

HOW TO RUN

--ABOUT--

DATE: 03/01/2014

COURSE: CSCI-B673

YEAR: Spring 2014

DEVELOPER: Prachi Shah (pracshah)

ASSIGNMENT: ex1

--CONFIGURATION--

OPERATING SYSTEM: Linux

RAM SIZE: 2 GB or higher

PROCESSOR SPEED: 2GHZ or higher

--SOFTWARES USED--

COMPILER: mpicc compiler

TEXT EDITOR: gedit

SYSTEM: odin.cs.indiana.edu

SOFTWARE: MATLAB

--FILE LISTINGS AND PURPOSE--

rr.c	: The main file that contains the code for implementing Round robin. Contains variables and function to implement the Round Robin scheduling.
results	: The output file that stores the results.
report	: Contains the analysis that were found after using the 'result' file with Matlab script. The analysis of implementing the round robin message passing amongst different processes by decreasing the number of cycles and increasing the message size.
howToRun	: The 'README' file.
slurmlog	: Log file.
showres.m	: Matlab script file.

--DESCRIPTION OF CODE--

> The assignment includes writing MPI code to pass a message in a round robin fashion amongst different processes on different nodes.

> The time required for message passing per cycle for a combination of different number of processes (4, 5,..., 16) and different message sizes (1, 10, 100,..., 1000000.) are calculated. The MPI_Wtime function is used to record the time.

> MPI_Bcast function is used to broadcast message to different processes.

> ROUND ROBIN SCHEDULING:

- Performs message passing between different processes in a time-sharing mode.

> To run the program for 16 processes, the maxNumProc variable in the 'runit' file is changed from 8 to 16.

> The input file 'sizes' contains the following:

0 1 6

minSz stride maxSz

The message size range from 10^0 to 10^6 and a stride of 10^1 .

> The output file 'result' is of the following format:

#processes msgSize #cycles totaltime

where, #processes gives the number of processes

msgSize gives the size of the message

#cycles gives the number of cycles

totaltime gives the time taken for execution

--COMPILING THE PROGRAMS--

- The computer machine should have OpenMPI package installed in order to execute this program.

> make

Create object file

--RUNNING THE PROGRAMS--

> ./launcher

Enables execution of the program on a node on the Odin machine.

> cat slurmlog

Stores the log of the processes being created.

> cat result

Store the output results which include number of processes, size of message, number of cycles and time taken for execution.

- Open Matlab software and run the script 'showres.m'. This script will use the 'results' file and analyze the data to create different graphs.

- These graphs are used to understand the relation between how an increase in message size will effect the execution time when the number of cycles are decreased.

--CREDITS--

1) Round-robin scheduling. http://en.wikipedia.org/wiki/Round-robin_scheduling

2) <http://nf.nci.org.au/training/MPIProg/slides/>

3) <http://particlephysicsandcode.com/2012/11/04/installing-open-mpi-1-6-3-ubuntu-12-04-fedora/>

4) <http://hpcc.usc.edu/support/documentation/examples-of-mpi-programs/>

5) Akshay Dorwat

6) <http://www.open-mpi.org/faq/?category=building>

7) Matlab software