The HEPCloud Facility: elastic computing for High Energy Physics – The NOvA Use Case

S Fuess[1], G Garzoglio[1], B Holzman[1], R Kennedy[1], A Norman[1], S Timm[1, †], and A Tiradani[1]

[1] Scientific Computing Division, Fermilab, Batavia, IL 60563, USA

[†] Corresponding author: timm@fnal.gov

Abstract. The need for computing in the HEP community follows cycles of peaks and valleys mainly driven by conference dates, accelerator shutdown, holiday schedules, and other factors. Because of this, the classical method of provisioning these resources at providing facilities has drawbacks such as potential overprovisioning. As the appetite for computing increases, however, so does the need to maximize cost efficiency by developing a model for dynamically provisioning resources only when needed.

To address this issue, the HEPCloud project was launched by the Fermilab Scientific Computing Division in June 2015. Its goal is to develop a facility that provides a common interface to a variety of resources, including local clusters, grids, high performance computers, and community and commercial Clouds. Initially targeted experiments include CMS and NOvA, as well as other Fermilab stakeholders.

In its first phase, the project has demonstrated the use of the "elastic" provisioning model offered by commercial clouds, such as Amazon Web Services. In this model, resources are rented and provisioned automatically over the Internet upon request. In January 2016, the project demonstrated the ability to increase the total amount of global CMS resources by 58,000 cores from 150,000 cores - a 25 percent increase - in preparation for the Recontres de Moriond. In March 2016, the NOvA experiment has also demonstrated resource burst capabilities with an additional 7,300 cores, achieving a scale almost four times as large as the local allocated resources and utilizing the local AWS s3 storage to optimize data handling operations and costs. NOvA was using the same familiar services used for local computations, such as data handling and job submission, in preparation for the Neutrino 2016 conference. In both cases, the cost was contained by the use of the Amazon Spot Instance Market and the Decision Engine, a HEPCloud component that aims at minimizing cost and job interruption.

This paper describes the Fermilab HEPCloud Facility and the challenges overcome for the CMS and NOvA communities.

1. Introduction

The computing needs of High Energy Physics are expected to grow by more than a factor of ten in the next decade. Experiments such as the High Luminosity LHC (HL-LHC) and DUNE, to mention a few, will push computing technology beyond their current limits. Preliminary analyses of the HL-LHC data storage needs for the first year, for example, estimate an increase of raw and derived data from 130 PB in 2016 to 1.5 EB in 2027, with a required 60-fold increase in CPU cycles [1]. The complexity of computing operations will be compounded by the fact that utilization will continue to follow the already familiar trends of

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

peaks and valleys of demand, due in part to the schedule of the experiments, with additional seemingly stochastic request bursts to satisfy the needs of data analysis.

The HEP computing facilities need to evolve to cope with these challenges. The report of the Particle Physics Project Prioritization Panel (P5) to the US funding agencies [2] underlines the importance of the "strategic partnership of national laboratories and universities with the industry" to overcome these challenges. In this regard, commercial cloud providers offer computing services that look promising for HEP in terms of capabilities and costs. National Laboratories with strong HEP programs, such as Brookhaven and Fermilab, have started projects to integrate commercial clouds with their facilities. Together with the integration of High Performance Computers (HPC), this strategy aims at satisfying the bursts of computing demands from the community.

At Fermilab, the evolution of the computing facility is carried through the HEPCloud project. The resulting HEPCloud facility is envisioned as a portal to an ecosystem of diverse computing resources, commercial and academic. The facility routes workflows to local or remote resources based on workflow requirements, cost, and efficiency of accessing the resources. It provides a complete solution to the users, managing costs at commercial providers and user allocations at HPC. The pilot project to explore the feasibility and capability of HEPCloud was started in fiscal year 2016. The pilot work was funded through seed money provided by the industry, initially by Amazon Web Services (AWS). The pilot executed workflows from the Compact Muon Solenoid (CMS) and the NOvA experiment. The CMS results are discussed elsewhere [3]. This paper focuses on the experience from the NOvA experiment.

2. The NOvA Use Case

The NOvA experiment has worked as a partner with the Fermilab Scientific Computing Division (SCD) to access cloud resources for their production computing since 2014. SCD and the NOvA collaboration applied and obtained a research grant from Amazon Web Services for $30,000. The grant funds were used to prototype and execute the demonstration of commercial clouds described in this paper. The funds in the grant were made available through a credited account with AWS. The account was enabled through a procurement process that selected DLT as the reseller of AWS services.

