The Neuronix HPC Cluster: Cluster Management Using Free and Open Source Software Tools[1]

C. Campbell, N. Mecca, I. Obeid and J. Picone

The Neural Engineering Data Consortium, Temple University
{christopher.campbell, tuf89560, iobeid, picone}@temple.edu

One of the most notable impacts of computing advancements over the last few decades has been the decentralization of resources. As the cost of computer hardware continues to decrease, significant computational power continues to become more accessible to the consumer market. Similar to personal computing, this phenomenon has also enabled the growth of low-cost high-performance computing (HPC) (e.g., desktop supercomputers). Combined with advances in computational statistics and machine learning, HPC systems can now accommodate computationally expensive research using consumer-grade hardware.

Graphic Processing Units (GPUs) have become an integral part of today's high-performance compute cluster [1][2]. GPUs are absolutely critical now to a new generation of big data machine learning systems that require massive amounts of computing to develop. These chips and the software that supports them adds additional burden to cluster management. Key issues include software compatibility (e.g., Nvidia's CUDA support is problematic), as well as job control (e.g., open source schedulers do not seem to adequately support nodes with multiple GPU chips) and load balancing by distributing computing jobs to compute nodes based on the state they report to the main node. When there are large numbers of compute nodes in the cluster, system administration of these nodes becomes a time-consuming process. The goal of this poster presentation is to introduce researchers to cost-effective ways to manage such resources.

In the Neuronix cluster, we manage compute nodes by placing them under the control of the main server (i.e. CPU/GPU compute nodes). We use Warewulf [3] for operating system provisioning as well as for synchronizing important system files such as the hosts and password files. Warewulf boots the compute nodes over the network from kernel and filesystem images on the main server. The primary advantage of this architecture is that changes can be made to one set of images and sent to all the nodes. For the nodes that function independently of the main node (e.g. backup servers, web servers), we are in the process of implementing Ansible [4] to automate their setup and configuration.

The queue manager that controls job submission is composed of a resource manager (TORQUE [5]), monitors node resource statistics. It also handles everything related to submitting and running jobs from the main node on one or more compute nodes. Torque is accompanied by a job scheduler (Maui [6]) that communicates with the resource manager and, based on the status of the compute nodes and the internal scheduling of node can be requested, and setting scheduling single or multi-dimensional scheduling policies (e.g. setting scheduling policies per user per queue).

A number of free and open source monitoring tools are available to keep track of system statistics (e.g. network bandwidth, CPU/memory usage) and hardware failures. As computing systems scale, it is important to identify and resolve bottlenecks, which can limit the performance gain from scaling. Ganglia [7] is a system monitoring software that collects information from cluster nodes and displays the information graphically through a web interface [8]. *Mdadm* is a standard Linux utility that we use to manage the RAID arrays on the cluster. With these arrays, the cluster becomes significantly more robust in the face of hard drive failures [9]. *Smartctl* is another standard Linux utility that can be used to report information about the status of all the hard drives (e.g. the number of bad sectors).

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

There is a very large space of potential software solutions for these clusters. The goal of this abstract is to introduce readers to a core set of tools we find useful in developing and maintaining a low-cost cluster. In this poster, we discuss the tools we find most useful in efficiently managing our cluster and will provide a demo. We emphasize tools that are easy to learn and yet provide the necessary capabilities to manage a heterogeneous cluster. We provide support to a growing base of users on these issues.

REFERENCES

[1] V. V Kindratenko, J. J. Enos, G. Shi, M. T. Showerman, G. W. Arnold, J. E. Stone, J. C. Phillips, and W. m. Hwu, "GPU clusters for high-performance computing," *Proceedings of the IEEE International Conference on Cluster Computing and Workshops*, 2009, pp. 1–8. Retreived from *https://doi.org/10.1109/CLUSTR.2009.5289128*.

[2] K. Sajjapongse, T. Agarwal, and M. Becchi, "A flexible scheduling framework for heterogeneous CPU-GPU clusters," *Proceedings of the 21st International Conference on High Performance Computing* (HiPC), pp. 1–11, 2014. Retrieved from *https://doi.org/10.1109/HiPC.2014.7116892*.

[3] Lawrence Livermore National Laboratory, "Welcome to the Warewulf Project," 2017. Retrieved from *http://warewulf.lbl.gov/*.

[4] The Redhat Corporation, "Ansible," 2017. Retreived from *https://www.ansible.com/*.

[5] Adaptive Computing, "The Torque Resource Manager," 2017. Retrieved from *http://www.adaptivecomputing.com/products/open-source/torque/*.

