Compiling and Running Parallel Programs Locally

In parallel programming, students may find it useful to install Cygwin, multiple Cygwin Packages, and the Eclipse for Parallel Application Developers IDE (aka Eclipse PTP) to run parallel programs using C, C++, and/or Fortran90.

You can find some slightly dated instructions to install Cygwin and several packages such as gcc, ssh, mpi, and others at the following site:

http://catpages.nwmissouri.edu/m/monismi/cs345/InstallingCygwin.pdf

Note that when installing Cygwin, you must install the following packages to ensure compatibility with Eclipse PTP.

gcc-core
g++
openmpi (a library)
nano (an editor)
vi (another editor)
bash
openssh
openssl
make

You can find the download link for Eclispe for Parallel Application Developers (also called Eclipse PTP) below:

http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/luna/SR1/eclipse-parallel-luna-SR1-win32-x86_64.zip

You can find the slides that have information on how to use Eclipse PTP at the following link:

http://download.eclipse.org/tools/ptp/docs/ptp-xsede14.pdf

Be aware that much of the information in these slides is meant for individuals who have not used an IDE before. You may be able to skip past many slides

It is possible to synchronize a project with a remote system such as LittleFe and use the compiler on that system to compile and run your code remotely.

If you are using a windows machine you will need to modify your path on your own machine to run programs from Eclipse PTP through Cygwin.

You must add the location of the cygwin1.dll file to your System path in windows. Generally, this means adding the following location to your path:

C:\cygwin64\bin

You can also find instructions to add Cygwin to your path for use in Eclipse PTP in the "Additional Instructions for Windows Users" at the following site:

http://wiki.eclipse.org/PTP/photran/documentation/photran8installation#Additional_Instructions_for_Windows_Users

If you install Eclipse PTP, you may want to try to run an OpenMP C program such as the Hello world program provided with Eclipse or the one provided below:

```c
#include <omp.h>

main ()  {

int nthreads, tid;

/* Fork a team of threads with each thread having a private
tid variable */
#pragma omp parallel private(tid)
  {
  /* Obtain and print thread id */
  tid = omp_get_thread_num();
  printf("Hello World from thread = %d\n", tid);

  /* Only master thread does this */
  if (tid == 0)
    {
    nthreads = omp_get_num_threads();
    printf("Number of threads = %d\n", nthreads);
    }

  }  /* All threads join master thread and terminate */
}
```

Source code is from https://computing.llnl.gov/tutorials/openMP/