GPU Scheduling at NICS: The Good, The Bad, and The Weird

Troy Baer, Stephen McNally, and John Wynkoop

HPC System Administrators

National Institute for Computational Sciences

University of Tennessee

Overview

- **Introduction to NICS**
- **Keeneland Initial Delivery (KID)**
  - **The Good, the Bad, and the Weird**
  - **Status & Wishlist**
- **Nautilus**
  - **The Good, the Bad, and the Weird**
  - **Status & Wishlist**
- **Conclusions and Future Work**

National Institute for Computational Sciences

University of Tennessee & ORNL partnership

- **NICS is a NSF HPC center**
  - **Operated by UT, located at ORNL**
  - **XSEDE (formerly Teragrid) Service Provider**
- **Systems**
  - **Kraken (Cray XT5, 112,984 cores)**
  - **Nautilus (SGI UV, 1,024 cores)**
  - **Keeneland initial delivery system (HP GPU cluster, 1,440 cores + 360 GPUs) in conjunction with Georgia Tech**

Keeneland - Initial Delivery (KID)

CPU	2 x Intel Xeon X5660 2.80 Ghz (Westmere)
GPU	3 x Nvidia Tesla M2070 6GB (Fermi)
Interconnect	Infiniband QDR (single rail)
# Compute Nodes	120
Total CPU cores	1,440
Total GPUs / cores	360 / 161,280
# Login Nodes	4
# Management Nodes	2

KID Scheduling Overview

- **Priority scheme giving higher priority to larger jobs**
- **Fair share targets of ~10% per allocated project**
- **Development reservations Mon, Wed, Thur, Fri 9AM – 9PM Eastern to support jobs requesting less than 4hr wall time and less than 48 nodes.**
- **Weekly maintenance reservations on Tuesdays followed by capability jobs (ad-hoc reservations are used for these).**
- **Several other reservations for software/hardware testing on a subset of nodes.**

TORQUE/Moab on KID – The Good

- **Initial experiences were great, albeit without any real GPU support.**
- **Started with Moab 6.0.2 and Torque 2.5.7, and these both worked well for quite some time.**
- **Customization and flexibility in these releases were great because everything worked as expected.**

TORQUE/Moab on KID – The Bad

- **TORQUE version 2.5.[0-7] only supported Nvidia driver versions 260.x.x and 270.x.x.**
- **CUDA 4.1 requires Nvidia driver 285.x.x, necessitating an upgrade to TORQUE 2.5.8. This version had necessary GPU fixes, but `pbs_server` did not honor Moab-assigned node lists or user specified GPU compute modes.**
  - **This resulted in numerous user jobs being run on nodes where they should not have been placed.**

TORQUE/Moab on KID – The Bad (con't.)

- **Upgraded to 2.5.9 which was even less successful as `pbs_moms` (and sometimes `pbs_server`) segfaulted in such a way that no information could be obtained from them.**
- **TORQUE 2.5.10 appeared to be mostly the same as 2.5.8 plus bug fixes from 2.5.9 eliminated most segfault problems, but it still had problems with Moab-assigned node lists and compute mode declarations.**

TORQUE/Moab on KID – The Weird

- **For some reason, the Nvidia GPU driver support fixes in TORQUE 2.5.[8-10] and greater caused `pbs_server` to ignore the node lists assigned to jobs by Moab.**
- **TORQUE 2.5.[8-10] also ignored user-requested GPU compute modes. These were treated as node features/properties instead of GPU compute modes.**
- **Both of these issues were largely resolved by TORQUE 2.5.11, but..**

TORQUE/Moab on KID – The Weird (con't.)

- **The “default” GPU compute mode used by TORQUE doesn’t map to the “default” compute mode per the NVIDIA driver documentation.**
  - **Instead, it maps to “exclusive_thread”, which is rarely what our users want or expect.**
- **Setting the compute mode via a prologue or epilogue script works, sort of…**
  - **TORQUE tries to cache the state of the GPU compute mode**
  - **In practice, its reset the compute mode to “exclusive_thread” before every *other* job.**

KID – Current Status

- **Running TORQUE 2.5.11 as it appears to have fixed all of TORQUE GPU issues and now handles node assignments correctly and allows users to specify compute modes (even though “default” isn’t default…)**
- **Running Moab 6.1.5**
- **Running the CUDA 4.1 NVIDIA kernel module**
- **Everything with the exception of reservation profile ACLs are working, which is an unrelated issue with Moab 6.1.x.**

KID – Wishlist

- **Fix the GPU compute mode requested by users to use same terminology as NVIDIA driver. Default should be the default!**
  - **Ideally, this should be configurable on a per installation basis.**
- **Add support to record GPU usage in the accounting logs the same as CPU time, memory, etc. are tracked now.**

