

Parallel Info

Parallel programming related information

What are some reasons for using the cluster

- access to MPI based parallel programs
- access to larger amounts of memory than 32bit computers offer
- access to the large public domain of scientific computing programs
- access to compilers
- access to large amounts of storage
- access to batch processing for running numerous independent serial jobs
- access to 64bit versions of programs you may already have on your 32bit desktop

What is MPI?

MPI stands for Message Passing Interface. The goal of MPI is to develop a widely used standard for writing message-passing programs.

What installed programs provide a parallel solution?

The following provide **MPI** based solutions: Abaqus, Ansys, Fluent, Mathematica, Matlab, Paraview

The following programs provide thread based parallelism: Comsol, Matlab

The default settings for Matlab are number of threads equals number of cores on a compute node.

When does 64bit computing matter?

When there is a need for memory and storage beyond the 32bit barriers.

Is it possible to run linux 32-bit executables on the cluster?

There is a good chance that it will succeed. But there might be other issues preventing it from running. Try it out...

Where can I find additional information about MPI?

<http://www-unix.mcs.anl.gov/mpi/>

<http://www.nersc.gov/nusers/resources/software/libs/mpi/>

<http://www.faqs.org/faqs/mpi-faq/>

<http://www.redbooks.ibm.com/abstracts/sg245380.html>

<http://www.nccs.gov/user-support/training-education/hpcparallel-computing-links/>

<http://software.intel.com/en-us/multi-core/>

What are some good web based tutorials for MPI?

<http://ci-tutor.ncsa.uiuc.edu/login.php>

<http://www.slac.stanford.edu/~alfw/Parallel.html>

https://computing.llnl.gov/tutorials/parallel_comp/

How do I run a compiled mpi based program?

```
-bash-3.2$ bsub -l -q parallel_public -a mvapich2 -n 8 mpirun.lsf yourcode
```

This will submit your code executable to the parallel queue using mvapich2 and requesting 8 cpus.

Is there a queue for testing parallel programs that require a short run time?

There is queue called **paralleltest_public** just for this purpose. It has a run time limit of 10 minutes and a high priority.

Can parallel jobs be sent to any queue?

No... The Parallel_public queue and the test version is where most jobs should go. This queue has a limit of 64 cores/cpus. If you need access to more cores, we can add you to the Express_public queue which has a limit of 256 cores.

What mpi software is available?

OpenMPI, Mvapich and Mvapich2 are available on the cluster as a loadable module. Once the corresponding module is loaded, your environment will provide access to the various MPI compilers.

```
> module load openmpi
```

For example, OpenMPI provides the following:

```
mpic++ mpicxx mpicc mpiCC mpif77 mpif90
```

Likewise for mvapich and mvapich2.

How can I use Portland Compilers and MPI?

Try the broadcast example found in the PGI directory tree:
`/opt/pgi/linux86-64/7.2-3/EXAMPLES/MPI/mpihello/mynane.c`

As an example that requests 8 cores:

```
>module load pgi
>module load mvapich2
>pgcc myname.c -o myname -Mmpi -I/opt/pgi/linux86-64/7.2-3/include/
>bsub -l -q parallel_public -a mvapich2 -n 8 mpirun.lsf ./myname
```

Where are all the Portland executables and what are their names?

When you load the module for Portland, all executables will be on your path.

```
> ls /opt/pgi/linux86-64/7.2-3/bin
```

Can you recommend a good text on MPI?

The Tisch Library has:

William Gropp, Ewing Lusk, Anthony Skjellum, Using MPI: Portable Parallel Programming with the Message-Passing Interface, Second Edition, MIT Press, 1999, ISBN: 0262571323.

Another resource, Designing and Building Parallel Programs may be useful.

*Some additional supporting info on a parallel computing course can be found on this Tufts Computer Science [link](#).

Interesting Intel thread parallelism links and codes

Threading Building Blocks

See attachment for Intel white paper introduction pdf document.

GPU computing and CUDA resources

As part of the recent research cluster summer 2011 upgrade, one compute node was provisioned with two Nvidia Tesla M2050 GPU processors. GPU processing is an excellent means to achieve shorter run times for many algorithms. There are two approaches to use this resource. One is to program in Nvidia's programming language **Cuda**. The other approach is to use Matlab and other commercial applications that have GPU support.

