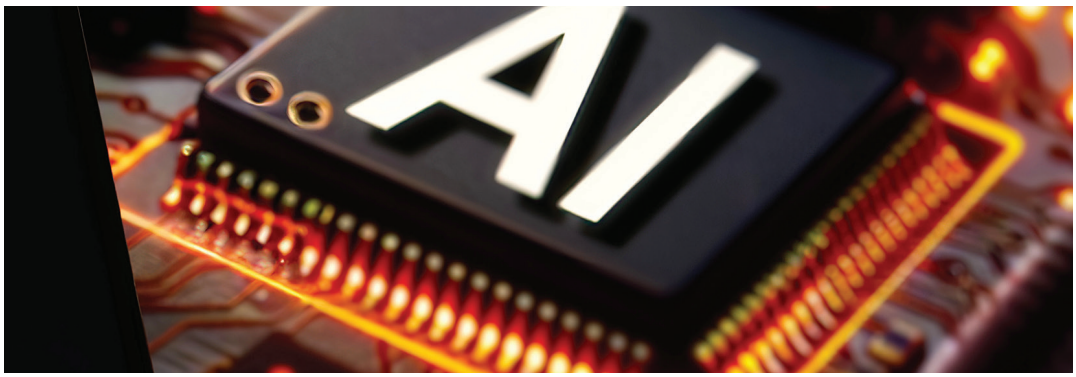
TECHNICAL DOCUMENT

USING ALTAIR® PBS PROFESSIONAL® WITH NVIDIA NCCL FOR AI WORKLOADS

The worlds of high-performance computing (HPC) and artificial intelligence (AI) are colliding. Traditional HPC applications embrace more machine learning techniques to augment existing algorithms, and digital twin drives yet more hybrid AI-simulation workloads. On the pure AI side, workloads are scaling, models are getting larger, and it's increasingly important to apply HPC-style resource management to the compute environment, whether it's on-premises or in the cloud.



Altair® PBS Professional® can be leveraged for workflow management alongside AI workflow tools and integrated with distributed communication technologies such as the message passing interface (MPI) standard from the HPC world and the NVIDIA Collective Communication Library (NCCL) from the AI space.

What's the Difference Between MPI and NCCL?

Conceptually, NCCL is similar to MPI in that they're both used for distributed programming. MPI and various popular implementations of the standard have come out of the HPC world. It's the most common way of running applications across dozens, or even thousands, of physical machines. The MPI standard includes methods for launching processes across different machines and managing communication between them.

NCCL implements multi-GPU and multi-node communication primitives optimized for NVIDIA GPUs and networking. With NCCL users can communicate directly between GPU devices instead of sending data via the CPU. While both traditional HPC and AI/machine learning applications both use GPUs heavily, the NCCL has been developed since the rise of large machine learning models and so has a more AI/machine learning-centric design.

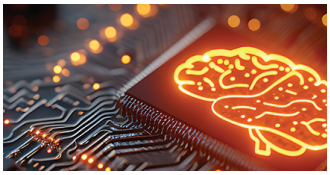
From the API of NCCL, there are various distributed computing concepts similar to MPI, such as scatter, broadcast, gather, etc. There are subtle differences between the communication protocols regarding allocation of ranks and other concepts. At the highest level, MPI should be used for CPU-to-CPU communication and NCCL should be used for GPU-to-GPU communication.

Should You Use MPI or NCCL for AI/Machine Learning Workloads?

Most AI/machine learning workloads use the GPU heavily. When partitioning a model across multiple GPUs, you should use NCCL or a similar library to efficiently communicate between GPUs. Although you could move the data via the CPU, that will incur unnecessary performance penalties. To move data between GPUs, the library should be used to directly move data between devices. This will take advantage of the high-speed PCIe and NVLink interconnect within nodes.

Can You Use NCCL to Manage GPUs Across Multiple Machines?

Many models are now so large it's necessary to partition them across not just multiple GPU devices, but also multiple machines. NCCL integrates with MPI so it may be used across multiple machines. Both NCCL and MPI must first be installed on the machines, then the applications can be integrated with both to take advantage of communication between CPUs and between GPUs across multiple machines. Most orchestration frameworks and AI workflow tools integrate with both.



When partitioning a model across multiple GPUs, use NCCL or a similar library to efficiently communicate between GPUs.

Using NCCL with PBS Professional on One Host

When running a workload on one machine, PBS Professional will simply allocate the requested resources, which will be visible to the workload. So if an AI application needs two GPUs on one host, you can simply request two GPUs on one host to run the application.

```
#PBS -l select=1: ngpus=2
```

The statement above requests two GPUs in one chunk, which will run on one host. Both GPUs will be available to the job, so NCCL can be used to optimize communication between the two GPUs.

Running NCCL on Multiple GPUs in Different NUMA Nodes

Compute nodes have grown in complexity over the years with more and more compute, memory, and acceleration packed into a single machine. It's now common for a single host to have multiple GPU and CPU sockets with a hierarchy of connectivity between the devices, arranged in multiple NUMA nodes. To ensure workloads are placed with CPU, GPU, and memory all local to each other on the same PCIe bus, PBS Professional can divide compute nodes into vnodes based on NUMA topology. In this way, PBS Professional can guarantee smaller jobs are optimally placed on resources that are topologically close to one another when using part of a host, but with the flexibility of allocating a whole host to a larger job if needed.

To request resources in multiple vnodes, you'll request multiple PBS Professional chunks.
For example, the following directives ask for two chunks, each with two GPUs, making four GPUs total. The statement place= pack:exclhost instructs PBS Professional to allocate vnodes on the same host and not allow any other job on that host. To allow other jobs to take unused resources on the vnodes such as spare CPUs, you can use shared instead of exclhost. To allow unused vnodes on that host to be used by other jobs, use excl instead.

```
#PBS -l select=2: ncpus=4:ngpus=2:mem=300g

#PBS -l place=scatter:exclhost
```

The job will be assigned to a whole host that will have at least four GPUs. The GPUs may be arranged as one vnode or as two vnodes, each with two or more GPUs.

Using NCCL with PBS Professional and MPI to Connect GPUs Across Multiple Compute Nodes

When distributing a large model across multiple machines, it's necessary to use both MPI and NCCL together to manage the CPU-to-CPU communication, GPU-to-GPU communication, process launch and management, and configuration of communication between machines. NCCL and PBS Professional are both integrated with MPI, so you can use the combination of the three technologies. You should install MPI that has been configured for use with PBS Professional in the usual way. For example, for OpenMPI 5.x or later compile it with the following:

```
./configure --with-pbs --with-tm=$PBS_EXEC
```

Then set up your application or AI workflow tool to use MPI and NCCL before launching your AI application or workflow tool through PBS Professional.

Note: Some distributed AI frameworks, such as Pytorch, are integrated with Gloo as well as MPI for CPU-to-CPU communication. Gloo uses MPI to handle the launch of MPI processes on each machine even when Gloo replaces the inter-device communication layers in MPU or NCCL, so integration with PBS Professional is the same.

Find Out More:
Request a Demo