Notes

- **Users can work around the compute mode default by requesting “shared”**

```
troy@kidlogin1 ~]$ qsub -l nodes=3:gpus=1 –I
[troy@kid025 ~]$ nvidia-smi -q -d compute
==============NVSMI LOG==============
Timestamp                           : Wed Apr  4 08:14:46 2012
Driver Version                      : 285.05.32
Attached GPUs                       : 3
GPU 0000:06:00.0
    Compute Mode                    : Exclusive_Thread
GPU 0000:14:00.0
    Compute Mode                    : Exclusive_Thread
GPU 0000:11:00.0
    Compute Mode                    : Exclusive_Thread
[troy@kid025 ~]$
```

```
troy@kidlogin1 ~]$ qsub -l nodes=1:gpus=3:shared –I
[troy@kid025 ~]$ nvidia-smi -q -d compute
==============NVSMI LOG==============
Timestamp                           : Wed Apr  4 08:14:46 2012
Driver Version                      : 285.05.32
Attached GPUs                       : 3
GPU 0000:06:00.0
    Compute Mode                    : Default
GPU 0000:14:00.0
    Compute Mode                    : Default
GPU 0000:11:00.0
    Compute Mode                    : Default
[troy@kid025 ~]$
```

- **Thanks to Andy Regan (http://andyregan.net/blog/archives/522) for documenting this behavior.**

Nautilus – Remote Data Analysis and Visualization

- **SGI UV1000**
  - **1,024 cores, 4 TB memory, and 8 GPUs in a single host**
  - **64 blades, each with two 8-core Nehalem-EX processors and 64 GB of memory (128 NUMA nodes total)**
  - **4 blades have Tesla interface card connected to one of two Nvidia Tesla S2050s**

Nautilus Scheduling Overview

- **Two main queues**
  - **`computation` (lower priority, preemptee)**
  - **`analysis` (higher priority, preemptor)**
- **Weekly maintenance reservations on Tuesdays**
- **“Debug” reservation on 4 NUMA nodes to improve turnaround for small, short jobs**
- **Reservations to keep jobs off NUMA nodes with 10GigE and GPUs unless there's nowhere else for them to run**

TORQUE/Moab on Nautilus – The Good

- **After some initial teething pains, the NUMA branch of TORQUE became solid enough to use in production in early 2010.**
- **However...**

TORQUE/Moab on Nautilus – The Bad

- **It took significantly longer for the Linux kernel and the Nvidia driver to mature to the point where Nvidia GPUs would work reliably on a large NUMA system.**
  - **Originally Nautilus was to have 16 GPUs, but this had to be scaled back to 8 for driver reasons**
- **Once the driver issues were resolved, TORQUE 3.0.x had some issues with GPUs on NUMA systems.**
  - **No GPU equivalent to “ncpus” at first**
  - **Syntax for assigning GPUs to NUMA nodes clearly not designed for a system with >100 NUMA nodes.**

TORQUE/Moab on Nautilus – the Weird

- **Moab had its own interesting issues with GPUs on NUMA**
  - **GPU requests interpreted as a per-task request rather than a per-job request**
  - **Effectively impossible to ask for fewer GPUs than cores**
  - **Appears to stem from an implicit assumption that every NUMA node will have GPUs**
  - **Resolving this has been long and painful, and has not been completed**
    - **Requires TORQUE 4.0.x and Moab >7.0**

Nautilus – Current Status

- **Running TORQUE 3.0.3-snap.201107121616**
  - **Final 3.0.3 release doesn't handle gpus=>GRES translation properly**
  - **Have not had opportunity to test 3.0.[45].**
- **Running TORQUE 6.0.4**
- **Waiting on TORQUE 4.0.x and Moab >7.0**

Nautilus – Wishlist

- **`qsub -l ncpus=M,gpus=N` where M>>N should be possible**
  - **Treat GPU request as independent of number tasks (i.e. a multi-req job) on NUMA system**
- **Major GPU use case for Nautilus is hardware-accelerated rendering, not GPGPU**
  - **Start up X server(s) on user's behalf on GPUs in job prologue?**
  - **This is largely a site-specific matter**

Conclusions and Future Work

- **GPU support in TORQUE and Moab has come a long way, but needs to go a bit farther**
  - **TORQUE should handle multiple Nvidia driver versions in the same build**
  - **Usage monitoring**
  - **Both TORQUE and Moab need to handle heterogeneous systems (i.e. some nodes with GPUs and others without) better**
  - **NUMA systems have additional requirements**
- **Leverage Nvidia GPU support code in TORQUE for other accelerator technologies?**
  - **AMD GPU**
  - **Intel MIC**