Leveraging HPC resources for High Energy Physics

B. O’Brien¹, R. Walker², A. Washbrook¹

¹ SUPA, School of Physics and Astronomy, The University of Edinburgh, James Clerk Maxwell Building, Mayfield Road, Edinburgh, EH9 3JZ, United Kingdom
² Ludwig-Maximilians-Universitat Muchen, Munchen, Germany

E-mail: awashbro@ph.ed.ac.uk

Abstract. High Performance Computing (HPC) supercomputers provide unprecedented computing power for a diverse range of scientific applications. The most powerful supercomputers now deliver petaflop peak performance with the expectation of “exascale” technologies available in the next five years. More recent HPC facilities use x86-based architectures managed by Linux-based operating systems which could potentially allow unmodified HEP software to be run on supercomputers. There is now a renewed interest from both the LHC experiments and the HPC community to accommodate data analysis and event simulation production on HPC facilities. This study provides an outline of the challenges faced when incorporating HPC resources for HEP software by using the HECToR supercomputer as a demonstrator.

1. Introduction

As data continues to be successfully collected at the Large Hadron Collider at record rates it is expected that there will be a growing demand for computing resources to process and simulate data well above the current pledges from Tier-1 and Tier-2 sites on the LHC computing grid. There is now a renewed interest in HEP experiments investigating whether the use of High Performance Computing (HPC) facilities can be used to close the gap in CPU resources.

HPC queue utilisation is often less than full as high priority jobs requesting a large number of compute nodes wait for adequate resources to become available. A sample 24-hour utilisation of the HECToR supercomputing facility is shown in Figure 1. Significant ”backfilling” opportunities are therefore available to process high throughput workload which would not adversely impact existing user demand.

Despite the sizeable computing resources on offer there are a number of technical barriers that limit the use of HPC resources for High Energy Physics applications. HPC facilities have traditionally opted for specialised hardware architectures and favoured tightly coupled parallel MPI-based workloads rather than the high throughput commodity computing model typically used in HEP. However, the increasing adoption of x86 architectures and Linux based OS in HPC systems implies that HEP workloads are now possible if differences between the HPC environment and execution environment expected for HEP software running on the grid can be successfully resolved.

A feasibility study was performed to determine how an example HPC resource could be incorporated into a Tier-2 grid site hosted at the same facility. A pragmatic approach was taken to resolve any technical challenges encountered with the aim of providing feedback into a more
general design that can be used at other HPC sites willing to provide resources to High Energy Physics.

![Figure 1. Example 24-hour utilisation of the HECToR facility](image)

2. High Throughput Computing in a HPC environment

There are significant differences between the High Throughput Computing model (HTC) used to deliver resources to HEP and the computing environment provided for High Performance Computing. Some of the main differences are highlighted below:

- Network access to compute nodes is more restrictive than worker nodes on the grid with no WAN connectivity available on the compute nodes. All communication to and from the HPC facility is routed through dedicated gateway servers and in most cases only ssh/scp access is only allowed by default. Access to external data sources during the lifetime of a job is not possible and so any input data required for job execution is expected to be pre-fetched to the shared filesystem accessible to the compute node.

- Operating systems deployed on compute nodes may be more lightweight than grid-enabled worker nodes in order to optimise code execution by limiting the number of interruptions to compute processes. This may result in standard libraries and packages available on the standard worker node configuration to be missing on HPC compute nodes.

- All job input data is expected to reside on the shared filesystem rather than on the local disk of a compute node. HPC systems are not designed to cater for applications that process large input data sets or use sustained I/O calls during job execution.

- All user application software and associated dependencies must be resident on shared mounted file system.

- HPC jobs typically request a large number of processors across many compute nodes in contrast to typical HEP jobs that only request single-core CPU by default. Given the large amount of CPUs that can be reserved by a single job there is a necessary restriction on the amount of jobs submitted per user at any given time. As an example, Table 1 illustrates queue setup and queue limitation for the HECToR facility.

- Each HPC system has its own identity management policy that cannot be coupled to federated systems used by grid middleware. Users wishing to access HPC resources through these existing mechanisms would need to be mapped to an appropriate set of users managed by the HPC facility. Additional access rules are also enforced which will vary for each supercomputing facility.
Note that the above observations do not necessarily apply to all HPC systems and are driven by user expectations rather than by technical barriers. In some cases adjustments to system configuration can be potentially adapted to accommodate HEP workloads where reasonable.

| Compute nodes (max cores) | 20m | 1h | 3h | 6h | 12h | 24h |
|---------------------------|-----|----|----|----|-----|-----|
| 4 (128)                   | 64  | 64 | 64 | 64 | 64  | -   |
| 8 (256)                   | 64  | 64 | 64 | 64 | 64  | 16  |
| 16 (512)                  | 48  | 48 | 48 | 48 | 48  | 16  |
| 32 (1024)                 | 32  | 32 | 32 | 32 | 32  | 12  |
| 64 (2048)                 | 20  | 20 | 20 | 20 | 20  | 4   |
| 128 (4096)                | 16  | 16 | 16 | 16 | 16  | 2   |
| 256 (8192)                | 8   | 8  | 8  | 8  | 8   | -   |
| 512 (16384)               | 4   | 4  | 4  | 4  | 4   | -   |
| 1024 (32768)              | 2   | 2  | 2  | 2  | 2   | -   |
| 2048 (65536)              | 1   | 1  | 1  | 1  | 1   | -   |

