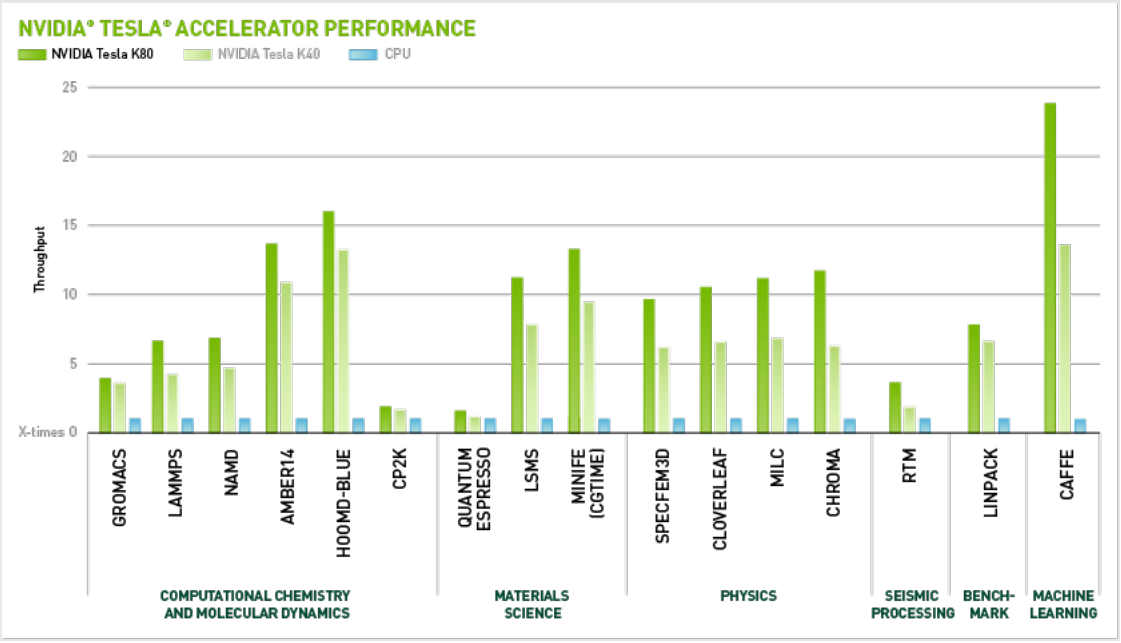
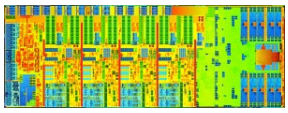
Description of Power8 Nodes Available on Mio (ppc[001-002])

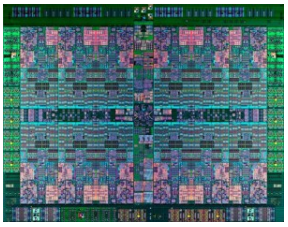
Introduction:

HPC@Mines has released two brand-new IBM Power8 nodes (identified as ppc001 and ppc002) to production, as part of our Mio cluster. These are state-of-the art units; Mines acquired them pre-production and HPC@Mines has readied them for student and faculty use. In fact, our benchmarking and testing efforts gave rise to a fruitful collaboration with IBM; our results have been incorporated by IBM to fine-tune the performance potential of the Power8 nodes. The Power8 models acquired by Mines feature two NVIDIA Tesla K80 GPUs, data-crunching accelerators designed to handle demanding computational tasks, that specifically boost HPC workload performance. The histogram below illustrates potential gains for several applications.



To augment the case for investment in Power8 architecture, following is a performance comparison between IBM Power8 nodes and Intel's current Xeon hardware, considered commensurate for HPC performance. The majority of the processors on Mio are of the x86 variety. To give some perspective on how the two hardware designs stack up, in order that users might engender expectations, following are some textual and graphic comparisons among various components of each.