NOvA managed three separate computational campaigns on AWS through the Fermilab HEPCloud facility. Each campaign was systematically aimed at accomplishing three complementary goals. First, each campaign was designed to explore increasing the scale of resources that were provisioned compared to the previous campaign; second, each campaign would increase the stability of the system at the new scale and improve operational performance in both efficiency and operational support load; third, produce a specific useful physics result for the 2015/2016 NOvA oscillation analysis.

The third campaign was the official (final) reconstruction of neutrino events in the NOvA near detector. This sample was the baseline for performing the extrapolation of neutrino signals to the far detector and an essential component in the computation of the final systematic uncertainties on the oscillation signal. The workflow processed 56 TB of data spread over 57,000 files (average 1 GB/file), representing 114 million events in the near detector. The reconstruction time required for this stage of processing consumed approximately 4.5 hours per file, constituting a total of 260,000 core hours. This set of jobs produced 124 TB of output data.

3. Production Operations and Processes

The general workflow and data management that was used throughout the campaigns relied on a combination of the existing standard tools (used by NOvA computing and the production data processing group) and a set of extensions to those tools. These were specifically developed to address the integration of the standard data processing environment with the specialized environment of commercial clouds provisioned by HEPCloud.

In this model, the input datasets were specifically pre-staged to predetermined locations on the AWS S3 storage service using a combination of the standard SAM4Users tool suite [4] and modifications to the migration tools to allow for highly parallel staging of the data. The tool uses multiple nodes from an on-premises grid to push files to S3 from Fermilab storage. With each node having a 1 Gbps network interface, the peak aggregate upload bandwidth saturated at 12 Gbps with a few dozen nodes. This resulted in an average transfer/replication time of a few hours for the input datasets used in the campaigns.

Even though jobs were configured to run on three AWS regions to improve resource availability, the input dataset was transferred from Fermilab to only one of the AWS S3 regions[1] (Oregon). The jobs relied on the AWS internal network for data transfers. This incurred a data transfer fee that was less expensive than replicating the data two additional times.

Jobs were submitted with the jobsub tool to the FIFE batch submission system [5]. The jobs were routed to HEPCloud through an attribute that requested cloud resources explicitly. Each job processed multiple files one after the other (typically 5), asking SAM each time to transfer an unprocessed file from the dataset. If a job failed because it was preempted (e.g. the instance was overbid on the spot market), HEPCloud automatically resubmitted the job, which resumed processing the next file in the sequence maintained by SAM. A file that was not fully processed because of preemption or application failures, was not transferred again by SAM. After the jobs had all finished, therefore, operations typically submitted recovery jobs to process all such files in the dataset.

At the end of each campaign, the SAM4Users tool was used to transfer the files from S3 to the Fermilab mass storage system for long-term archiving, using multiple nodes to improve transfer rate as in the case of data uploading.

The production campaigns relied on several supporting services deployed at AWS and at Fermilab [3,6]. NOvA used the following services:

Service	Deployed at	Notes
Squid / CVMFS	AWS	Local cache for software distribution. We configured the service to automatically scale the number of instances depending on the load [7].
Network Configuration	AWS	Defines network access for instances running at AWS.
AWS Limits	AWS	Defines usage limits for AWS service such as the maximum number of VM, storage, etc.
Spot Market	AWS	Users bid on the excess VM capacity at AWS. Price is reduced several times, but VM may be preempted.
Accounting and Billing	FNAL	Resource accounting and monitoring; alarms on spending rate thresholds for intrusion detection.
GlideinWMS	FNAL	Workload management for FIFE and internal to HEPCloud.

[1] Even though S3 is a global service with a global namespace, AWS organizes S3 data in "buckets" that are stored at a given region

In addition, NOvA relied on the FIFE job submission service and SAM data handling, as discussed above.

4. Workflow Performance

The HEPCloud Facility was configured to provision five types of 8-core and 16-core instances[2] from AWS in eight Availability Zones covering three AWS Regions. We base our statistics on the second submission of the third campaign, because we consider it representative of standard operations. This submission consumed 203,000 core hours over a period of 48 hours, peaking at 7,300 concurrent jobs on 900 instances for 21 hours (Figure 1).

In general, the variety of instance types at AWS is key to enabling cost-effective large scale computing; NOvA was somewhat limited in their capacity to exploit the diversity of resources. NOvA could not run on any of the c3 and c4 instance types, which all have less than 2 GB of memory per core, because the application required 2 GB or more. In addition, jobs could not run on any 4-core instance types because of the configuration of the FIFE factory, which submitted requests to HEPCloud to provision slots only with 8-core and 15 GB of RAM. These parameters were difficult to change because the factory was part of ongoing FIFE production operations. On the other hand, the 4-core instances were the most popular for CMS and were key in enabling the scale of 58,000 cores during the CMS demonstrator of HEPCloud. In response, the design of HEPCloud is being reconsidered to enable more flexibility in the resource allocation.

