Running jobs on Scholar -- a quick guide

To prepare the environment do do a compile (where "$" is the command prompt):

```
$ module load devel
$ module load intel
```

This loads the necessary libraries into your session so that you can use the intel compiler and development libraries.

To compile your program:

```
$ mpicc -fopenmp pgm.c -o pgm
```

This will compile a program that can run across different nodes and communicate using MPI. On each node OpenMP pragmas can be used.

To submit your program:

```
qsub -l nodes=16:ppn=16,walltime=00:10:00 ./mpi.sub
```

This will get you 16 nodes and 16 threads per node. The scheduler will kill the job after it has run for 10 minutes. The script/binary/executable to be run is named "*mpi.sub*". Note that this *schedules* your job. It may take a while to run. Do not submit multiple jobs to the scheduler at the same time, especially when the wait times for a job to run are long. If people are too abusive I will have to do something to limit this.

The mpi.sub script:

mpi.sub can be a script that runs your program. For a full run over multiple numbers of nodes, it can look like:

```
#!/bin/sh -l
# FILENAME: mpi.sub

module load devel
cd $PBS_O_WORKDIR

./run_test_mpi.sh

run_test_mpi.sh contains:
echo $(ls -xS docs/*.txt) | xargs mpiexec -n 1 ./mpi_test
echo $(ls -xS docs/*.txt) | xargs mpiexec -n 2 ./mpi_test
echo $(ls -xS docs/*.txt) | xargs mpiexec -n 4 ./mpi_test
echo $(ls -xS docs/*.txt) | xargs mpiexec -n 8 ./mpi_test
echo $(ls -xS docs/*.txt) | xargs mpiexec -n 16 ./mpi_test
```

This script will run your job on 1, 2, 4, 8 and 16 nodes.