**PRACTICAL # 01**

**OBJECT:**

Tools and environment Setup for OpenMPI

**THEORY:**

This course is aimed to cover the basics and general ideas behind and basic principles of high performance computing (HPC) as performed on a supercomputer. The importance of communications overhead in high performance computing and practical aspects of message passing. These concepts should remain valid even as the technical specifications of the latest machines continually change.

By definition, supercomputers are the fastest computers. The term supercomputer refers to machine with hundreds of thousands of processors. Personal computers are small in size and cost to be used by an individual user, yet powerful enough to run advanced scientific applications. These PCs nowadays contain dual processor cores or quad cores. So they can also be utilized for parallel processing.

**Message Passing Interface (MPI)**

MPI is a communications protocol used to program parallel computers. We will use OpenMPI as an implementation of MPI. It supports both point-to-point and collective communication. MPI "is a message-passing application programmer interface, together with protocol and semantic specifications for how its features must behave in any implementation." MPI's goals are high performance, scalability, and portability. MPI remains the dominant model used in high-performance computing today.

Some important Features of OpenMPI:

* Supports Network Heterogeneity
* Different OS environments supported
* OpenMPI is portable

**OpenMPI Installation in Linux**

The following instructions will help installing OpenMPI on your linux machine.

**1. Create a temporary directory** for compiling OpenMPI. Type in a terminal:

mkdir $HOME/local/src

**2. Download openmpi**-*version*.tar.bz2 from [http://www.open-mpi.org](http://www.open-mpi.org/).

The *version* should be latest.

**3. Move the openmpi**- *version*.tar.bz2 to the directory just created:

mv $HOME/Downloads/openmpi-*version* .tar.bz2 $HOME/local/src/

**4. Change to the directory and extract the package** using

cd $HOME/local/src

tar -jxf openmpi-*version* .tar.bz2

**5. Go into the source directory**

cd openmpi-*version*

**6. Configure, compile and install** by executing the following commands

./configure --prefix=$HOME/opt/openmpi

make all

make install

This will install OpenMPI in your home directory in the sub-folder opt/openmpi. To speed up the compilation, replace the make all command with make -j4 all (this will compile using 4 cores).

**7. Remove the temporary directories**:

rm $HOME/local/src/openmpi-*version*.tar.bz2

rm -r $HOME/local/src/openmpi-*version*

To use MPI you will have to adapt your PATH and LD\_LIBRARY\_PATH environment variable:

echo "export PATH=\$PATH:\$HOME/opt/openmpi/bin" >> $HOME/.bashrc

echo "export LD\_LIBRARY\_PATH=\$LD\_LIBRARY\_PATH:\$HOME/opt/openmpi/lib" \

>> $HOME/.bashrc

This appends the two lines to your .bashrc file which is executed when starting a terminal session.

To compile your MPI C++ programs you have to use mpicxx with the same arguments as you would use for g++.

Example: mpicxx prg.cpp -o PRG

Note that to compile a C program use mpicc instead of mpicxx.

To run a program PRG with N MPI processes, you would then use mpirun -np N PRG.

where -np switch is used to specify the number of parallel processors N that will be used to run the target program PRG.

**OpenMPI Installation in Windows**

The OpenMPI setup for windows is available at

[www.open-mpi.org/software/ompi/v1.6/](http://www.open-mpi.org/software/ompi/v1.6/)

Download and install the executable setup of OpenMPI. Set the environment path for OpenMPI pointing to the location of OpenMPI compiler (mpicc/mpic++).

The OpenMPI compiler is a wrapper compiler over the C++ compiler. So you much have some C++ compiler like cl.exe. To get cl.exe install MS Visual C++. If you already have cl.exe, add set its environment path so that the OpenMPI wrapper compiler can use it to compile its programs.

To check the installation, open CMD and type

**mpicc -v**

If it shows the information, OpenMPI is successfully installed and setup. Write a simple C source program and compile it using OpenMPI to check if cl.exe usable by OpenMPI.

**mpicc Hello.c**

If the program is successfully compiled, it will generate an executable file.

**Note:** The mpicc is the C program compiler and mpic++/mpicxx is the C++ compiler.

**ACTIVITIES**

**Activity 1**

Setup the OpenMPI environment on your computer. Follow the above instructions for setup.

**Activity 2**

Write a simple C “Hello World” program and compile it using mpicc command line compiler.

**Review Questions:**

Q1: What is meant by MPI?

Q2: What is OpenMPI?

Q3: What are the important features of OpenMPI?

Q4: How does OpenMPI compile the source program?