Using NERSC High-Performance Computing (HPC) systems for high-energy nuclear physics applications with ALICE

Markus Fasel for the ALICE Collaboration

Lawrence Berkeley National Laboratory

E-mail: mfasel@lbl.gov

Abstract. High-Performance Computing Systems are powerful tools tailored to support large-scale applications that rely on low-latency inter-process communications to run efficiently. By design, these systems often impose constraints on application workflows, such as limited external network connectivity and whole node scheduling, that make more general-purpose computing tasks, such as those commonly found in high-energy nuclear physics applications, more difficult to carry out. In this work, we present a tool designed to simplify access to such complicated environments by handling the common tasks of job submission, software management, and local data management, in a framework that is easily adaptable to the specific requirements of various computing systems. The tool, initially constructed to process stand-alone ALICE simulations for detector and software development, was successfully deployed on the NERSC computing systems, Carver, Hopper and Edison, and is being configured to provide access to the next generation NERSC system, Cori. In this report, we describe the tool and discuss our experience running ALICE applications on NERSC HPC systems. The discussion will include our initial benchmarks of Cori compared to other systems and our attempts to leverage the new capabilities offered with Cori to support data-intensive applications, with a future goal of full integration of such systems into ALICE grid operations.

1. Introduction

The processing and analysis of event-based data from High Energy Nuclear physics (HENP) experiments is well suited for commodity computing hardware, configured to match the memory and I/O bandwidth requirements per CPU of the workload. The event structure of the data and processing tasks are embarrassingly parallel, whereby event sets can be processed in independent jobs run on separate CPUs, on different nodes, or even at different facilities. As a result, the experiments have come to rely on conventional compute clusters that range in capacity from a few hundred to many thousand CPUs, which can be distributed over a number of facilities without a general loss of capability.

Supercomputers or High-Performance Computing (HPC) systems are large-scale clusters that have been optimized for massively parallel processing tasks that require concurrent use of large numbers of compute cores to operate efficiently. As such, HPC systems are specialized hardware for targeted use cases that rely on low-latency communication between their processors. However, the scheduling of many large-scale jobs on a single HPC system can leave resource gaps that can be available for use by small-scale tasks, such as those in HENP applications. This

condition provides an opportunity for HENP experiments to make use of these resources, while, at the same time, increasing the overall utilization of those systems.

While the potential for opportunistic use of HPC resources can be significant, these systems are generally configured with restrictions on access and utilization not present on more conventional, commodity based clusters. These restrictions can include vendor-specific environments and workflow limitations that make these systems difficult to integrate into the computing infrastructures developed by the HENP experiments to leverage their distributed resources. We report here the results of some initial work done by the ALICE collaboration to make use of HPC systems operated at the US Department of Energy's National Energy Research Scientific Computing Center, NERSC. The goal of this work is to evaluate the efficacy of these HPC systems for our HENP applications, carried out within a framework developed to simplify an eventual integration into the larger distributed workflow of the experiment.

NERSC is the primary scientific computing center for the DOE Office of Science and the principal provider of high performance computing services to their science programs. NERSC currently operates two HPC systems, Edison and Cori, at its facility at Lawrence Berkeley National Laboratory (LBNL). Edison[1], commissioned in 2013, consists of over 130k CPU cores with peak performance of over 2.5 PF. Cori, named after the US biochemist Gerty Cori, is the newest NERSC HPC system and is being deployed in two phases. Cori Phase 1 [2], available since November 2015 and referred to as the "Data Phase", consists of over 1600 Intel Xeon Haswell™ nodes and has been designed with data intensive features such as a large memory per core configuration, a 28 PB Lustre scratch file system, and an 800 TB SSD-based Burst Buffer file system for workflows that demand high I/O capacities. The phase II [3] Cori system is expected to arrive in late 2016 and will include about 9000 nodes, each with a large number (60) of Intel Phi Knights Landing™ cores. Thus, Cori Phase II will be optimized for massive parallel computing and is therefore an excellent testbed for AliceO^2 [4], the next generation ALICE Software framework.