The full production campaign processed a dataset of about 57,000 files (56 TB). This was organized in multiple job submissions that processed the same input dataset. The second submission consisted of 10,000 jobs, the maximum allowed by the FIFE batch submission system, processing a maximum of 5 input files per job. This job submission processed 46,858 files out of 57,000. With an average processing time per file of 5 hours, the expected duration of a job was about a day. The remaining files were processed by further recovery submissions, together with the files unprocessed due to application failure or preemption. Considering that AWS is a highly preemptive environment, this was considered an appropriate and acceptable operational practice.

To give a sense of scale on the amount of preemption that occurred during the run, Figure 2 shows the total number of virtual machines running and preempted ("overbid") every hour. In total, 1,035 virtual machines were preempted over a period of 2 days. This is a similar scale to the number of machines that were not preempted (blue histogram plateau in the figure). Overall from the perspective of the jobs, only 37% completed without being preempted and 38% of the jobs were resubmitted twice by the system. It should be noted that preempted jobs are automatically resubmitted by the system and do not necessarily result in a job failure. The file that was being processed at the time of preemption, however, was not presented by SAM again for processing and had to be recovered.

[2] The instance types were m3.2xlarge, m4.2xlarge, m4.4xlarge, r3.2xlarge, and r3.4xlarge. An additional 4-core instance, c3.xlarge, was used only for on-demand Squid servers and not for worker nodes.

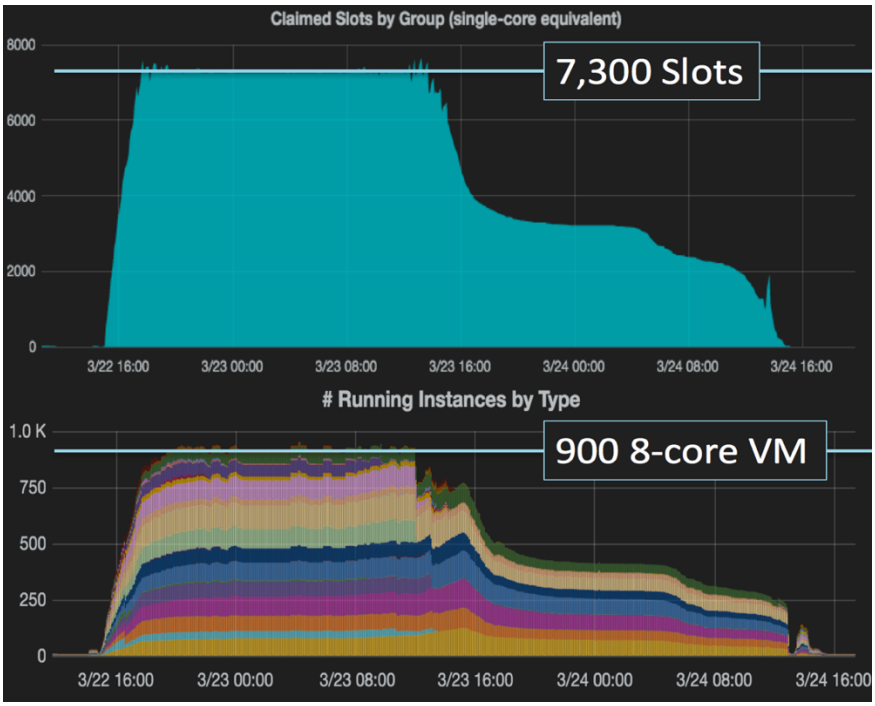

Figure 1: Total number of computing slots (top) and of virtual machines per instance types, availability zone, and region (bottom) for the third nova campaign

