**Low-Cost High-Performance Computing Via Consumer GPUs1**

P. Somaru, I. Obeid and J. Picone

The Neural Engineering Data Consortium

Temple University

pat.somaru@temple.edu, iobeid@temple.edu, joseph.picone@gmail.com

Big data and machine learning are rapidly developing fields with evolving and increasingly diverse hardware requirements. The goal of this project was to demonstrate that an enterprise-ready, Warewulf-based HPC compute cluster could support heterogeneous hardware via the integration of a GPU compute server. The benefits of this were two-fold. First, the integration of the GPU compute server into the cluster would validate its ability to support heterogeneous hardware, demonstrating that our approach to HPC cluster management is an effective long-term strategy. Second, a GPU compute server would significantly reduce the runtime of experiments, increasing the rate at which research advancements are made.

Due to the nature of the tasks involved in working with big data and machine learning, the need to support heterogeneous hardware is great. When working with big data, there is no such thing as a “minor task.” Apparently simple tasks, such as appending a newline to the end of each text file within a dataset, can take days. For tasks such as these traditional computing, computing in which code is written to be compiled and executed by a CPU, is the best approach. Languages such as Python and Java have well developed libraries, such as Python’s OS.walk, to abstract away the complexity involved in day-to-day computing tasks. When working with machine learning, arithmetic operations are critical since these tasks involve large numbers of relatively simple computations. Machine learning tasks involve the manipulation of vectors, so specialized hardware that can vectorize linear algebra operations becomes critical to boosting performance. I/O also becomes important since these tasks operate on large data sets that cannot be held in memory.

Designing a computing environment that can support both specialized hardware and traditional CPUs, as well as address the needs of big data and machine learning, is a multi-faceted problem. From the perspective of those writing code, the problem is that is that learning the intricacies of or re-writing code for each new piece of specialized hardware is an infeasible task. From the perspective of those running code, the problem is that writing wrappers for each piece of specialized hardware is inefficient. From the perspective of administrators, the problem is that any solution needs to be general-purpose enough to be usable across all machines while abstracting away the particulars that need addressing while working with specialized hardware. From the perspective of investors, the problem is that any solution needs to scale with minimal overhead and require minimal maintenance.

This work builds on the work of Trejo et al. presented at the 2015 IEEE SPMB in which a low-cost high performance Linux cluster was described. The goal of our work was to verify that the manner in which we addressed these issues worked by adding a GPU compute node to the cluster. The addition of this node increased the compute capacity of this HPC cluster from 390 GFlops to 22.5 TFlops. The GPU compute node added cost $5.8K. This node has proven to reduce runtimes for experiments that optimally use it by at least two orders of magnitude.

The main software components of the final system configuration included Warewulf, Torque, Maui and Ganglia. The system we describe is a cluster with 5 compute nodes that includes: 128 cores, a 20TB RAID, 1TB of RAM, and 4 GPUs. The main node is equipped with two Intel Xeon (4C) @ 3.0 GHz CPUs. Each CPU compute node is equipped with two AMD Opteron (16C) @ 2.4GHz CPUs. For the CPU compute nodes, we went with a high core count since our jobs are batch processing based and use little memory. The GPU node consists of 4 Nvidia 980GTX 6GB GPUs, 128 GB of RAM, a 100GB SSD and two 3-core Intel CPUs in a SuperMicro rackmount server kit. The main node resides in a 24-Bay 4U chassis and supports 10Gb/sec networked communications. The density of the compute nodes, defined as “performance/(cost\*volume)” is quite impressive since they only occupy 3U and have plenty of room for expansion. The total system cost was $31K. The system delivers 22.5 TFLOPS, which translates to a very competitive 726 MFlops/$.

Accommodating heterogeneous hardware is crucial to our long-term strategy of supporting low-cost upgrades and minimizing the cost of cycles. New compute nodes can be easily added to the system. The system is being used to develop AutoEEG on a large corpus of over 28,000 EEGs as part of a commercialization effort. We will discuss some of our experimental results generated with the system.

1. Research reported in this publication was supported by the National Human Genome Research Institute of the National Institutes of Health under Award Number U01HG008468.