Along side the HPC systems, NERSC operates a large mid-range system of more conventional configuration, in which the HENP compute cluster, PDSF, exists. PDSF is an evergreen system that has been operated for over 15 years for the High Energy and Nuclear Physics communities at LBNL and consists of about 3000 compute cores with more that 3.0 PB of disk storage. PDSF is shared by several experiments and is part of the ALICE Grid as a WLCG ALICE Tier-2 center. The proximity of ALICE storage and grid resources on PDSF to the NERSC HPC systems makes NERSC an ideal site for evaluating use of HPC by ALICE computing operations.

2. The ANALISA tool running nuclear physics applications on HPC systems

HPC compute clusters often have features that limit their usability by distributed workflows developed for serial applications. Such features include whole-node scheduling, limited external network connection, time-based scheduling and historically small amounts of memory per CPU core. Several of these limitations require adaptions in the workflow, particularly during the integration of HPC systems into the ALICE computing grid.

In order to make job submission transparent to the user, we developed a tool called ANALISA, which takes into account the various features of all supercomputers currently supported. The tool consists of two parts, a submission layer and a worker layer. The submission layer runs on interactive (submit) hosts to manage job submission, preparation of the user software and the sandbox to contain the sub-job contents. The worker layer is executed on the compute nodes (or intermediate "mom" nodes depending on the cluster) and provides the MPI interface. It also handles the transfer of input files to the sandbox and output files to their final storage location. Fig. 2 shows the general workflow of our tool.

Steering of the job submission is done using a configuration file. The content can be expressed either in a key-value format, as an xml-tree or as a json string. The minimal content must provide

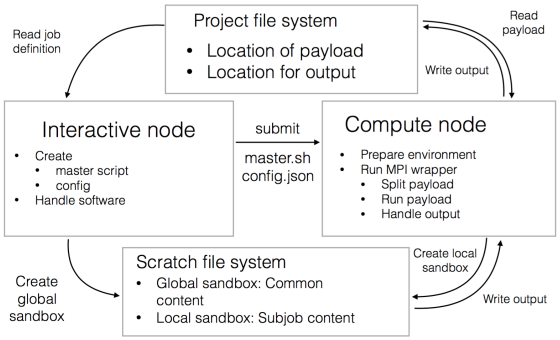

Figure 1. Workflow of ANALISA: The submitter prepares the job sandbox on the scratch file system, splits the job into n-master jobs and submits the master jobs to the batch queue. The worker provides the actual MPI interface. It copies the payload from the file system to the sandbox, runs the payload assigned to the different cores within the master job and copies the output back the output registered via the job definiton.

the set of information necessary to schedule the job, such as the amount of CPU cores, the time requested for each job, the name of the job and pointer to the executable. Other parameters like a set of input files and an output location are optional.

3. Experience with ALICE simulation jobs on NERSC supercomputers

ALICE simulation jobs consist of a variety of components: event generators (Pythia6/8 [5, 6], HIJING [7], DPMJet [8], others), collision systems (pp, p-Pb, Pb-Pb) center-of-mass energies, transport engines (GEANT3/4 [9, 10]), and software versions of ROOT and AliRoot. Therefore, the simulation tool must support a large number of configurations to be executed on the NERSC computing clusters. For our tests, a cocktail of different configurations was created as a test suite to cover a range of typical use cases. All collision systems are included in this test suite, while for pp collisions two different center-of-mass energies were used. The tests were limited to minimum bias collisions, acknowledging that biased simulations might skew to a higher multiplicity and consequently longer job execution times, not relevant to our current investigation. The settings of the test suite are listed in Tab. 1.

Tests were performed on the supercomputers Cori and Edison, as well as the local batch farm PDSF for direct comparison with a more conventional cluster. In all cases, the same versions of ROOT, AliRoot and GEANT3 were used. In the initial test, the software was obtained from a local build system component of the simulation tool on Cori and Edison but directly from CVMFS on PDSF. The ALICE conditions database, OCDB, was mirrored onto Edison and Cori scratch file systems, while direct access via CVMFS on PDSF was possible.

