Expanding an HPC Cluster to Support the Computational Demands of Digital Pathology[1,2]

C. Campbell, N. Mecca, T. Duong, I. Obeid and J. Picone
The Neural Engineering Data Consortium, Temple University
{christopher.campbell, nmecca, thuc.duong, iobeid, picone}@temple.edu

The goal of this work was to design a low-cost computing facility that can support the development of an open source digital pathology corpus containing 1M images [1]. A single image from a clinical-grade digital pathology scanner can range in size from hundreds of megabytes to five gigabytes. A 1M image database requires over a petabyte (PB) of disk space. To do meaningful work in this problem space requires a significant allocation of computing resources. The improvements and expansions to our HPC (high-performance computing) cluster, known as Neuronix [1], required to support working with digital pathology fall into two broad categories: computation and storage. To handle the increased computational burden and increase job throughput, we are using Slurm [3] as our scheduler and resource manager. For storage, we have designed and implemented a multi-layer filesystem architecture to distribute a filesystem across multiple machines. These enhancements, which are entirely based on open source software, have extended the capabilities of our cluster and increased its cost-effectiveness.

Slurm has numerous features that allow it to generalize to a number of different scenarios. Among the most notable is its support for GPU (graphics processing unit) scheduling. GPUs can offer a tremendous performance increase in machine learning applications [4] and Slurm's built-in mechanisms for handling them was a key factor in making this choice. Slurm has a general resource (GRES) mechanism that can be used to configure and enable support for resources beyond the ones provided by the traditional HPC scheduler (e.g. memory, wall-clock time), and GPUs are among the GRES types that can be supported by Slurm **Error! Reference source not found.**. In addition to being able to track resources, Slurm does strict enforcement of resource allocation. This becomes very important as the computational demands of the jobs increase, so that they have all the resources they need, and that they don't take resources from other jobs. It is a common practice among GPU-enabled frameworks to query the CUDA runtime library/drivers and iterate over the list of GPUs, attempting to establish a context on all of them. Slurm is able to affect the hardware discovery process of these jobs, which enables a number of these jobs to run alongside each other, even if the GPUs are in exclusive-process mode.

To store large quantities of digital pathology slides, we developed a robust, extensible distributed storage solution. We utilized a number of open source tools to create a single filesystem, which can be mounted by any machine on the network. At the lowest layer of abstraction are the hard drives, which were split into 4 60-disk chassis, using 8TB drives. To support these disks, we have two server units, each equipped with Intel Xeon CPUs and 128GB of RAM. At the filesystem level, we have implemented a multi-layer solution that: (1) connects the disks together into a single filesystem/mountpoint using the ZFS (Zettabyte File System) [6], and (2) connects filesystems on multiple machines together to form a single mountpoint using Gluster [7].

ZFS, initially developed by Sun Microsystems, provides disk-level awareness and a filesystem which takes advantage of that awareness to provide fault tolerance. At the filesystem level, ZFS protects against data corruption and the infamous RAID write-hole bug by implementing a journaling scheme (the ZFS intent log, or ZIL) and copy-on-write functionality. Each machine (1 controller + 2 disk chassis) has its own

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.
2. This material is also based in part upon work supported by the National Science Foundation under Grant No. CNS-1726188. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

separate ZFS filesystem. Gluster, essentially a meta-filesystem, takes each of these, and provides the means to connect them together over the network and using distributed (similar to RAID 0 but without striping individual files), and mirrored (similar to RAID 1) configurations [8].

By implementing these improvements, it has been possible to expand the storage and computational power of the Neuronix cluster arbitrarily to support the most computationally-intensive endeavors by scaling horizontally. We have greatly improved the scalability of the cluster while maintaining its excellent price/performance ratio [1].

REFERENCES

[1] D. Houser, G. Shadhin, R. Anstotz, C. Campbell, I. Obeid, J. Picone, T. Farkas, Y. Persidsky and N. Jhala, "The Temple University Hospital Digital Pathology Corpus," *IEEE Signal Processing in Medicine and Biology Symposium*, 2018, p. 1.

[2] C. Campbell, N. Mecca, I. Obeid, and J. Picone, "The Neuronix HPC Cluster: Improving Cluster Management Using Free and Open Source Software Tools," *IEEE Signal Processing in Medicine and Biology Symposium*, 2017, p. 1.

[3] A. B. Yoo, M. A. Jette, and M. Grondona, "SLURM: Simple Linux Utility for Resource Management," in *Job Scheduling Strategies for Parallel Processing*, 2003, pp. 44–60.

[4] J. Schmidhuber, "Deep Learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[5] "Generic Resource (GRES) Scheduling." [Online]. Available: *https://slurm.schedmd.com/gres.html*.

[6] J. Bonwick, M. Ahrens, V. Henson, M. Maybee, and M. Shellenbaum, "The Zettabyte File System," in *Proceedings of the 2nd Usenix Conference on File and Storage Technologies*, 2003, pp. 1–13.

[7] "What is Gluster?" [Online]. Available: *https://docs.gluster.org/en/v3/Administrator Guide/GlusterFS Introduction/*.

[8] B. Depardon, G. Le Mahec, and C. Seguin, "Analysis of Six Distributed File Systems," Institut National de Recherche en Informatique et en Automatique (INRIA), Lyon, France, 2013. *https://hal.inria.fr/hal-00789086/document*.