CMS ML Training Facility

Like many fields, particle physics has seen an explosion in the use of Machine Learning (ML) to perform their research. Initially, these algorithms were targeted towards small, independent parts of CMS' workflows (e.g. tagging), but as the state-of-the-art of ML representations have progressed, these models have become more complex, capable of processing increasingly large parts of the data produced by the detector. The complexity of the models and volumes of data processed means CMS is quickly outstripping the ability to perform training of these models on small ad-hoc training initiatives.

For example, the ParticleNet model (https://arxiv.org/pdf/1902.08570.pdf) has a mere 366k weights, but a hyperparameter search to train this particular model took nearly two weeks on a single GPU. If instead, the training was performed on a distributed basis over many GPUs, this two week turnaround could be on the order of hours.

There are a number of techniques to utilize multiple GPUs for training, but performing them efficiently on an ad-hoc basis involves a level of technical knowledge that most users do not and should not require. Analogously to how the grid abstracted away the details around large scale utilize large-scale CPU resources A distributed GPU training facility, with sufficient processing, networking, and storage would be able to significantly decrease the amount of time needed to develop this model, decreasing the time-to-publish of this researcher. We propose to prototype such a facility at Vanderbilt.

Summary

We propose to leverage the knowledge and hardware available within the Vanderbilt research computing center (ACCRE) to develop a distributed GPU ML training facility. Such a facility requires not just a number of GPUs to perform the training, but also fast network interconnects and storage coupled together to keep the GPUs fully utilized. The second important component of this facility is to provide a common, stable, performant software substrate for users to develop, submit, and analyze their training workflows.

Both the diversity of ML needs and the speed at which the state of the art advances necessitates that this initial resource have as broad a capability set as possible – given a choice between capabilities and scale, we choose to invest more heavily in capabilities. Future iterations of this facility would use user feedback and needs to direct potential future expansions.

Vanderbilt has internal funding available for GPUs, though not the host machines themselves. We propose combining VU-purchased GPUs with U.S. CMS-purchased hardware to develop this proof of concept, and are seeking additional funding from VUs computational biology department, who would also assist developing this PoC.

Hardware

Training large models requires exploiting model-parallelism and/or data-parallelism. Roughly speaking, model parallelism means spreading a single instance of a large model over multiple GPUs, while data parallelism means instantiating multiple instances of the model and then sharding the large training data amongst these models.

To support both of these use-cases, we propose to utilize two different classes of hardware:

- To demonstrate model-parallel capability, we will exploit large 8-GPU machines with fast inter-GPU interconnects, like a DGX or similar. The interconnects allow the model to be larger than what an individual GPU could store in RAM, with a minimal performance impact.
- To demonstrate data-parallel capability, we will use several more modest GPUs attached to quad-GPU host machines. It is, of course, possible to utilize data-parallelization on much more performant hardware, but this comes at a significant cost, since the performance-per-dollar of the larger machines is much higher, not to mention that a lot of the cost of those machines are in the interconnects, which would be (relatively) unused by these types of models

In addition to these GPU resources, we propose to purchase a modest amount of high-performance NVMe-based storage to be used as a buffer for training input data. With the proper hardware, training data can be read from the storage directly into the GPU memory, bypassing the host CPUs. This is a capability we would like to investigate with this pilot installation.

We have received quotes for the host machines, and propose to purchase 3 machines: two of which support 4 GPUs ($23,545) and an additional machine which supports 8 GPUs ($38,750) – a total of $85,840 (subject to availability). These machines would each contain:

- The ability to host four or eight GPUs
- Two dual-port 100Gigabit network adaptors (NICs), which support Remote Direct Memory Acccess (RDMA)
    - It is important to have two separate NIC so that each NIC is directly attached to two GPUs via the PCI-Express fabric, enabling data to be moved directly from the network into the GPU memory. Without the second NIC, half of the GPUs would be separated by the (much slower) inter-socket CPU connections
- 1TB of RAM
- Two 9000-series AMD EPYC processors
- 60TB of NVMe storage

These machines will be populated with GPUs provided by Vanderbilt and hosted at ACCRE's datacenter, where they will be connected to existing networking capacity. The Proof-of-Concept hardware described above will be maintained by Vanderbilt at no additional cost to U.S. CMS for their entire lifecycle.

