needed CPU for HL-LHC

At the moment the HEPCloud prototype provides access to various HPC and commercial cloud resources, but in a targeted way. That means that the jobs have to be explicitly targeted for the desired resource, without HEPCloud itself making any decision to route the job. Work on the Decision Engine, which will implement intelligent routing, continues and CMS plans to use it sometimes in 2019. For the work in 2018 CMS took advantage of the resource provisioning aspect of HEPCloud, but the decision to which resources to send job to was fully done on the CMS side (and HEPCloud provisioned resources based on this CMS decision).

3 HPC connected through HEPCloud

CMS can currently provision resources at three different HPC centers through HEPCloud. First are the Cori and Edison clusters at NERSC (National Energy Research Scientific Computing Center). NERSC is a HPC user facility operated by Lawrence Berkeley National Laboratory for the United States Department of Energy (DOE) Office of Science. In addition we can also provision resources at the Bridges cluster at PSC (Pittsburgh Supercomputing Center) and the Stampede2 cluster at TACC (Texas Advanced Computing Center). Both PSC

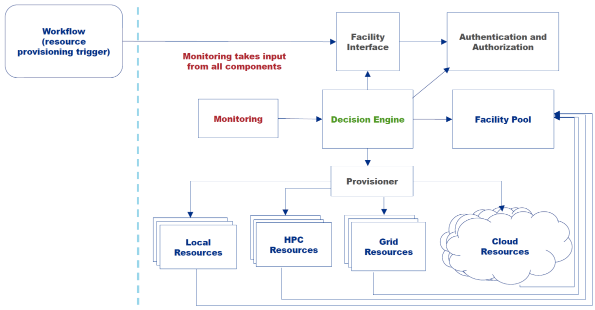

Figure 3. HEPCloud Architecture

and TACC are funded through the United Stated National Science Foundation (NSF) XSEDE (eXtreme Science and Engineering Discovery Environment) project [2].

4 HEPCloud integration into CMS Workflow Management

The CMS Workflow Management System is based on HTCondor, with schedds at both Fermilab and CERN that submit to a Global Pool of resources [1]. The HEPCloud resources are not integrated into the Global Pool, but remain in their own HEPCloud Pool. We allow CMS jobs to access HEPCloud resources via flocking, i.e. the Fermilab CMS schedds are members of both the CMS Global Pool and the HEPCloud Pool. Almost all kinds of job (with very few exceptions) are allowed to run at NERSC. Job input that isn't available locally will be read remotely from another CMS site [3].

5 NERSC Cori and Edison

CMS has been trying to use NERSC based HPC clusters for years, starting in 2014 with the now retired Carver cluster. In 2016 we switched our efforts to the Edison and Cori clusters and in 2017 we started provisioning resources on Cori and Edison through HEPCloud. NERSC is integrated into the HEPCloud resource provisioning via GlideInWMS [4], HTCondor [5] [6] and Bosco [7]. Bosco allows the HEPCloud pilot factory to submit pilot jobs to both Cori and Edison via a remote ssh tunnel. More on this and also on the NERSC runtime environment for CMS jobs can be found here [8].

5.1 Edison and Cori Hardware

Edison is a Cray XC30 supercomputer with worker nodes containing dual 12-core Intel "Ivy Bridge" processors at 2.4GHz. Due to hyper-threading we have 48 logical CPU cores available. Each node has 64GB memory. Cori is a Cray XC40 supercomputer with both Intel Haswell and Intel KNL worker nodes. The Haswell worker nodes contain dual 16-core Intel® Xeon^{TM} Processor E5-2698 v3 ("Haswell") at 2.3GHz. The KNL worker nodes contain a single 68-core Intel® Xeon Phi^{TM} Processor 7250 ("Knights Landing") at 1.4GHz. The Haswell nodes have 128GB of memory and with hyper-threading provide 64 logical CPU cores. The KNL nodes have 96GB of memory and with hyper-threading provide 272 logical CPU cores.

5.2 Edison and Cori Usage

After initially targeting both Edison and Cori and all possible batch queues (both shared nodes and full nodes), we switched in 2018 to only use Cori Haswell and KNL full node batch queues. Over the last two years Cori has received many upgrades to its networking, which benefited CMS workflows greatly [9]. We are using remote reads from Fermilab and other CMS sites in our workflows and such remote reads now happen at peak rates of order 100MB/s, where in the beginning we only saw a few MB/s or even less. These upgrades were never applied to Edison and as such the performance and failure rates differences now are such that a mixed use of Edison and Cori does not make sense anymore. We also disabled provisioning resources on the Cori Haswell shared node batch queues (where we could get access to a subset of a node). In principle we would have liked to keep using these resources since they are usually quicker to provision (although at lower overall scales). Mixing the shared and full node batch queues caused problems with job pressure calculations in the HEPCloud provisioning system which prevented us from scaling to higher number of cores overall. The same effect might force us to stop provisioning a mix of Cori Haswell and KNL resources, instead choosing only one of them as we attempt to push scale even higher. Alternatively, the deployment of a fully featured Decision Engine is expected to fix the problems with the job pressure calculations and would allow us to target all the Cori Haswell and KNL batch queues again.