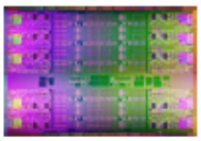


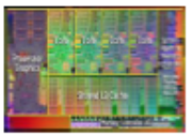


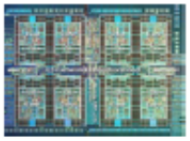
Intel Haswell Processor Die IBM Power8 Processor Die

Source: Xcelerit Blog June 02 2015

Figure 1: Major Differences in Intel Xeon vs. POWER Architectures







	Westmere EX	Sandy Bridge EP	POWER Systems
Clock rates per processor	1.8–2.67 GHz	1.8–3.6 GHz	3.1–4.4 GHz
Symmetric multi-threading per core	1,2	1,2	1, 2, 4
On-chip L3 Cache	30 MB	20 MB	80 MB
Maximum Memory per server	4 TB	1.5 TB	16 TB
Max Threads per server	160	64	1024

More processing power per core

Faster cache intensive workloads

Run more memory intensive workloads

Larger servers for consolidation

Source: IBM Corporation, December, 2013

Comparison of Architectures:
IBM POWER and Intel Xeon

Both Power8 nodes and x86 nodes house microprocessors designed for high-performance-oriented general purpose processing, with each processor built to execute in serial, parallel and compute-intensive environments. Compared to x86 architecture, Power8 architecture aims to deliver higher processor efficiency through enhanced threading capabilities, such as an increased number of threads that can be exploited per clock cycle, augmented memory bandwidth and a higher-capacity on-chip cache. The Power8 also boasts PCIe Gen 3 logic on-chip, with an accelerator processor interface (CAPI) that enables devices and coprocessors to communicate directly and speedily with processors. CAPI connects a custom acceleration engine to the coherent fabric of the POWER8 chip, providing a simple programming paradigm while delivering performance outpacing current I/O attached acceleration engines. In general, users might anticipate memory bandwidth-limited

codes to especially benefit from the Power8 architecture; for FLOPs-limited codes improvement may be noticed, albeit not so starkly. Augmented performance can also be expected to be resource-dependent, with applications consuming large amounts of resources able to exploit the Power8 architecture to greatest advantage.

Table 1: Mines' 10-core Power8: Intel Xeon E52697v3 vs IBM Power8 S822LC

Feature:	Intel Xeon Haswell	IBM Power8 S822LC
1. Processor Speed:	2.6 GHz	2.92 GHz
2. Number of Cores/Socket:	14	10
3. Number of Threads/Core:	2	8
4. Main memory:	256GB DDR4	512GB of 1333MHz DDR3 ECC
5. Number Memory Controllers	1	2
6. On-Chip Cache: L2: L3: L4:	 256KB/core 35MB SRAM None	 512KB/core 8MB/core eDRAM 64MB/processor
7. Memory Bandwidth:	68GB/sec	192GB/sec
8. Input/Output	PCI Gen3, 40 lanes	3 x PCI Gen3, 32 lanes
9. Device Accelerator:	QPI	CAPI
10. Accelerator:	Not Included	NVIDIA Tesla K80 GPU (x2)

Modules:

HPC@Mines is working toward the deployment of a new module system, to simplify the creation of users' compute environments for running applications calculations on BlueM and Mio. We hope to make the module loading process more intuitive and to require less loading of individual modules one at a time. The Power8 nodes are the first with this advantage; thus the procedure to load modules on ppc001 and ppc002 invites some new instructions. And since the Power8 nodes are of a different architecture from the rest of the nodes on Mio, some slight alterations in submitting jobs to the queue will be discussed. Finally, some sample runscripts are included; we strongly advise that you use them as presented at first; some inclusions may seem extraneous that are not; of course you are free to alter them once you understand their function(s).

How to Submit a Job:

1. Log on to mio.
2. Go to your job directory in scratch.
3. Submit job as usual: sbatch runscript.sh. Crucial changes to runscript (see example below):
    - Preamble must have partition set as ppc (-p ppc or - -partition=ppc);
    - Preamble must include 'gres' option: see `http://slurm.schedmd.com/sbatch.html`;
    - Preamble must have 'export' option set to NONE;
    - Preamble must have 'get-user-env' set appropriately: see `http://slurm.schedmd.com/sbatch.html`;
    - IBM's poe must be used as launcher (not srun);
    - IBM's poe requires a host list: see example runscript below;
    - Modules must be loaded in the order displayed in the sample scripts.