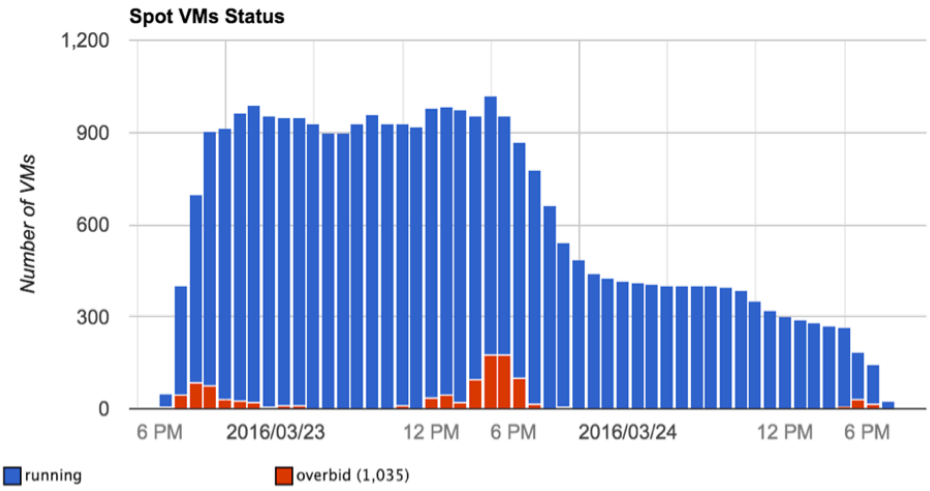

Figure 2: The total number of instances running and preempted ("overbid") every hour

Table 1 shows the efficiency of the workflow in terms of wall clock time and CPU time. Considering only successful jobs[3], the efficiency calculated as CPU time over wall clock time was 73% for this campaign submission, although for other campaigns it reached 96%. These are considered good efficiencies, considering that the workflow interacts with storage and external databases and, thus, is not CPU-bound. It should be noted that despite the large

[3] A limitation of the underlying infrastructure – we were tracking only CPU time for jobs which completed successfully.

difference in wall clock times between all the jobs and the final jobs, in our processing scheme preempted jobs can contribute to processing files, as opposed to the CMS use case, for example. It would be a mistake, therefore, considering that the inefficiency due to preemption, generally calculated as the ratio of the two numbers, ranges to up to ~50%.

	Time	Subsequent Recovery
Sum of all jobs wall clock (h)	202,706	60,059
Sum of successful jobs wall clock (h)	102,552	45,436
Sum of successful jobs CPU time (h)	74,958	43,589

Table 1: Wall clock and CPU time for all jobs and the successful jobs

5. Costs

The cost of commercial clouds has steadily decreased in recent years and provisioning burst capacity on the cloud is becoming ever more attractive. Cloud costs include prices for the computation as well as data storage and movement. In the AWS model, data ingress is free, but data egress is charged. For scientific institutions, however, AWS contains the cost by waiving the fee of data egress if the monthly data movement cost is 15% or less of the total cost [7]. In addition, for the NOvA campaigns, AWS waived all costs of data movement, even above the 15% threshold. All data movement costs mentioned in this section are estimates.

To minimize costs, the data egress waiver billing scheme produces an artificial operational incentive to produce data and transfer it back within the calendar month boundary. For example, in a single campaign, producing data one month and transferring it the next would not take advantage of the waiver.

The total cost of the second submission of the near detector reconstruction amounted to $6,160, broken down as

- $4,400 of EC2 costs, with $1,300 due to inter-region output transfers and the rest due to VM instance allocations
- $1,000 of S3 costs, with $700 due to inter-region input transfers and the rest to storage
- $530 of AWS support

As mentioned, this does not include any costs of data egress from S3 to Fermilab. The inter-region transfers, charged at $0.02 / GB are incurred by storing the data at one AWS region but accessing it and writing from three. This cost was estimated to be less than replicating the entire 57 TB input and 124 TB output datasets to all three regions for an estimated month of processing at ~$0.03 / GB per month.

When running at the scale of 7,300 slots, the cost rate was approximately $100 / h. For this submission, the total cost of failure was $120. The relatively small amount was mainly due to the fast failure mode for most jobs. The total cost of preempted jobs was $2,100, but it would be unfair to consider this a total loss since preempted jobs may contribute to processing files.

The estimated total cost for running the full reconstruction campaign is $7,900 for 260,000 core hours. The additional cost of data egress without the waiver is estimated at $9,400 for a total output dataset of 124 TB. The estimated ratio of the cost of data egress over the total cost is therefore about 54% for this workflow. The egress cost discounted by the 15% waiver would be $5,300. In general, workflows with a small data output as compared to their computation should be preferred as candidates for execution at AWS.

6. Cost Comparisons

The comparison of the costs to run at Fermilab and AWS are summarized in Table 2. The cost at Fermilab has been calculated including factors such as the amortization of the cost of the computing center building, power and cooling, cost of the hardware (computing, networking, etc.) and its lifetime, system administrators, etc. [3]

The cost for NOvA derives from the cost of support, computing, and storage, including inter-region transfers, but not the cost of data egress, as discussed above. The error is derived from the distribution of costs on a 6-hour basis.