5.3 Cori Haswell and KNL Performance

In order to estimate expected performance for both Haswell and KNL nodes (and especially performance differences between them), we ran extensive benchmarks using a standard ttbar event simulation chain (generation, simulation, digitization and reconstruction) at nodes available at Fermilab. The Fermilab nodes didn't have exactly the same node configuration as the Cori worker nodes, but were close (same CPU architecture for both Haswell and KNL and not memory limited). To estimate Cori Haswell performance we scaled node event throughput numbers measured on a dual socket Xeon E5-2698 system with 128GB memory by the 64/48 core count ratio. To estimate Cori KNL performance we scaled throughput numbers measured on a single socket Xeon Phi 7250 system with 96GB memory by the 1.4GHz/1.3GHz clock ratio. The results show a factor 3.3 event throughput per node advantage for the Cori Haswell nodes compared to the Cori KNL nodes. These numbers have since been validated with real workflows, although the exact differences are workflow dependent. Taking the different core hour charge factors for Haswell and KNL nodes into account, it costs about 4 times as many allocation core hours to produce the same CMS MC events on Cori KNL than on Cori Haswell. The obvious conclusion of this, to only use Cori Haswell nodes for CMS jobs, is not a practical approach due to both NERSC policy (allocations are for shared Haswell/Cori use, but a large imbalance towards Haswell is highly discouraged) and resource availability (it is much easier to get compute cycles on KNL nodes). In fact, over any extended period of time we always use many more hours on KNL nodes than Haswell nodes. Table 1 shows a list of CPU time spent in CMS jobs on both Cori Haswell and KNL full node batch queues for a few selected time periods.

5.4 Cori Haswell and KNL Scale of Operations

As of mid-October 2018, CMS has used about 26M core hours (allocation charge, not actual hours) at NERSC. A small fraction of these were used on Edison or Cori Haswell shared batch queues, but the vast majority were used on Cori Haswell and KNL full node batch queues.

Table 1. Aggregate CPU time for Cori KNL and Haswell in CMS jobs

Time Period (2018)	Cori resource type	CPU time (in seconds)
January to August	KNL	16.3M
January to August	Haswell	1.4M
September	KNL	4.1M
September	Haswell	1M
October week 1&2	KNL	6M
October week 1&2	Haswell	0.8M

Different from 2017 we did not run any multi-node jobs at NERSC this year. The reason for this was that the batch queue policies were relaxed to allow a high enough number of parallel jobs. In October 2018 we also hit some of our scaling goals for the year, reaching 30k cores in use with very good job efficiency and low failure rates. Figure 4 shows the number of cores utilized by CMS jobs running at NERSC during late September and early October 2018.

Figure 4. Cores in running CMS jobs at NERSC

6 XSEDE PSC Bridges and TACC Stampede2

XSEDE resources are integrated into HEPCloud differently than NERSC. We access them through a Hosted-CE provided by OSG, details of this setup can be found here [10].

6.1 PSC Bridges

HEPCloud can provision resources at the PSC Bridges cluster. We have access to the Regular Shared Memory (RSM) node batch queue. The RSM nodes contain dual 14-core Intel Haswell (E5-2695 v3) CPU and have 128GB of memory. Hyper-threading is disabled, so there are 28 logical CPU cores available, same as physical CPU cores. PSC provided CVMFS for CMS and also supports the singularity container system. This makes Bridges nodes usable the same way as CMS grid sites. Basic commissioning of Bridges for CMS workflows only took a few months, stable operations and full integration into normal CMS computing operations was achieved within 6 months. Figure 5 shows the number of cores utilized by CMS jobs running at PSC during late September and early October 2018. Scale is limited by the size of our allocation on Bridges. The sporadic nature of utilization is due to the cycle of pilots being submitted, pilots waiting for execution in the Bridges batch queue, pilots running and terminating followed by new pilots being submitted and so on.

