Scheduling Jobs onto Intel® Xeon Phi™ using PBS Professional®

Scott Suchyta[1]
[1] *Altair Engineering Inc., 1820 Big Beaver Road, Troy, MI 48083, USA*

Abstract

As new hardware and technology arrives, it is imperative that workload managers and job schedulers scale and support the new technology. By working closely with Intel, Altair has ensured that PBS Professional supports the Intel® Xeon Phi™ (previously known as many-integrated cores, or MIC) architecture targeting high performance computing applications. This collaboration will allow users to take full advantage of the performance and power of this new Intel architecture immediately without relying on any workarounds. This document will review how to configure PBS Professional in an Intel Xeon Phi environment and provide tips. As early adopters (i.e., software and hardware vendors) continue to explore the new technology, we are looking for feedback on how PBS Professional can improve support for scheduling jobs onto the Intel Xeon Phi.

Introduction to PBS Professional

PBS Professional is the professional version of the Portable Batch System (PBS), a flexible workload management solution, originally developed to manage aerospace computing resources at NASA. PBS Professional has since become the leader in supercomputer workload management and the de facto standard on Linux clusters.

Today, growing enterprises often support hundreds of users running thousands of jobs across many different types of machines in diverse geographical locations. In this distributed heterogeneous environment, it can be extremely difficult for administrators to collect detailed, accurate usage data, or to set system-wide resource priorities. As a result, many computing resources are under-utilized, while others are over-utilized. At the same time, users are confronted with an ever-expanding array of operating systems and platforms. Each year, scientists, engineers, designers and analysts must waste countless hours learning the nuances of different computing environments, rather than

focusing on their primary goals. PBS Professional addresses these problems for computing-intensive industries such as science, engineering, finance and entertainment. Now you can use the power of PBS Professional to better control your computing resources. This allows you to unlock the potential in the valuable assets you already have, while at the same time reducing the load on system administrators and operators, freeing them to focus on other activities. PBS Professional can help you effectively manage growth by tracking real usage levels across your systems and allowing you to target future purchases to your needs.

PBS Professional consists of two major component types: commands and daemons / services. A brief description of each is given here to help you understand how the pieces fit together.

Commands – PBS Professional supplies both a graphical interface and a set of POSIX 1003.2d-conforming command-line programs. These are used to submit, monitor, modify and delete jobs. These client commands can be installed on any system type supported by PBS and do not require local presence of any of the other components of PBS.

There are three command classifications: commands available to any authorized user, commands requiring special PBS privilege, and administrator commands. Administrator commands require root access or the equivalent 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 the basic batch services such as receiving/creating a batch job, modifying the job, and passing the job to the execution node. One server manages the machines and jobs in a PBS complex; a secondary server may be configured to handle failover.

Execution Node Manager (MOM) – The MOM is the daemon/service, which actually places the job into execution. The pbs_mom daemon is informally called MOM, as it is the mother of all processes for jobs. (MOM is a reverse-engineered acronym that stands for Machine Oriented Mini-server). MOM places a job into execution when it receives a

copy of the job from the Server. MOM creates a new session for each job and gathers information about the resource usage of that job. MOM also has the responsibility for communicating with all MOMs assigned to the job and returning the job's output to the user when directed to do so by the Server. One MOM runs on each execution host.

Scheduler – The job scheduler daemon/service, pbs_sched, implements the site's scheduling policy, controlling when each job is run and on which resources. 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.

Introduction to Intel® Xeon Phi™

Relative to the multi-core Intel Xeon processors, Intel Xeon Phi Architecture has many more smaller cores, many more hardware threads, and wider vector units. This is ideal for achieving higher aggregate performance for highly parallel applications.

As developers embrace high degrees of parallelism (instruction, data, task, vector, thread, cluster, etc.), important and popular programming models for Intel architecture processors extend to Intel Xeon Phi architecture without rethinking the entire problem. The same techniques that deliver optimal performance on Intel processors – scaling applications to cores and threads, blocking data for hierarchical memory and caches, and effective use of SIMD – also apply to maximizing performance on Intel Xeon Phi architecture.

