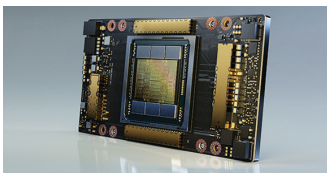
ALTAIR® PBS PROFESSIONAL® SUPPORT FOR NVIDIA® DGX™ SYSTEMS

Developed to meet the needs of demanding high-performance computing (HPC), artificial intelligence (AI), and analytics workloads, NVIDIA® DGX™ systems are built on the revolutionary NVIDIA A100 and V100 Tensor Core GPU platforms with GPU-optimized software and tools including workload management and job scheduling by the industry-leading Altair® PBS Professional® workload orchestration solution.

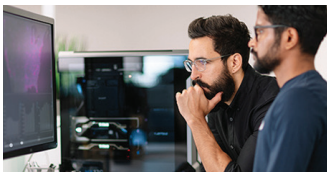
About NVIDIA DGX Systems

NVIDIA DGX systems are a family of products from NVIDIA purpose-built for deep learning applications. The NVIDIA DGX family is comprised of the NVIDIA DGX Station™, NVIDIA DGX-1™ and DGX-2™ rackmount servers, and the newer NVIDIA DGX™ A100 system.



NVIDIA A100 GPUs deliver unprecedented acceleration

In AI data centers, managing distributed GPU-powered machine learning frameworks is a central challenge. Data scientists run diverse workloads ranging from data preparation to model training to model validation to inference. Workloads need to run quickly, use resources efficiently, and be deployed considering factors such as CPU and GPU architecture, memory, cache, bus topologies, and NVIDIA interconnect and network switch topologies.

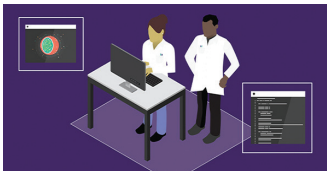


Purpose-built to meet the needs of demanding HPC, AI, and analytics workloads

The latest generation of NVIDIA DGX A100 systems integrate 8 NVIDIA A100 Tensor Core GPUs with an NVIDIA NVLink™ powered NVSwitch™ fabric. When configured with 80GB A100 GPUs, these systems deliver over 3X the performance of an NVIDIA DGX-2 system based on the standard deep learning recommendation model (DLRM) for PyTorch benchmark. NVIDIA DGX A100 systems are 6U servers that can be deployed individually or as part of the NVIDIA DGX POD or DGX SuperPOD reference architecture. Each DGX SuperPod cluster has 140 x DGX A100 systems for a total of 1,120 GPUs. DGX multi-system configurations employ NVIDIA InfiniBand switched fabric with 8 x 200 Gb/s connections per DGX A100 server.

About PBS Professional

PBS Professional is a fast, powerful workload manager designed to improve productivity, optimize utilization and efficiency, and simplify administration for clusters, clouds, and supercomputers — from the biggest HPC workloads to millions of small, high-throughput jobs. PBS Professional automates job scheduling, management, monitoring, and reporting, and it's the trusted solution for complex Top500 systems as well as smaller clusters. Cloud bursting to and between your favorite providers is easier than ever with an intuitive bursting GUI built right in. PBS Professional also delivers a workload simulator that makes it easy to understand job behavior and the effects of policy changes, plus enterprise-wide allocation and budget management capabilities.



Solve bigger, more complex data science problems faster

Architecture and Components – PBS Professional consists of two major component types: user-level commands and system daemons/services. Each is briefly described here to help you understand how the pieces fit together.