Table 1. HECToR facility batch queues and job limits

3. An example HPC resource for HEP: HECToR
The High-End Computing Terascale Resource (HECToR) phase 3 facility [1] is a Cray XE6 system [2] offering a total of 2816 compute nodes and has theoretical peak performance of over 800 Tflops. As of June 2013 is it the 41st most powerful supercomputer in the world. Each compute node has two AMD 2.3 GHz 16-core processors and 32 GB of main memory which total 90 TB of shared memory and 90,112 cores across the facility. The processors are connected with a high-bandwidth interconnect using Cray Gemini communication chips. A 1PB Lustre distributed parallel file system is accessible from all compute nodes and user jobs are scheduled by PBSpro batch system. HECToR is a member of the PRACE Europe-wide supercomputing initiative [3].

Access to the HECToR supercomputer for HEP feasibility studies was enabled by middleware services available at the Edinburgh Tier-2 grid site. Compute resources for this site are provided by the Edinburgh Compute and Data Facility (ECDF) which is located at the same site as HECToR and so it was decided to incorporate these resources to leverage the existing middleware infrastructure installed for access to ECDF.

3.1. HPC Service Integration
An overview of the inclusion of HECToR into existing grid middleware services hosted at ECDF is shown in Figure 2. An ARC Computing Element (CE) server was deployed to handle incoming HPC job requests from external sources such as workload management systems and pilot factories.

Jobs submitted to this CE were then forwarded to the HECToR batch system by modifying ARC A-REX job management scripts to use passwordless ssh as part of the submission process. This was seen as a temporary solution due to access restrictions whilst the ARC server was external to the HECToR network. A more robust solution will be required if the ARC server could not be placed within HECToR network or if remote batch system access cannot be arranged.

In the simplest scheduling scenario shown on Figure 2 the ARC CE directly forwards jobs to the HECToR batch queue. However this is not the most efficient use of HPC resources especially when single-core job requests are submitted to the CE. It is therefore preferable to
submit a lower amount of jobs that request a larger amount of resources for batch processing of high throughput workloads. A job "marshall" script could therefore be used to aggregate job submission if the job volume submitted to the site can be anticipated. This type of job would reserve a suitable amount of cores for a selected time period by selecting the most suitable queue and would pull in workload in a similar manner to the pilot model used by current distributed computing models.

3.2. HPC Service Testing
A Monte-Carlo event generation script used by the ATLAS experiment was chosen as sample code to test the additional steps needed for execution in a HPC environment. This type of workload was relatively simple with no dependencies on external data but was representative enough to help identify any configuration assumptions or dependencies when executing the software.

The event generation software and auxiliary tools which is normally distributed by CVMFS was not available on the compute nodes. As a temporary measure a sample software release was copied and placed onto the HECToR shared filesystem. Some effort was then needed to ensure hard-coded paths in environment scripts contained in the CVMFS setup was modified. This setup was far from ideal.

Providing the correct environment setup for the execution of the code on HECToR compute nodes was found to be the most challenging aspect of the study and engagement with HPC administrators was required to resolve incompatibility issues. Any dependencies on packages and libraries were put into a dedicated external dependencies directory and job setup environment scripts and paths were modified to include this directory.

Figure 2. Job workflow from Tier-2 site (ECDF) to HPC facility (HECToR)
4. Outlook
HECToR is one of several HPC sites that recently started to accommodate the running of HEP software through dedicated share of resources or through the use of backfilling opportunities.

Despite the significant computational resources on offer it is non-trivial to migrate HEP software and in some cases these workloads may be completely unsuitable to run in an HPC environment. Current facilities are more suited to CPU-intensive event generation and Monte-Carlo simulation jobs that require little to no external job input, and complications with experiment software and associated dependencies, external data retrieval and site access will limit general use.

However some effort is now underway within the HEP community to overcome these challenges and to create a more general framework that can be deployed at HPC facilities such as HECToR. There is increasing engagement between the HPC and Grid communities to accommodate ”big data” processing and high throughput computing at the new class of supercomputer facilities. This feasibility study has provided an initial insight into how this can be achieved using an existing HPC facility. Effort will continue in this area with the eventual aim of providing a production-level supercomputing service for High Energy Physics applications.

References
[1] HECToR: The UK National Supercomputing Service: http://www.hector.ac.uk/abouthector/
[2] Cray XE Systems: http://www.cray.com/Products/Computing/XE.aspx
[3] Partnership for Advanced Computing in Europe: http://www.prace-ri.eu/