Low-Cost High-Performance Computing Via Consumer GPUs[1]

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.

NEURAL ENGINEERING DATA CONSORTIUM
www.nedcdata.org

Low-Cost High-Performance Computing Via Consumer GPUs

Pat Somaru, Dr. Iyad Obeid and Dr. Joseph Picone

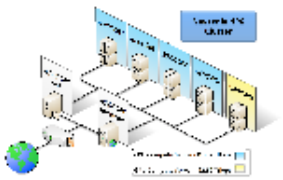
The Neural Engineering Data Consortium, Temple University

College of Engineering
Temple University

Abstract

- 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.
- Our initial cluster implementation, designed in Summer 2015, had 4 compute nodes with 256 GB of memory and 32 cores per node, 20 TB of network-mounted disk space and a compute capacity of .4 TFlops. This cluster cost $26.5k.
- In Summer 2016, we added a GPU-based compute node that had 4 Nvidia 980 GTX Ti GPUs, each equipped with 6GB of RAM, adding 22.5 TFlops to the cluster. The new server cost $5.8K.
- Each of these GPUs is capable of completing our deep learning benchmark, which takes 2,362 minutes to complete on 16 CPU cores, in 2 minutes. Four of these jobs can be run in parallel, representing a 288x increase in throughout when the old cluster is compared to this SINGLE node.
- The unprecedented price/performance ratio of this GPU node is transforming the way we train deep learning systems and enabling significant optimizations of our technology.

Cluster Overview



- The Neuronix HPC cluster consists of 7 nodes:
  - Login Sever (nedc_000): dual 8-core Intel CPUs, a 20TB hardware RAID array, 64GB of RAM and two 1Gbit/s NICs.
  - Web Server (nedc_001): dual 3-core Intel CPUs, a 5TB hardware RAID array, 16GB of RAM and two 1Gbit/s NICs. This server hosts two popular scientific websites: www.isip.piconepress.com and www.nedcdata.org.
  - Compute Nodes (nedc_002-005): each node has a one 1Gbit/s NIC, two 16-core 2.4 GHz AMD CPUs, a 500GB SSD and 256GB of RAM.
  - GPU Node (nedc_006): 4 Nvidia GPUs, 128GB of RAM, two 3-core Intel CPUs and a 100GB SSD.
- This cluster also contains an internal 1 Gigabit ethernet switch. The login server provides DHCP, TFTP, and NFS service for all other nodes.
- Each node is also has an IPMI chip connected to the internal router, granting physical-equivalent access.

System Design

Software Infrastructure

- The NeuroNix cluster's software was selected for the purposes of maximizing maintainability, scalability and support for hardware heterogeneity.
- Our cluster management suite consists of the Torque resource manager, the Maui job scheduler, the Warewulf operating system provisioner and the Ganglia resource monitor.
- These software tools are all open source, preventing vendor lock-in. Our use of their advanced functionality is kept to a minimum to maintain the portability of our development environment.
- While configuring the cluster, we configured our environment to select the software to be ran on a node based upon it's CPU architecture. This was necessary as the performance of computationally intense programs, such as Kaldi, can benefit significantly from the use of processor specific instruction sets.
- All compute nodes are booted via stateless provisioning. This was done as it allows for new hardware to be added without having to undergo a security audit.
- Installing all GPUs required upgrading to CentOS 7.

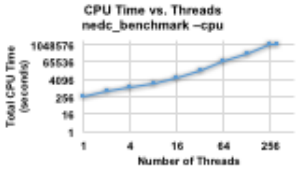
GPU Selection

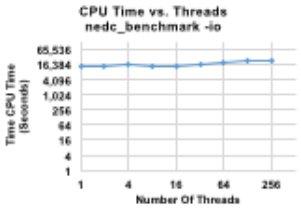
- The GPUs we selected were the Nvidia GTX 980 Ti with 6GB of ram. We purchased four of these.
- Each of these GPU's cost $647.99. Each GPU is capable of 5.63 trillion floating point operations per second.
- The Nvidia GTX 980 Ti is a consumer GPU. By selecting this GPU, we were able to get 22.5 Tflops of GPU compute capacity for 8.7 Gflops per dollar.
- We decided to use consumer GPUs as the most low-end, least-expensive enterprise GPU available, the Nvidia M2090, had cost 1299.99 dollars at the time of purchase. This GPU has a compute capacity of 1.33 TFlops, priced at 1.02 Gflops per dollar at the time of purchase.
- Selecting consumer GPUs restricted the hardware we could purchase. NVIDIA's enterprise GPU's are tested by server manufacturer's to ensure compatibility and fit. Fewer server's are tested for compatibility and fit with consumer GPUs.
- Selecting consumer GPUs also meant not having any double precision GPU compute capacity. This was acceptable as the deep learning library that we primarily use, Theano, does not support the concurrent use of double precision float's and GPU acceleration.
- We purchased with a SuperMicro SYS-1028GQ-TR server with two three-core Broadwell Xeon Processors, 128 GB of RAM and a 100GB Intel SSD to support the GPUs. The total cost of nedc_006 was $5.8K.

Benchmarking and Testing

Benchmarks

- The NeuroNix cluster was benchmarked with our benchmark, nedc_benchmark. nedc_benchmark has two modes, I/O and CPU.
  - The I/O benchmark performs processes a list of EDF files distributed over the number of threads at which the benchmark is ran.
  - The CPU benchmark processes an array containing doubled a fixed number of times, performing of multiplications and transcendental operations each time.
- These tests were used in conjunction to verify that both CPU bound and file I/O bound jobs ran on the cluster would not experience, and would not cause concurrent jobs to experience, significant bottlenecks when the number of threads ran in parallel were between 1 and 256.
- The following graphs demonstrate that the cluster did not experience significant bottlenecks for either the I/O benchmark or the CPU benchmark. These benchmarks were also written to demonstrate the maximum respective workloads possible while using our libraries, assuring that code using these libraries would would not bottleneck at either CPU usage or file I/O, provided that less than 256 parallel threads were being processed.



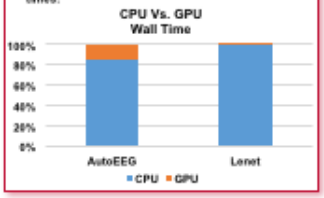


- As a result of this benchmark, we have decided that while it may be beneficial to move data to the SSDs, it was not necessary to do so at the moment.

GPU Performance

GPU Benchmark

- As we are in the early stages of optimizing our software for GPUs, we used a Theano experiment in which a LeNet model is trained to recognize digits to benchmark the GPUs. The speed-up resultant from using GPUs is 230ix.
- For the dataset with which we are currently using GPUs to work upon, the speed-up resultant from using GPUs is 5x.
- The disparity between these speed-ups is due to the constant amount of time necessary for data preparation. We expect that the speed-up for our experiments will increase in a similar manner first when the size of the dataset has been increased, and once more once data preparation has been separated from training. The following graph displays the discussed GPU and CPU processing times.



Summary

- The addition of nedc_006 to the Neuronix cluster will enable faster research by allowing researcher's to develop better software faster by allowing them to test and debug more code in less time.
- Nedc_006 will also enable researchers to work with deep learning models complex enough to more correctly identify complex relation's.

Acknowledgements

- Research reported in this poster was supported by National Human Genome Research Institute of the National Institutes of Health under award number 1U01HG008468.
- The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.