Fermilab CMS Tier-1	\$0.009 ± 25% per core-hour
CMS at AWS	\$0.014 ± 12% per core-hour
NOvA at AWS	\$0.029 ± 14% per core-hour
NOvA cost at AWS per consumed file	\$0.20 per 2,000 events

Table 2: Cost comparisons between Fermilab and AWS for the NOvA and CMS campaigns

We did not perform a direct comparison of worker node performance running the same NOvA workflows at Fermilab and AWS. We have executed benchmarks on both systems and the performance is very similar. We have executed a simulation of a $t\bar{t}$ production for the CMS experiment to mimic the behavior of a Monte Carlo workflow. The systems at Fermilab produced on average 0.0163 events / s per core, while at AWS 0.0158 events / s per core; therefore, AWS was nearly equivalent. Benchmarks based on HEPSpec06 produced similarly comparable results. Given the similar performance, the cost comparison is based directly on the cost of core hours at Fermilab and at AWS.

As shown in the table, the cost of running at AWS was about three times larger than running locally at Fermilab. This premium is generally considered acceptable to achieve large burst capacity in case the additional slots could not be provisioned in any other way, e.g. on the Open Science Grid (OSG)..

Running the NOvA workflow at AWS was costlier than running the CMS one. This was due to the need to store more data in S3 and because we were unable to access a variety of less expensive instances. This was in part because of the larger memory footprint of the application and in part because the setup of the production FIFE submission system was more difficult to change, and 4-core instances could not be provisioned. These more cost-effective 4-core instances comprised 60% of the total number of instances in the CMS campaign.

7. Lessons Learned

On AWS, the large scale of affordable resources can be achieved only by a combination of instance types, availability zones and regions. To improve on the cost per core hour, there is an incentive in lowering the memory footprint of the workflow to gain access to more diverse and less expensive instance types. This is in general true also for the OSG environment, although the cost implications are less evident.

For NOvA, the current scale of slots is limited by the FIFE batch submission capacity. That is limited to 20,000 running slots for each of two schedulers. This limit affects the overall capacity of the system and it is particularly relevant when attempting resource bursts.

To reduce the risk of preemption, a good strategy is keeping the job short (a few hours). With SAM, a good way of achieving this is submitting many jobs (say 10,000, considering the limits above) for a given large dataset (say 50,000 files). This way, jobs process only a few files in average and finish quickly. In any case, if several jobs are preempted, the

remaining running ones can continue processing files until the dataset is completed, at which time all jobs exit. This simplifies bookkeeping and minimized the need for recovery.

To simplify operations with the NOvA (SAM) data processing model, the jobs stored data to S3 and declaring metadata and file location to SAM immediately. This is an improvement, as before files where declared to SAM only after they had been transferred to the Fermilab archive, introducing a significant latency in bookkeeping. This change was key to enabling rapid turnaround of job recovery, since at the end of the jobs, the status of the processed files was up to date in the database. In general, this experience has forced us to evaluate the costs of data egress. Workflows that cannot take full advantage of the waiver should be scrutinized as to their suitability for cloud computing.

When running jobs on the AWS spot instance, the efficiency based on the success rate of jobs tends to be lower than on local resources, due to the high occurrence of preemption. While this increases the overall computing cost for a fixed amount of work, the SAM processing model of consuming input files serially reduces the overall impact.

8. References

[1] I Bird 2016 WLCG Workshop Introduction, San Francisco, 8th October 2016

[2] Particle Physics Project Prioritization Panel 2014 Building for Discovery – Strategic Plan for U.S. Particle Physics in the Global Context

[3] L Bauerdick *et al*, 2016 HEPCloud, a new paradigm for HEP facilities: CMS Amazon Web Services Investigation, FERMILAB-PUB-16-170-CD.

[4] R A Illingworth 2014 A data handling system for modern and future Fermilab experiments, Journal of Physics: Conference Series, Volume 513, Track 3

[5] J Boyd *et al* 2016 Advances in Grid Computing for the FabrIc for Frontier Experiments Project at Fermilab, Proceedings of CHEP2016

[6] S Fuess *et al*, 2016 Virtual Machine Provisioning, Code Management and Data Movement Design for the Fermilab HEPCloud Facility, Proceedings of CHEP2016

[7] G Bernabeu *et al*, 2015 Cloud Services for Fermilab Stakeholders, *J. Phys.: Conf. Ser. 664 022039*.

Acknowledgments

Fermilab is operated by the Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the United States Department of Energy. This work is also supported by KISTI and Fermilab under the joint Cooperative Research and Development Agreement CRADA-FRA 2015-001 / KISTI-C15005.