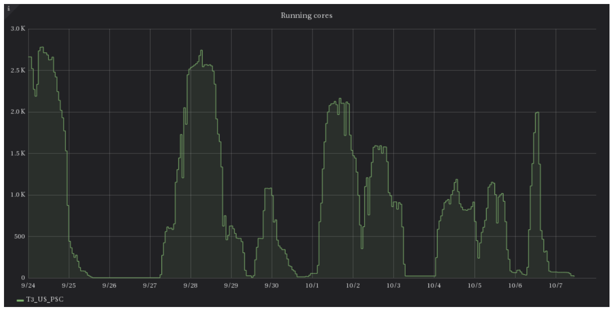

Figure 5. Cores in running CMS jobs at PSC

6.2 TACC Stampede2

HEPCloud can provision resources at the TACC Stampede2 cluster. Stampede2 is a mixed cluster with both Intel Xeon Platinum 8160 ("Skylake") and Intel Xeon Phi 7250 ("Knights Landing") worker nodes. At the moment HEPCloud is only accessing the former (not an inherent restriction, we wanted to keep things simple for now). TACC provided CVMFS on Stampede2 nodes and singularity is also available, so similarly to PSC Bridges the site is usable just like any other CMS grid site. Basic commissioning of Stampede2 for CMS workflows took a few months, but stable operations and full integration into normal CMS computing operations has only been achieved at very small scales. The main reason for this is that the Stampede2 batch policies only allow to run a very limited number of parallel jobs. Scale is only achievable with the use of multi-node batch jobs. Work will be needed in the Bosco layer to support multi-node jobs at Stampede2.

7 Summary and Outlook

Slowly HPC are becoming a normal part of the resource mix for CMS (or at least U.S.CMS). As such, the focus of our efforts will change somewhat from HPC internal efforts to making sure that such a large pool of compute resources can be seamlessly and efficiently integrated into the CMS Workflow Management systems. We also have to address questions like how the availability of large HPC allocations will affect planning of computing activities throughout the year. Managing a constant pool of hardware that is available whether it is used or not is different from managing a limited pool of hours that can be used only once. In parallel we will keep pursuing HPC integration efforts, particularly at TACC for Stampede2 (for multi-node jobs) and at DOE LCF facilities like Argonne Theta for instance.

References

[1] J. Balcas *et al.*, "Stability and Scalability of the CMS Global Pool: Pushing HTCondor and GlideinWMS to New Limits," J. Phys. Conf. Ser. **898**, no. 5, 052031 (2017). doi:10.1088/1742-6596/898/5/052031

[2] J. Towns, T. Cockerill, M. Dahan, I. Foster, K. Gaither, A. Grimshaw, V. Hazlewood, S. Lathrop, D. Lifka, G. D. Peterson, R. Roskies, J. R. Scott, N. Wilkins-Diehr, "XSEDE: Accelerating Scientific Discovery", Computing in Science & Engineering, vol.16, no. 5, pp. 62-74, Sept.-Oct. 2014, doi:10.1109/MCSE.2014.80

[3] K. Bloom *et al.*, "Any Data, Any Time, Anywhere: Global Data Access for Science," arXiv:1508.01443 [physics.comp-ph].

[4] I. Sfiligoi, D. C. Bradley, B. Holzman, P. Mhashilkar, S. Padhi and F. Wurthwrin, "The pilot way to Grid resources using glideinWMS," WRI World Congress **2**, 428 (2009). doi:10.1109/CSIE.2009.950

[5] HTCondor homepage: http://research.cs.wisc.edu/htcondor/

[6] M. Litzkow, M. Livny, and M. Mutka, "Condor – A Hunter of Idle Workstations", Proc. of the 8th Int. Conf. of Dist. Comp. Sys., June, 1988, pp 104-111.

[7] D. Weitzel, I. Sfiligoi, B. Bockelman, J. Frey, F. Wuerthwein, D. Fraser and D. Swanson, "Accessing opportunistic resources with Bosco," J. Phys. Conf. Ser. **513**, 032105 (2014). doi:10.1088/1742-6596/513/3/032105

[8] D. Hufnagel [CMS Collaboration], "CMS use of allocation based HPC resources," J. Phys. Conf. Ser. **898**, no. 9, 092050 (2017). doi:10.1088/1742-6596/898/9/092050

[9] R. Canon, T. Declerck, B. Draney, J. Lee, D. Paul, D. Skinner, "Enabling a SuperFacility with Software Defined Networking," CUG2017 Proceedings https://cug.org/proceedings/cug2017_proceedings/includes/files/pap165s2-file1.pdf

[10] S. Thapa, R. W. Gardner, Jr., D. Lesny, D. Hufnagel, K. Herner and M. Rynge, "Homogenizing OSG and XSEDE: Providing Access to XSEDE Allocations through OSG Infrastructure," FERMILAB-CONF-18-221-CD.

Acknowledgments

Fermilab is operated by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the United States Department of Energy.

This research used resources of the National Energy Research Scientific Computing Center, a DOE Office of Science User Facility supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-1548562.

This work used the Extreme Science and Engineering Discovery Environment (XSEDE) resource Bridges at PSC through allocation PHY170050.

This work used the Extreme Science and Engineering Discovery Environment (XSEDE) resource Stampede2 at TACC through allocation PHY170050.