COMP 605: Introduction to Parallel Computing
Homework 3:
Characterizing 1D MPI Communication

Mary Thomas

Department of Computer Science
Computational Science Research Center (CSRC)
San Diego State University (SDSU)

Due: 03/16/17
Posted: 02/13/17
Updated: 03/16/17

Table of Contents

HW3 P1: MPI Comms: *Ping-Pong*

In this project we will characterize and time how long it takes for two PE's pass a message packets of difference sizes back and forth to each other.

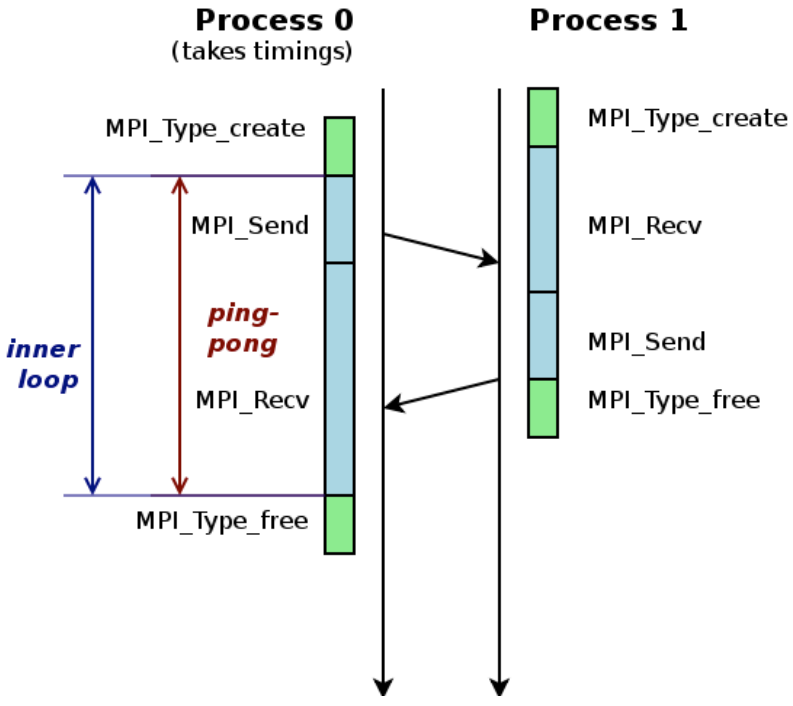
For both problems you will use:

- *point-to-point* comm routines: *MPI_Send* and *MPI_Recv*.
- *collective* comm routine: *MPI_Send_Recv*.
- You will be given test code (or download some other reference code), and modify it to suit the requirements specified below.
- See lectures on Performance and MPI Communications.

Timing MPI Messages - Ping-Pong Algorithm

TimeStep	P_0	P_1
t_0	MPI_Send message to P_1	WAITS for message from P_0
t_1	WAITS for message from P_1	MPI_Recv message from P_0
t_2	WAITS for message from P_1	MPI_Send message to P_0
t_3	MPI_Recv message from P_0	

System has $sz = comm\_sz = 2$
Processors numbered $[P_1, P_2]$



Img source: `http://htor.inf.ethz.ch/research/datatypes/ddtbench/benchmark_expl.png`

HW3 P1: MPI Comms: *Ping-Pong* & *Ping-Exch* Code

- For all problems:
  - Message is an array of doubles, of size based on packet size if 10^N.
  - Process a packet size N from command line argument
  - Packet Size is gradually increased until a bandwidth limit (max) is reached.
- For all problems, choose test codes located in */COMP605/hw3/* on tuckoo, or another equivalent code (include URLs/ references).
  - **P1a:** Use *point-to-point* comm routines: *MPI_Send* and *MPI_Recv*.
  - **P1b:** Use *point-to-poin* comm routine: *MPI_SendRecv*.
- You will have to modify the code:
  - to eliminate STDIO input(no scanf)
  - remove hard coded dimensions, instead use dynamic allocation based on N
  - handle errors, etc.