[6] Adaptive Computing, "The Maui Job Scheduler," 2017. Retrieved from *http://www.adaptivecomputing.com/products/open-source/maui/*.

[7] M. Massie, "The Ganglia Monitoring System," 2017. Retrieved from *http://ganglia.info/*.

[8] C. T. Yang, T. T. Chen, and S. Y. Chen, "Implementation of Monitoring and Information Service Using Ganglia and NWS for Grid Resource Brokers," *Proceedings of the 2nd IEEE Asia-Pacific Service Computing Conference* (APSCC 2007), 2007, pp. 356–363. Retrieved from *https://doi.org/10.1109/APSCC.2007.74*.

[9] Y. Joshi, D. Sharma, U. Gaur, V. Kumar, and R. Kalmady, "Design of a Fault Tolerant Architecture for Private Cloud Computing Infrastructure," *Indian Journal of Science and Technology*, vol. 10, no. 4, 2017. Retrieved from *https://doi.org/10.17485/ijst/2017/v10i4/110663*.

NEURAL ENGINEERING DATA CONSORTIUM
www.nedcdata.org

College of Engineering
Temple University

The Neuronix HPC Cluster: Cluster Management Using Free and Open Source Software Tools

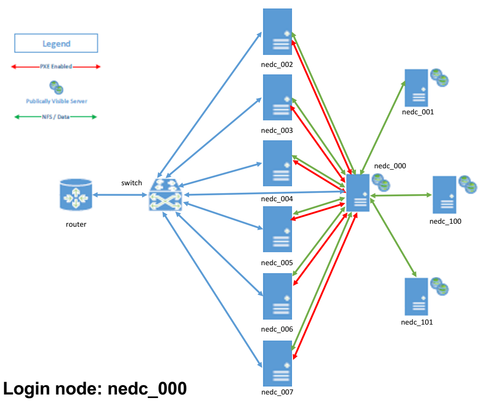
C. Campbell, N. Mecca, I. Obeid and J. Picone

The Neural Engineering Data Consortium, Temple University

Abstract

- In order to be effective, machine learning must operate on problems of scale, requiring suitably large data and computing resources.
- HPC clusters based on open source software and consumer grade hardware have enabled a new generation of extremely computationally demanding research based on deep learning and big data.
- In this poster we discuss the Neuronix cluster, an implementation of the HPC cluster concept that provides an unprecedented price/performance ratio using commercial off the shelf parts (COTS).
- The environment is heterogeneous because of the need to mix GPUs and CPUs. GPUs are critical today to the success of deep learning algorithms.
- Methods of horizontal scaling and managing node availability based on requested resources and server load are discussed.
- Tools that are central to our management strategy include Ganglia, mdadm and smartctl.

The Neuronix Cluster

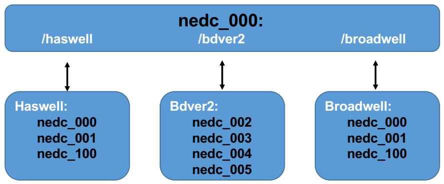


Login node: nedc_000

- 42 TB HDD (14x3TB), 64 GB RAM, 2x Intel ® Xeon ® E5-2623 v3 (3.00 GHz)

- CPU compute nodes: nedc_002 – nedc-005
  - 500GB SSD, 256 GB RAM,2x AMD Opteron™ 6378 (2.4 GHz)
- GPU compute node: nedc_006
  - 128GB SSD, 128 GB RAM, 2x Intel ® Xeon ® CPU E5-2603 v4 (1.70GHz), 4x NVIDIA GeForce GTX 980 Ti
- GPU compute node: nedc_007
  - 500GB SSD, 128 GB RAM, 2x Intel ® Xeon ® CPU E5-2603 v4 (1.70GHz), 4x NVIDIA GeForce GTX 1070
- Newest node (nedc_008): 4x NVIDIA Tesla P40

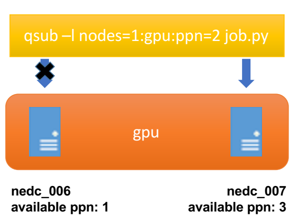
Cluster Management



A Heterogeneous Microarchitecture Environment

- Compute nodes are booted from the network (PXE) with Warewulf (Lawrence Berkeley National Labs).
- A single kernel/initramfs (bootstrap) and root filesystem (VNFS) image combination can boot any number of machines over the network, allowing nodes to be added to the cluster in O(1) time.
- Software tools and data are made available to the compute nodes via NFS mount of the login node.
- Since the nodes on the cluster have CPUs using 3 different microarchitectures, each type of node has its own software environment, stored on nedc_000.
- The use of RAID arrays provides fault tolerance in the event of disk failure.
- Ganglia provides a web-based method for viewing node statistics.