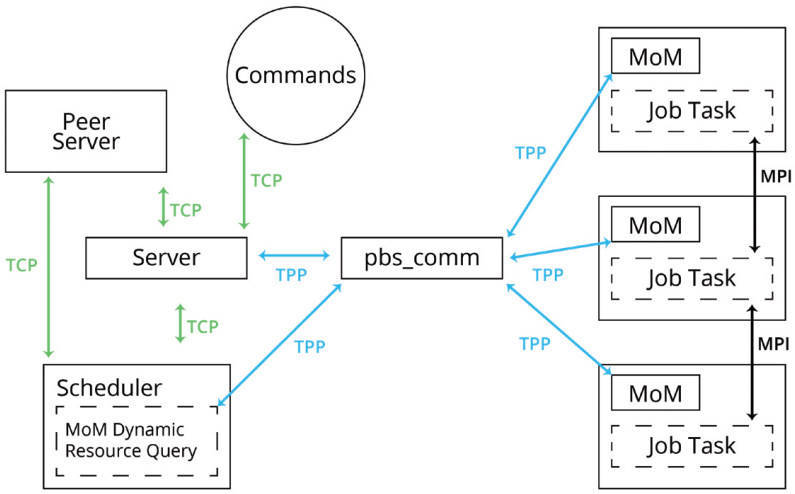
Commands – PBS Professional supplies both command-line programs that are POSIX 1003.2d-conforming and a graphical interface. These are used to submit, monitor, modify, and delete jobs. These client commands can be installed on any system type supported by PBS Professional and do not require local presence of any of the other PBS Professional components. There are three command classifications — user commands, which any authorized user can use; operator commands; and manager (or administrator) commands. Operator and manager commands require special privilege (like root access) to use.

Server – The server daemon/service, pbs_server, is the central focus of PBS Professional. All commands and other daemons/services communicate with the server via an Internet Protocol (IP) network. The server's main function is to provide basic batch services such as receiving/creating a batch job, modifying the job, and passing the job to the execution node. Normally, there is one server managing given set of resources.

Execution Node (MoM) – The execution node daemon/service, pbs_mom, is responsible for instantiating and monitoring the job process(es) — creating a new session for each job and gathering information about that job's resource usage. The pbs_mom service is informally called MoM, as it is a mother of all processes for that job. If the job requires multiple execution nodes, then Mother Superior is the MoM on the head or first host of a multi-host job. Mother Superior controls the job, communicates with the server, and controls and consolidates resource usage information.

Scheduler – The job scheduler daemon/service, pbs_sched, implements the site's scheduling policy, controlling when each job is run and on which resource. The scheduler may communicate with the various MoMs to query the state of system resources and with the server for availability of jobs to execute.

Communication Daemon – The communication daemon/service handles communication between the other PBS daemons. The PBS server, scheduler, and MoM daemons communicate with each other using TPP through the communication daemon, pbs_comm, except for scheduler-server and server-server communication, which uses TCP. The server, scheduler, and MoMs are communication endpoints, connected by one or more pbs_comm daemons. The following figure illustrates communication within a PBS complex using TPP.



Communication daemons are connected to each other. If there are multiple pbs_comms and two endpoints on different pbs_comms transmit data, communication between endpoints goes from the first endpoint to the endpoint's configured pbs_comm daemon, then to the pbs_comm configured for the receiving endpoint, and finally to the receiving endpoint.

The NVIDIA DGX A100 system includes 8 GPUs, 6 NVIDIA NVSwitches, 10 Mellanox network interfaces, dual 64-core AMD CPUs, 2TB system memory, and 30TB Gen4 NVMe SSD

Obtaining PBS Professional

Visit the Altair website to obtain a copy of PBS Professional or request a free software trial.

Configuring PBS Professional With GPUs

PBS Professional treats GPUs/GPU instances like a consumable resource (first-class resource) and allows users to request them in integer units (e.g., 1, 2, 3, etc.). PBS Professional can automatically detect the GPUs/GPU instances on the DGX and isolate the GPU(s)/GPU instance(s) for the job. A very quick overview is below. PBS Professional schedules and allocates each GPU instance equally, regardless of instance size.

IMPORTANT: The pbs_cgroup hook restricts memory usage on shared nodes by default so that a user doesn't cause swapping with other user or system processes. On Ubuntu systems this is configurable via the file /etc/default/grub.

```
GRUB_CMDLINE_LINUX="cgroup_enable=memory swapaccount=1"
```

As root on the PBS server:

1. Export the pbs_cgroups configuration file.

```
qmgr -c "export hook pbs_cgroups application/x-config default" > pbs_cgroups.json
```

2. Update the pbs_cgroups.json configuration file to enable the device's subsystem and add the nvidia-uvm parameter. Below is an example of the update (note the commas).