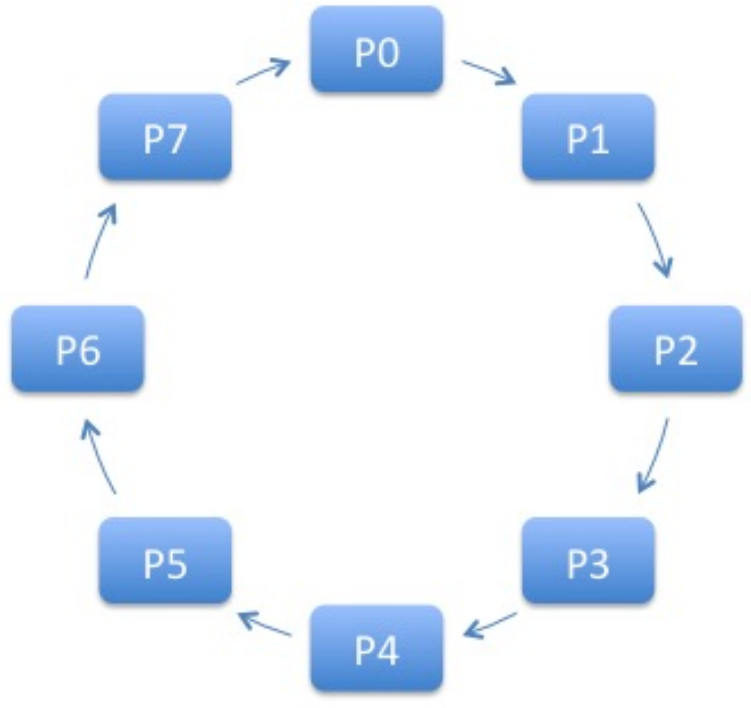
HW3, P2: Comms: *Ring*

In this project we will characterize and time how long it takes for a group of PE's to pass a message packet around the entire group.

- For all problems:
  - Message is an array of doubles, of size based on packet size if 10^N.
  - Process a packet size N from command line argument
  - Packet Size is gradually increased until a bandwidth limit (max) is reached.
- For all problems, choose test codes located in */COMP605/hw3/* on tuckoo, or another equivalent code.
  - **P1a:** Use point-to-point comm routines, *MPI_Send* and *MPI_Recv*.
  - **P1b:** Use point-to-point comm routine, *MPI_SendRecv*.

Timing MPI Messages - Ring Algorithm

- System has $sz = comm\_sz$ processors numbered:
  $P_0, P_1, , P_{r-1}, P_r, P_{r+1}, ..P_{sz-1}$
- P_0 *sends* msg to P_1
  P_0 *waits* for msg from P_{sz-1}
  ...
  P_r *waits* for msg from P_{r-1}
  P_r *rcvs* msg, *sends* msg to P_{r+1}
  ...
  P_{sz-1} *sends* to P_0
  P_{sz-1} *waits* for msg from P_{sz-2}



8 Processors arranged in a ring

General Test Case Instructions

- All tests should be run for the following conditions:
  - Number of PEs: $[2, 4, 8, 16]$
  - Packet Sizes $= 10^n$, where $n = [1, 2, \ldots, 8]$.
  - Test for double data types
  - Timing Statistics: no need to do more than 5 runs.
- Report should include:
  - Measurements of elapsed times and Bandwidth statistics (see below under General Instructions).
  - Theoretical prediction of run-times.
  - Estimate the time per point required for your calculation (using T_{ser}) and predict how long your runs should take.
  - Calculate Bandwidth (see below under General Instructions).
  - Calculate the startup time or latency.
  - Table of data for data runs: Packet Size, BW, Timing, & Stats
  - Plot(s) of the BW as a function of Packet Size, #PEs.
  - Additional items in General Instructions (below).

Writeup & Turn-in Instructions

Description: This homework involves measuring the performance of MPI Communications.

- All code must be run on the student cluster as batch jobs.
- For each problem, create a homework directory for each problem in your home directory:

  /home/605/accountname/hw/hw3/p1

  /home/605/accountname/hw/hw3/p2
- You may work with copies of source codes, located on the student cluster in the directory */COMP605/hw3* or find another source. Be sure to reference your code.
- Add timing diagnostics where needed.
- Report should Include: code; a few examples of batch scripts, and relevant results; tables; plots; stats; comments.

Calculating BandWidth (BW) and latency

- Latency is the time needed to start the message (beginning)
- BW is the rate observed for large messages.
- Units typically Mega or Giga Bytes per second (e.g. GByte/sec)
- Estimate packet size per send or recv
- Estimate the number of sends or recvs you are counting
- Units: are you calculating BITS/sec, or BYTES/second? Convert packet size accordingly
- Units: what are your timer units?
- Example estimation: Ping-pong:

$$BW\left[\frac{a}{b}\right] \cong \frac{(\#exchanges)*packetSize[floats]*size[1float]}{rawTime[\mu sec]}$$

$$\cong 2 * \frac{[exchanges]*10^6[floats]*32[bits/float]}{3x10^{-3}[seconds]}$$

$$\cong 21x10^9 \frac{bits}{second} * \frac{1Byte}{8bits}$$

$$\cong 2.67x10^9 \frac{Bytes}{second}$$

$$\cong 2.67 \frac{GBytes}{second}$$

Statistical Methods

Run times on any computer are not reproducible, hence, it is important to analyze the distribution of a codes' run times, and not just take one measurement.

- Standard statistical variables used to describe the distribution of the data include:
  - Max/Min (maximum/minium values)
  - Mean (average value)
  - Median (central value)
  - Variance (variance)
  - StandardDeviation (σ) of the timings.
- To test your codes:
  - Run and time critical blocks
  - Vary key parameters (packet or problem sizes, number of processors, etc.).
  - Calculate the statistics at run-time.
- Refs:
  `http://reference.wolfram.com/language/tutorial/BasicStatistics.html`
  `http://edl.nova.edu/secure/stats/`