Job Submission



- The qsub command indicates what resources are required to execute the job.
- The resource manager will find a node that is able to fulfill the requirements of the job.

- Jobs are managed through the queue manager using a combination of Torque (resource manager) and Maui (scheduler).
- Users can submit their jobs using qsub, with which resources can be requested as well (otherwise they will use the system defaults).
- Jobs will wait in the queue until resources become available; the scheduler can be configured to provide the desired distribution of cluster resources.
- Node availability (i.e. online/offline) can be managed through Torque.
- Nodes can be given different resource specifiers to control what type of node a job runs on.
- For example, CPU compute nodes have the 'normal' resource and GPU compute nodes have the 'gpu' resource, so jobs can be directed to run on either a CPU or GPU node.

Job Scheduling Strategies

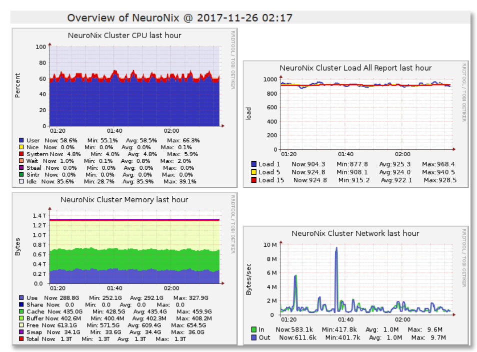
- Virtual processors: An abstract unit provided by the resource manager, can be used to represent compute threads, GPUs or a general measure of the processing power required for a job.
- Maui scheduler doesn't honor requests for GPUs when jobs are submitted (e.g. "qsub –l gpus=1...").
- A strategy to manage GPU scheduling is to map one virtual processor in the resource manager to one GPU on a compute node.
- In this manner, a user may request a GPU using "qsub –l ppn=1 …", where ppn=1 (virtual processor) is understood to correspond to a GPU.

Redundant Array of Independent Disks

- Used to expand the capacity of a filesystem beyond a single physical disk.
- Introduces fault tolerance by allowing for the failure of one or multiple drive failures.
- The Neuronix cluster makes use of RAID level 6, which prevents data loss in the event of any 2 disks in the array failing.
- Higher levels of RAID have not been useful.

Monitoring

- Ganglia: monitoring system that collects data from nodes in a cluster/grid and displays the data in graphical form from a web interface.
- Reported statistics include memory usage, network throughput and disk I/O.
- Other system monitoring tools include mdadm for software RAID arrays, which monitors the health of the disks in the array, and smartctl, which monitors the health of individual sectors in hard drives.

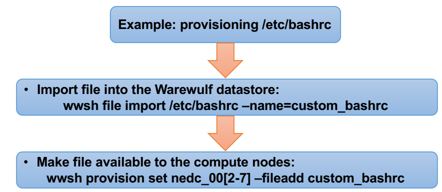


Configuration Management

- Ansible: a tool that allows for automating the setup and configuring of different types of node (e.g. file server, archive machine), reducing time to launch and ensuring that configurations are identical.

Provisioning with Warewulf

- In addition to operating system images, any file from the master node can be made available to the compute nodes.
- Files are added to Warewulf's data store on the main node, and then can be provisioned (made available) to compute nodes.
- Can be used for user authentication by provisioning /etc/passwd, /etc/shadow and /etc/group.



- When using driver software in a heterogeneous environment (e.g. NVIDIA drivers on nodes with different versions of the Linux kernel), the initramfs created by Warewulf can be manually edited to add the required kernel modules.
- By default, Warewulf initramfs images are located in /var/lib/tftpboot/warewulf/bootstrap/ and can be manipulated with the cpio command.
- When changes are made to a Warewulf VNFS image, the compute nodes using that VNFS would normally have to be rebooted for the changes to take effect. If this is undesirable, the wwgetvnfs command can be used to reload the entire filesystem without rebooting the node.

Summary

- A heterogeneous cluster using a free software stack running on a collection of inexpensive compute nodes can be used to work on even the most computationally intensive problems.
- Computing resources can be quickly assimilated using Warewulf and the queue manager, and can be administrated from the main node
- RAID and monitoring utilities can be used to minimize downtime, identify system bottlenecks and determine future growth directions.
- The Neuronix cluster has has 1.4TB of RAM, over 200 TB of hard disk capacity and 18944 CUDA cores

Acknowledgements

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