Collision system	Center-of-mass energy	Generator	Number of events / job
pp	7 TeV	Pythia6 Perugia2011	100
pp	8 TeV	Pythia8 Monash2013	100
p-Pb	5.02 TeV	DPMJET	100
PbPb	2.76 TeV	HIJNG	5

Table 1. Test setup used for our initial benchmarks: Jobs differ in the collision system, the center-of-mass energy and the event generator including tune.

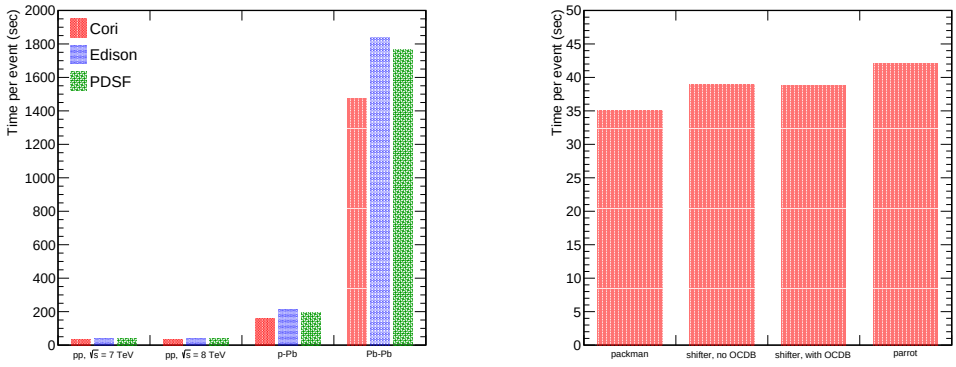

Figure 2. Left: Time per event spent for simulation jobs of the four different scenarios on the three platforms Cori, Edison and PDSF. The time takes into account event generation, particle transport, detector simulation and reconstruction. Right: Time per event for the test case pp, $\sqrt{s} = 7$ TeV using different methods for the distribution of the software stack and the OCDB.

The left panel in Fig. 2 shows the mean time spent per event from running our test cocktail on the different platforms. In all test cases, Cori performed the fastest of the three clusters. As mentioned above, PDSF is an evergreen cluster composed of several generations of CPUs. When we restrict the tests to a subset of machines on PDSF with the same CPU type as used in Cori, the same performance can be reached.

4. Software distribution on HPC systems

One critical task for distributing work, both onto a cluster and across many resources, is the consistent distribution of experiment software to the actual compute nodes. For the test already described, locally pre-compiled packages on each cluster's scratch system were used for software distribution. This option is optimal for development work in which strict consistency with officially released software is not required. In production, however, use of software that is centrally managed by the experiment is required.

The LHC experiments have adopted CVMFS [11] as their software distribution tool, on which software is built centrally at CERN and is efficiently migrated out to intermediate caches that the compute systems dynamically access. Normal use of CVMFS requires a kernel modification and a disk cache local to each node, neither of which generally exists on HPC systems. Our additional tests focus on making software distributed by CVMFS directly available on Cori.

NERSC has worked with Cray to develop a tool called Shifter [12] based on linux container technology [13] in which the experiments can customize some aspects of the operating environment. While use of Shifter does not allow CVMFS to run natively, it does give the experiments some additional options, two of which were investigated. In one option, the full ALICE repository is unpacked directly into a Shifter image and accessed during runtime as a normal file system. The second option used a preload procedure to deploy the ALICE repository onto a filesystem, mounted as a repository with a tool, parrot [14], installed into the Shifter image.

The tests were run on Cori using the pp-scenario at 7 TeV listed in Tab. 1. In the case where CVMFS is included in the image, the test is split into two parts: one with just the software stack taken from the image and the other with both the ALICE software and OCDB included in image. The results of the tests are shown in the right panel of Fig. 2 and compared with the locally built test (labeled packman) as a measure of the optimal performance which can be

reached on Cori. All options for CVMFS distribution were successfully commissioned; however, the performance found with the locally-built software stack has not yet been achieved, suggesting further optimizations of the software distribution via CVMFS onto NERSC supercomputers are needed.