Sample Runscripts:

****

Quantum Espresso:

```
#/bin/bash -x
#SBATCH - -job-name="QEpoeIBM"
#SBATCH - -nodes=2
#SBATCH - -ntasks-per-node=4
#SBATCH - -ntasks=8
#SBATCH -p ppc
#SBATCH - -gres=gpu:kepler:4
#SBATCH - -export=NONE
#SBATCH - -get-user-env=10L
#SBATCH - -time=06:00:00

module purge
module load XL
module load IBMpe
module load CUDA
module load QE5.2.0/QE5.2.ppc

export OMP_NUM_THREADS=2
export PHI_DGEMM_SPLIT=0.975
export PHI_ZGEMM_SPLIT=0.975
```

```
export QEGPU_GPU_PER_NODE=4
#default is 2

EXE=${QEROOT}/pw-gpu.x

#launch IBM MPI jobs with poe:
export MP_RESD=poe

#poe requires 'old style' hostlist:
export MP_HOSTFILE=SLURM_JOBID.list
/software/apps/generic/utility/expands > MP_HOSTFILE

export MP_STDINMODE=all

export LABELIO=yes
#per TK

JOBID=`echo SLURM_JOBID`
echo $JOBID
mkdir $JOBID
cd $JOBID

ln -s ../MP_HOSTFILE
ln -s ../si.scf.in
ln -s ../../../pseudo

cat $0 > $JOBID.runscript
printenv > $JOBID.env

echo "running job"
poe $EXE -procs $SLURM_NTASKS -input si.scf.in > $JOBID.siscf.out

unset MP_RESD
unset MP_HOSTFILE
unset MP_LABELIO

echo "job has finished"


**********
```

Notes, Observations and Tips:

1. The line 'ln -s ../../../pseudo' assumes that the PSEUDO directory is located three levels up. This parameter can also be set by environmental variable 'PSEUDO_DIR' (see QE docs);
2. Since running on ppc nodes, make sure to use 'expands' utility in /software/apps/generic/utility;
3. To avoid unbalanced configuration warnings, number of MPI tasks per node must be greater than or equal to, and a multiple of, the gpu 'count' entry for gres;
4. The environmental variable QEGPU_GPU_PER_NODE does not seem to affect output, although I have only tested on very short runs. Its default value is '2', which seems to suffice (meaning one needn't specify a value in the runscript);
5. OMP_NUM_THREADS affects, as usual, the total number of processes. This is reflected in the QE output file, but does not affect the numbers (shown) in the 'SLURM' output file;
6. It seems that time is 'lost', or that discrepancies occur, at the start of the calculation, prior to the first iteration. The time for the iterations to reach convergence tends to be consistent; it's just getting to the start of iteration 1 that varies wildly;
7. The 'count' field of gres determines CUDA_VISIBLE_DEVICES and GPU_DEVICE_ORDINAL environmental variable values.
8. For best performance: run a maximum of two MPI tasks per GPU card.

	IBM Power8	Intel Xeon E5-2697v3	IBM Power8	Intel Xeon E5-2699v3
CPU Cores	8 cores	14 cores	12 cores	18 cores
CPU Threads	64 threads	28 threads	96 threads	36 threads
CPU Freq (max)	3.75GHz	2.6GHz	3.1GHz	2.3GHz
CPU Freq (turbo)	4.1GHz	3.6GHz	3.6GHz	3.6GHz
Superscalar Execution	8 instr per cycle, OOO	8 micro-ops per cycle, OOO	8 instr per cycle, OOO	8 micro-ops per cycle, OOO
L2 Cache	512KB per core	256KB per core	512KB per core	256KB per core
L3 Cache	64MB eDRAM	35MB SRAM	96MB eDRAM	45MB SRAM
L4 Cache	64MB*	None	64MB*	None
DRAM Interface	16x 64-bit DDR3-1600*	4x 64-bit DDR4-2133	16x 64-bit DDR3-1600*	4x 64-bit DDR4-2133
DRAM B/W	205GB/s*	68GB/s	205GB/s*	68GB/s
Glueless SMP	2 sockets	2 sockets	2 sockets	2 sockets
Coherent SMP	12 sockets	2 sockets	12 sockets	2 sockets
PCI Express	32 lanes	40 lanes	32 lanes	40 lanes
SPECint‡	627 int	612 int	670 int	687 int
SPECfp‡	485 fp	427 fp	509 fp	460 fp
IC Process	22nm SOI	22nm FinFET	22nm SOI	22nm FinFET
Power	270W TDP*	145W TDP	270W TDP*	145W TDP
List Price (1ku)	$1,210*†	$2,702	$2,860*†	$4,100†

Linley Group Microprocessor Report December 29, 2014