NOTE: Make sure that the "allow" section excludes read and write access for the 195 major number ("c 195:* m"), which is what all NVIDIA devices use. Preserve mknod ("m") access, since other software such as the container hook may need "m" access. Device code numbers can change from system to system/distro to distort, so, refer to /dev/pts to add other devices if necessary.

```
"devices" : {
  "enabled" : true,
  "exclude_hosts" : [],
  "exclude_vntypes" : [],
  "allow" : [
    "b *:* m",
    "b 7:* rwm",
    "c *:* m",
    "c 195:* m",
    "c 136:* rwm",
    ["infiniband/rdma_cm","rwm"],
    ["fuse","rwm"],
    ["net/tun","rwm"],
    ["tty","rwm"],
    ["ptmx","rwm"],
    ["console","rwm"],
    ["null","rwm"],
    ["zero","rwm"],
    ["full","rwm"],
    ["random","rwm"],
    ["urandom","rwm"],
    ["cpu/0/cpuid","rwm","*"],
    ["nvidia-modeset", "rwm"],
    ["nvidia-uvm", "rwm"],
    ["nvidia-uvm-tools", "rwm"],
    ["nvidiactl", "rwm"] ]
},
```


3. Import the pbs_cgroup configuration file.

```
qmgr -c "import hook pbs_cgroups application/x-config default pbs_cgroups.json"
```

4. Enable the pbs_cgroup hook.

```
qmgr -c "set hook pbs_cgroups enabled = true"
```

5. Update the resources: parameter of the PBS scheduler configuration file to include ngpus.

```
resources: "ncpus, mem, arch, host, vnode, aoe, eoe, ngpus"
```

6. Update the flag of ngpus resource.

```
qmgr -c "set resource ngpus flag=nh"
```

Finally, as root on the DGX system(s), HUP or restart the pbs_mom daemon. This action will force pbs_mom to resend the hardware inventory with the PBS Professional server.

```
pkill -HUP pbs_mom
```
or
```
systemctl stop pbs ; systemctl start pbs
```

To run a job requesting GPU resources, the user will request the ngpus resource via qsub. For example, to run a job requiring two GPUs or GPU instances on a single DGX system the following qsub command can be used.

```
$ cat job.sh
#!/bin/sh
grep ^ID= /etc/os-release
nvidia-smi -L
echo $CUDA_VISIBLE_DEVICES

$ qsub -l select=1:ngpus=2 job.sh
62.server
$ cat job.sh.o62
ID="centos"
GPU 0: NVIDIA A100-PCIE-40GB (UUID: GPU-ade9b969-b95f-7fc1-9075-860078a3b0b7)
  MIG 1g.5gb      Device  0: (UUID: MIG-1587894d-e0db-5f61-8b47-9aac9ed49baf)
  MIG 1g.5gb      Device  1: (UUID: MIG-b77baeb1-8280-5af4-be1c-0ad656409721)
MIG-GPU-ade9b969-b95f-7fc1-9075-860078a3b0b7/7/0,MIG-GPU-ade9b969-b95f-7fc1-9075-
860078a3b0b7/8/0
```

PBS Professional provides the same level of GPU scheduling and isolation for container jobs. Please refer to the PBS Professional Administrator's Guide for complete container support setup instructions and capabilities. This example uses Singularity; Docker is also supported.

```
$ qsub -l select=1:ngpus=2 -lcontainer_image=ubuntu job.sh
63.server

$ cat job.sh.o63
ID=ubuntu
GPU 0: NVIDIA A100-PCIE-40GB (UUID: GPU-ade9b969-b95f-7fc1-9075-860078a3b0b7)
  MIG 1g.5gb      Device  0: (UUID: MIG-1587894d-e0db-5f61-8b47-9aac9ed49baf)
  MIG 1g.5gb      Device  1: (UUID: MIG-b77baeb1-8280-5af4-be1c-0ad656409721)
MIG-GPU-ade9b969-b95f-7fc1-9075-860078a3b0b7/7/0,MIG-GPU-ade9b969-b95f-7fc1-9075-
860078a3b0b7/8/0
```

In each of these examples PBS Professional provides device isolation to make sure job submitters only use the GPU resources they requested, rather than everything that exists on the execution hist.

NVIDIA DGX A100 systems can be deployed individually or as part of the NVIDIA DGX SuperPOD Solution for Enterprise

Reference

For additional information about managing and using PBS Professional, please consult the PBS Professional documentation (Installation & Update Guide, Administrator's Guide, and User's Guide).

For more information or technical support, please visiit **altair.com/contact-us**.

NVIDIA DGX systems were designed to meet the demanding needs of HPC, AI, and analytics workloads