Note, Nvidia Cuda and applications such as Matlab require specific coding to use gpu resources.

Note: Over time different versions of Cuda and sdk will change. Check the current versions available with the module command.

```
> module available
```

You'll find the CUDA toolkit in `/opt/shared/cudatoolkit` and the GPU computing **SDK** in `/opt/shared/gpucomputingsdk`. The SDK contains a number of CUDA sample C applications that can be found at `/opt/shared/gpucomputingsdk/4.2.9/C`. Compiled samples can be found in `/opt/shared/gpucomputingsdk/4.2.9/C/bin/linux/release`.

How does one find out gpu specific info?

```
> bsub -lp -q short_gpu /opt/shared/gpucomputingsdk/4.2.9/C/bin/linux/release/deviceQuery
```

Another option:

```
> bsub -q short_gpu -o gpu_info.txt nvidia-smi -a
```

To support GPU access new LSF GPU queues have been installed: **short_gpu**, **normal_gpu** and **long_gpu**.

For example to run one of the compiled cuda codes:

```
> cp /opt/shared/gpucomputingsdk/4.2.9/C/bin/linux/release/simpleStreams .
> module load cuda
> bsub -q short_gpu -lp -R "rusage [n_gpu_jobs=1 ]" ./simpleStreams
```

To view a description of sample codes from the command line:

```
> lynx file:///opt/shared/gpucomputingsdk/4.2.9/C/Samples.html
```

or

```
> firefox file:///opt/shared/gpucomputingsdk/4.0.17/C/Samples.html
```

The name of the cuda compiler is **nvcc** and other tools can be found in:

```
/opt/shared/cudatoolkit/4.2.9/cuda/bin
```

The **nvcc** help file is obtained:

```
> nvcc -h
```

Also, you can view local cuda pdf docs on the cluster:

```
> evince /opt/shared/gpucomputingsdk/4.2.9/C/doc/programming_guide/CUDA_C_Programming_Guide.pdf
```

Other pdf documents in:

```
/opt/shared/cudatoolkit/4.2.9/cuda/doc/ and /opt/shared/gpucomputingsdk/4.2.9/C/doc/
```

What gpu libraries are available on the cluster for linear algebra methods?

Cula routines are available for dense and sparse matrix settings and access is via the module environment. To see what is current:

```
> module available
```

```
Cula install directory is /opt/shared/cula/
```

Matlab GPU

A nice introductory article from [Desktop Engineering](#) of Matlab's GPU capability can be found [here](#).

Matlab's Parallel Toolbox GPU demonstration applications is an excellent introduction. Additional applications such as Mathematica, Ansys, Maple and others offer various levels of support within their product. Bsub usage would be similar.

For example, to run Matlab and access GPU resources:

```
> module load matlab
```

```
> bsub -q short_gpu -lp -R "rusage [n_gpu_jobs=1]" matlab
```

Additional GPU resources

There are many Cuda programming resources on the web and of course the Nvidia Cuda [website](#).

Stanford Seminars on High Performance Computing with CUDA

Stanford has posted videos from the Spring 2011 seminar series held at the Institute for Computational and Mathematical Engineering (ICME). The ICME is directed by Professor Margot Gerritsen.

- Lecture 1: Intro to HPC with CUDA 1 (Cyril Zeller)
- Lecture 2: Intro to HPC with CUDA 2 (Justin Luitjens)
- Lecture 3: Optimizations 1 - Global Memory (Inderaj Bains)
- Lecture 4: Optimizations 2 - Shared Memory (Steven Rennich)
- Lecture 5: Finite Difference Stencils on Regular Grids (Paulius Micikevicius)

The following Stanford Univ. video [lectures](#) are available for viewing.

HPC & GPU Supercomputing Group of Boston

A [group](#) for the application of cutting-edge HPC & GPU supercomputing technology to cutting-edge business problems.

Prof. Lorena Barba's research group at Boston University

She is a computational scientist and fluid dynamicist with research interests including [GPU](#) computing.

Look around the web as there are many similar GPU resources.

Tufts Parallel Users group

A recently formed group with some common interests: [link](#)