Many-core on the Grid: From Exploration to Production

T. Doherty[1]; M. Doidge[2]; T. Ferrari[1]; L. Salvador[1]; J. Walsh[4]; **A. Washbrook[5]**

[1]University of Glasgow, [2]University of Lancaster, [3]INFN CNAF, [4]Trinity College Dublin, [5]University of Edinburgh

Motivation

A number of High Energy Physics experiments have successfully run feasibility studies to demonstrate that many-core devices such as GPGPUs can be used to accelerate algorithms for trigger systems and data analysis [1,2]. After this exploration phase experiments on the Large Hadron Collider are now investigating how these devices can be incorporated into key areas of their software framework in advance of the significant increase in data volume expected in the next phase of LHC operations.

There is now increasing community interest for GPUs to be made available on the existing grid infrastructure. However, despite this anticipated usage there is no standard method available to run GPU-based applications in distributed computing environments. Before GPU resources are made available on the grid a number of operational issues such as job scheduling and resource discovery will need to be addressed.

Nvidia Tesla GPU (Image courtesy of Nvidia)

Tier-2 Site Experiences

The Edinburgh and Lancaster UK Tier-2 sites were selected to demonstrate GPU access on the Grid. Job submission direct to Computing Elements hosted by each site were validated to ensure that the new GPU queues were available to any authorised and authenticated Grid user. The properties of the GPU devices hosted at each of these sites in shown in Table 1 which also includes two additional GPUs available for local comparison tests. Notable changes to site configuration were:

- Installation of CUDA 5.x toolkit on worker nodes
- Separate batch system queues were defined to partition GPU-based workload from normal operations
- GPU-based jobs had exclusive access to worker node resources to avoid concurrent GPU device access
- GPU-specific job command flags were prepended to each incoming GPU-based grid job by modifying CE job submission scripts

Functionality and Performance Testing

A series of benchmark tests were created to check the functionality and performance of the GPU devices hosted at the candidate Tier-2 sites. These tests were derived from a subset of code examples available in the CUDA SDK [4]. Performance metrics on each test job were also gathered by using CUDA profiling tools [5].

Information Publishing

It is not expected that user jobs will be able to run on every type of many-core device due to compilation dependencies and requirements on functionality only available on later generation devices (e.g. dynamic parallelism). It will therefore be useful to extend information publishing tools to cover the capability of many-core devices advertised on the grid so that many-core jobs can be allocated matching resources. By using the output from the GPU functionality testing it was possible to highlight a selection of useful metrics of interest.

Job Accounting

Job accounting software could also be modified to measure the utilisation of many-core devices and to determine if these new resources are being used efficiently. CUDA profiling tools can measure the processing time spent by the GPU and the device occupancy for each kernel execution which could be aggregated to measure overall job performance. GPU processing time could be used to determine GPU efficiency either in relation to the total CPU time or to the overall job wallclock time. Device occupancy can also be used a secondary measure of efficiency to determine time-independent device utilisation.

Use Case: The EPIC Project

EPIC was chosen as an example Virtual Organisation to demonstrate a realistic use case of GPUs available on the Grid. The EPIC project [3] is developing an epidemiological model using the life histories of cattle and the location of farms to understand the dynamics of Bovine Tuberculosis transmission:

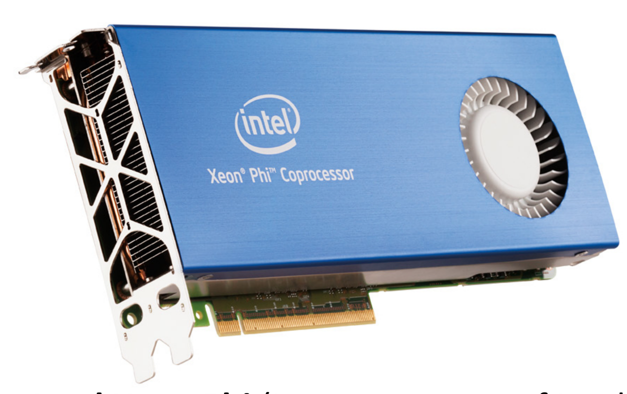
"The statistical nature of this work - where we need to run the stochastic model a few hundred times to get a stabilised average for each summary statistic - means the Grid is perfect for the scale of what we want to do. Spatial analysis techniques used to model thousands of agents lends itself well to GPU programming and will be very useful and extremely powerful for understanding a wide range of biological and ecological phenomena."

GPU Device Information					
Location	ECDF (1)	ECDF (2)	Lancaster	Test Host 1	Test Host 2
Model Name (Nvidia)	M2050	T10	M2075	C1060	K20c
CUDA Capability	2.0	1.3	2.0	1.3	3.5
Multiprocessors	14	30	14	8	192
Number of cores	448	240	448	240	2496
Global Memory	2687MB	4096MB	5375MB	4096MB	4800MB
Constant Memory	64kB	64kB	64kB	64kB	64kB
Shared Memory	48kB	16kB	48kB	16kB	48kB
Total registers per block	32768	16384	32768	16384	65536
Max threads per MP	1536	1024	1536	1024	2048
Max threads per block	1024	512	1024	512	1024
Max block size (x)	1024	512	1024	512	1024

Table 1: Device information for GPUs available at Tier-2 sites

Many Integrated Core Devices (MICs)

The inclusion of many-core devices into grid operations would also need to cover emerging Many Integrated Core devices (MICs) such as the Intel Xeon Phi. One such device is now available at the ECDF Tier-2 site and is being considered for integration as a grid resource. In contrast to GPUs there are less restrictions on device compatibility and so parallelised code should in principle work on any grid-enabled MIC device.



Intel Xeon Phi (Image courtesy of Intel)

Intel Xeon Phi 5110P Specs	
Cores	60
Clock Speed	1.0GHz
Max Mem size	8GB
Max Mem bandwidth	320GB/s

Table 2: Intel Xeon Phi 5110P Specifications

EGI GPGPU Working Group

The EGI GPGPU working group [6] will develop a Glue 2.0 Execution Environment (EE) definition for GPGPU resources. This is intended to describe the feature and state of GPGPU resources:

- Defining baseline device properties from vendor independent tools
- Runtime environment conventions
- Batch system integration
- Dynamic information such as availability of GPGPU resources

Contact

Dr. Andrew Washbrook
University of Edinburgh
Email: awashbro@ph.ed.ac.uk

Further Information

1. Algorithm Acceleration from GPGPUs for the ATLAS Upgrade P J Clark et al 2011 J. Phys.: Conf. Ser. 331 022031
2. Acceleration of multivariate analysis techniques in TMVA using GPUs A Hoecker et al 2012 J. Phys.: Conf. Ser. 396 022055
3. EPIC - The Scottish Government's Centre of Expertise on Animal Disease Outbreaks http://www.sruc.ac.uk/epic/
4. CUDA GPU computing SDK https://developer.nvidia.com/gpu-computing-sdk
5. CUDA profiler users guide http://docs.nvidia.com/cuda/profiler-users-guide/index.html
6. EGI Technical Forum 2013 https://indico.egi.eu/indico/sessionDisplay.py?sessionId=27&confId=1417#20130916