Software

A key to the success of this facility is to develop, deploy, and support a software infrastructure which allows researchers to develop and maintain training at-scale using best practices, without requiring that the researchers themselves be experts in the intricacies of the underlying hardware deployment. Providing a higher-level, maintained infrastructure which abstracts away certain details can lower the barrier of entry to future users and helps ensure that the resulting community can be more easily supported.

ACCRE has traditionally provided access to their compute resources via the SLURM batch system. While this system works very well, many in industry (particularly in the AI/ML field) have converged on Kubernetes as the underlying resource allocation and application deployment substrate. We propose to deploy these resources within Kubernetes to allow us to leverage the widespread availability of “cloud-native” training software. As an added bonus, Kubernetes itself can be considered a type of abstraction layer, which would allow a researcher to burst to and/or utilize cloud resources if desired.

As part of this prototype facility, we propose to deploy and integrate common ML training frameworks such as determined.ai to this Kubernetes infrastructure, and evaluate each of them on ease-of-use, feature set, and overall performance. Fortunately, the isolation features provided by Kubernetes don’t preclude the possibility of running multiple training frameworks simultaneously. Therefore if there is a defined need within the community for a specialized framework, this can be added with little interference with other frameworks.

Milestones and Deliverables

With the expected schedule to receive and provision hardware for this ML training facility (expected in Fall ‘23), we propose the following deliverables and corresponding milestones, contingent on receipt of hardware on Oct 1.

- Nov 1 - Perform training on a toy ML model, demonstrating that the facility is functional for standard single-GPU training.
- Jan 1 - Re-implement the training workflow using two of the aforementioned distributed ML training frameworks. Document and present these results to the appropriate CMS ML forum, extending an invitation for other (non-Vanderbilt) CMS collaborators to use this facility.
- May 1 - In consultation with U.S. CMS R&D staff, select and port an in-production ML model to be able to be trained at this facility. Demonstrate this functionality at the appropriate ML forum.

- Mar 1 - Demonstrate the utilization of RDMA to perform model-parallel training of a large CMS model over multiple GPUs. Document this work for the larger CMS audience to enable others to repeat this work for their own models.
- Jun 1 - Demonstrate the ability to use GPUDirect to quickly stream input data to GPUs for data-parallel training, again documenting this functionality so others can exploit this capability
- Jul ‘24 - Produce and lead a training session at the Fermilab HATS summer series. This will serve the purpose of both training users how to submit and train their models as well as to demonstrate the ability for this facility to be used by multiple users at once
- Sep 1 - Produce a report, recommending what a “standard” ML training facility should be capable of, and best practices for users to be able to effectively train their models at such future facilities.

Effort

This facility will require the expertise of computing professionals across the entire spectrum, beginning with the lower level hardware and networking provisioning, continuing through the kubernetes substrate and associated training frameworks, culminating with ML domain experts to test the infrastructure.

Andrew Melo will be leading and co-ordinating this effort, as well as providing technical expertise (0.17FTE from Vanderbilt’s NSF base grant). U.S. CMS has generously provided 0.5FTE postdoc through the R&D initiative, who will be primarily responsible for porting, testing, and documenting CMS ML workflows to this new facility. Additionally, Vanderbilt Office of the Vice Provost of Research Initiatives (OVPRI) will provide 0.5FTE of staff effort as a match to this postdoc effort. This effort will be responsible for the underlying hardware and GPU support (e.g. configuring RDMA). Finally, we hope to encourage early adopters both within CMS and Vanderbilt to gain user experience as early as possible to help us guide the development of this facility. We have spoken with researchers from both CMS and Vanderbilt who have expressed interest in this facility, and are confident they would be willing to help contribute to this effort.

Evaluation

As an initial goal, we suggest that this facility be in active use by at least three separate CMS ML initiatives at the end of this prototype phase. By demonstrating the effectiveness of this facility for a diverse set of use-cases, we show that it will be a useful asset to the CMS, and a tool that can be expanded to as-yet unknown target applications