5. Integration of the NERSC supercomputer Cori into the ALICE Computing Grid

A future goal of this work is the integration of Cori into the ALICE Grid facility [15]]. Accessing the distributed software through the CVMFS system is just a first step. For example, the test jobs presented here take their payload from the local file system while, for grid jobs, it is assigned dynamically from the remote AliEn task queue. Thus, additional integration will need to cope with scheduling differences between normal Grid and HPC systems. To simplify that work, Cori is initially intended only for simulations.

6. Conclusions

To summarize, we have successfully commissioned a tool for running serial ALICE jobs on HPC systems designed for parallel processing. We have tested the tool with a suite of ALICE simulation jobs in which the payload is the same as would be run centrally from the ALICE Grid and found that the single job efficiency was similar to that obtained on the more conventional PDSF cluster. We have tested and verified two methods to deliver the centrally managed ALICE software stack onto NERSC HPC system, which represents significant initial steps towards the integration of NERSC supercomputers into the ALICE computing grid.

7. Acknowledgements

This work is supported by the Office of Nuclear Physics within the U.S. DOE Office of Science under contact number DE-AC03-76SF00098. We wish to thank the NERSC Center at LBNL and specifically L. Gerhardt of NERSC for her help with the CVMFS installation into Shifter.

References

[1] Online, accessed Feb 21, 2016 URL `http://www.nersc.gov/users/computational-systems/edison/configuration/`
[2] Online, accessed Feb 21, 2016 URL `http://www.nersc.gov/users/computational-systems/cori/cori-phase-i/`
[3] Online, accessed Feb 21, 2016 URL `http://www.nersc.gov/users/computational-systems/cori/cori-phase-ii/`
[4] Buncic P, Krzewicki M and Vande Vyvre P 2015 Technical Design Report for the Upgrade of the Online-Offline Computing System Tech. Rep. CERN-LHCC-2015-006. ALICE-TDR-019 CERN Geneva URL `http://cds.cern.ch/record/2011297`
[5] Sjöstrand T, Mrenna S and Skands P 2006 *JHEP* **05** 026 (*Preprint* `hep-ph/0603175`)
[6] Sjöstrand T, Mrenna S and Skands P 2007 *Comput. Phys. Commun.* **178** 852–867. 27 p URL `https://cds.cern.ch/record/1064095`
[7] Wang X N and Gyulassy M 1991 *Phys. Rev.* **D44** 3501–3516
[8] Roesler S, Engel R and Ranft J 2000 *Advanced Monte Carlo for radiation physics, particle transport simulation and applications. Proceedings, Conference, MC2000, Lisbon, Portugal, October 23-26, 2000* pp 1033–1038 (*Preprint* `hep-ph/0012252`) URL `http://www-public.slac.stanford.edu/sciDoc/docMeta.aspx?slacPubNumber=SLAC-PUB-8740`
[9] Brun R, Hagelberg R, Hansroul M and Lassalle J C 1978 *Simulation program for particle physics experiments, GEANT: user guide and reference manual* (Geneva: CERN) URL `https://cds.cern.ch/record/118715`
[10] Agostinelli S *et al.* (GEANT4) 2003 *Nucl. Instrum. Meth.* **A506** 250–303
[11] Blomer J, Aguado Sanchez C, Buncic P and Harutyunyan A 2011 *J. Phys. Conf. Ser.* **331** 042003
[12] Botts J, Jacobsen D and Gerhardt L talk at the Workshop HEPIX Fall 2015 URL `https://indico.cern.ch/event/384358/session/12/contribution/59/`
[13] Online, accessed Feb 21, 2016 URL `http://www.docker.io`
[14] Online, accessed Feb 21, 2016 URL `http://ccl.cse.nd.edu/software/parrot/`
[15] Saiz P, Aphecetche L, Buncic P, Piskac R, Revsbech J E and Sego V (ALICE) 2003 *Nucl. Instrum. Meth.* **A502** 437–440