With greater reuse of parallel CPU code, software companies and IT departments benefit from creating and maintaining a single code base binary and not having to retrain developers on proprietary programming models associated with accelerators.

Configuring Intel Xeon Phi for use in PBS Professional

PBS supports both basic and advanced Intel Xeon Phi scheduling. Basic scheduling consists of prioritizing jobs based on site policies, controlling access to nodes with Intel Xeon Phi cards, ensuring that Intel Xeon Phi is not over-subscribed, and tracking use of Intel Xeon Phi in accounting logs. Configuring PBS to perform basic scheduling of Intel

Xeon Phi is relatively simple, and only requires defining and configuring a single custom resource to represent the number of Intel Xeon Phi cards on each node. Approach 1 describes basic Intel Xeon Phi scheduling.

Although basic Intel Xeon Phi scheduling will meet the needs of 95% of customers, PBS also supports more advanced Intel Xeon Phi scheduling: the ability for a job to separately allocate (request and/or identify) each individual Intel Xeon Phi device on a node. This capability is useful for sharing a single node among multiple jobs, where each job requires its own Intel Xeon Phi's. In this case, both PBS and the applications themselves must support individually allocating a node's Intel Xeon Phi. This more advanced scheduling requires defining a "PBS vnode" for each Intel Xeon Phi. Approach 2 describes advanced Intel Xeon Phi scheduling.

This section covers both approaches on how to configure Intel Xeon Phi in PBS Professional, and also compares the pros and cons of each approach. Note that in order to schedule jobs onto Intel Xeon Phi, the administrator must manually configure Intel Xeon Phi cards as resources. Note also that PBS Professional allocates Intel Xeon Phi, but doesn't bind jobs to any particular Intel Xeon Phi; the application itself (or the Intel library) is responsible for the actual binding.

Altair will be making a configuration tool available at Xeon Phi launch in 2013 which automates the configuration for Xeon Phi within PBS Professional.

Approach 1 – Basic Intel Xeon Phi Scheduling

Configure all Intel Xeon Phi devices as a single custom resource

The administrator adds only one custom resource in this approach. PBS Professional treats all Intel Xeon Phi devices with equal priority (similar to the built-in ncpus PBS resource) and doesn't bind jobs to any particular Intel Xeon Phi.

Steps to configure using single custom resource approach:

Assuming there are two execution hosts, nodeA and nodeB, present in the cluster, and each execution host has 2 Intel Xeon Phi devices, the administrator configures a single consumable custom resource "nmics" to represent all Intel Xeon Phi devices on the execution hosts.

The administrator then configures the nmics custom resource in the following way:

1. Stop the PBS Professional server and scheduler. On the server's host, type:

```
/etc/init.d/pbs stop
```

2. Edit $PBS_HOME/server_priv/resourcedef to add the following line:

```
nmics type=long flag=nh
```

3. Edit $PBS_HOME/sched_priv/sched_config to add nmics to the list of scheduling resources:

```
resources: "ncpus, mem, arch, host, vnode, nmics"
```

4. Restart the PBS Professional server and scheduler. On the server's host, type:

```
/etc/init.d/pbs start
```

5. Add the number of Intel Xeon Phi devices available (in our example, this number is 2) to each execution host in the cluster via qmgr:

```
qmgr -c "set node nodeA resources_available.nmics=2"
qmgr -c "set node nodeB resources_available.nmics=2"
```

6. Submit a one-node job (e.g., “my_mic_job”) on 2 Intel Xeon Phi devices in the following manner:

```
qsub -lselect=1:ncpus=1:nmics=2 my_mic_job
```

The administrator can find the number of Intel Xeon Phi devices available/assigned on the execution hosts via the ‘pbsnodes’ command.

Approach 2 – Advanced Intel Xeon Phi Scheduling

Configure Intel Xeon Phi devices as vnodes

In this approach, the administrator configures each Intel Xeon Phi device in its own vnode, including the Intel Xeon Phi device number.

Steps to configure Intel Xeon Phi as virtual nodes with device number:

1. Stop the PBS Professional server and scheduler. On the server’s host, type:

```
/etc/init.d/pbs stop
```

2. Edit $PBS_HOME/server_priv/resourcedef to add two new custom resources, nmics and mic_id:

```
nmics type=long flag=nh
mic_id type=string flag=h
```

3. Edit $PBS_HOME/sched_priv/sched_config to add nmics and mic_id to the list of scheduling resources:

```
resources: “ncpus, mem, arch, host, vnode, nmics, mic_id”
```

4. Restart PBS Professional services. On the server’s host, type:

```
/etc/init.d/pbs start
```

5. Create a vnode configuration for each execution host where Intel Xeon Phi devices are present. In this example, we create a script named “nodeA-vnodes” for “nodeA” which has 4 CPUs, 2 Intel Xeon Phi’s, and 16 GB of memory:

```
$configversion 2
nodeA: resources_available.ncpus = 0
nodeA: resources_available.mem = 0
nodeA[0]: resources_available.ncpus = 2
nodeA[0] : resources_available.mem = 8gb
nodeA[0] : resources_available.nmics = 1
nodeA[0] : resources_available.mic_id = mic0
nodeA[0] : sharing = default_exclusive
nodeA[1] : resources_available.ncpus = 2
nodeA[1] : resources_available.mem = 8gb
nodeA[1] : resources_available.nmics = 1
nodeA[1] : resources_available.mic_id = mic1
nodeA[1]: sharing = default_exclusive
```

6. Add vnode configuration information to PBS Professional in the following manner (for each node with Intel Xeon Phi):

```
$PBS_EXEC/sbin/pbs_mom -s insert nodeA-vnodes nodeA-vnodes
```

7. Signal each pbs_mom to reread the configuration files:

```
kill -HUP <pbs_mom PID>
```

8. Submit job (e.g., “my_mic_job”) requesting one node with one Intel Xeon Phi:

```
qsub -lselect=1:ncpus=1:nmics=1 my_mic_job
```

The PBS Scheduler looks for a vnode with an available Intel Xeon Phi card and assigns that vnode to the job. Since there is a 1-1 correspondence between Intel Xeon Phi and vnodes, the job can now ask PBS which vnode it was given, and determine the mic_id of that vnode. Finally, the application can use the appropriate Intel libraries to bind the process to the allocated Intel Xeon Phi device.

Alternatively, one can submit a job requesting a particular mic_id as well. The following requests 4 nodes, each with Intel Xeon Phi id 0:

```
qsub -lselect=4:ncpus=1:mic_id=mic0 my_mic_job
```

This provides a high degree of control for users and administrators, allowing them to run their applications based on the devices for which their applications are programmed.

Comparison of Basic and Advanced Intel Xeon Phi Scheduling

	Approach 1	Approach 2
Configuration & Administration	Easy and centralized. Must make configuration changes only in PBS Server and Scheduler.	More complicated and distributed. Must make configuration changes on all nodes with Intel Xeon Phi.
Granularity	All Intel Xeon Phi devices are given equal priority. The user does not specify Intel Xeon Phi on which to run job.	Users can select Intel Xeon Phi devices by specifying their device numbers.
Number of custom resources configured	Only one	Two, plus virtual nodes on all MOMs based on number of Intel Xeon Phi devices
Uses	Good choice when jobs do not share nodes (i.e., only one Intel Xeon Phi job at a time is run on any given node, exclusively).	Good choice when sharing of nodes by multiple jobs at the same time is required, or individual access to Intel Xeon Phi (by device number) is required.

Conclusion

The two approaches suggested in this paper provide the techniques necessary to configure and schedule jobs onto Intel Xeon Phi using PBS Professional. This document provides detailed information on how to make efficient use of hundreds of computing cores on Intel Xeon Phi with PBS Professional. Given that the Intel Xeon Phi is new and emerging technology, and as it continues to be integrated into hardware and software vendor solutions, we may need to revisit the approach.

In closing it should be noted that users could check the number of Intel Xeon Phi devices assigned and requested for the job in the PBS accounting log file, or via PBS Professional commands such as “pbsnodes”. It should also be noted that none of the approaches specified in this paper tries to pin tasks to particular Intel Xeon Phi devices. As with normal CPU-based jobs, assignment of tasks to cores is handled by the operating system services.