ci1316/ci316 /ci1009 - PROGRAMAÇÃO PARALELA -
2o semestre de 2022
por W.Zola/UFPR

Lab 3a: Vazão e latencia de mensagens em MPI (parte a)

Histórico:
- v1.0 a versão inicial

Data do enunciado: 14/dez/2022
Data da entrega Lab3a: **20/dez/2022 (DATA determinada! próxima TERÇA!)**

Objetivo do trabalho 3:
O objetivo deste exercício é fazer um trabalho inicial simples para prática de conceitos de MPI. Ao mesmo tempo gostaríamos de ter uma idéia inicial das medidas de desempenho da troca de mensagens em MPI.

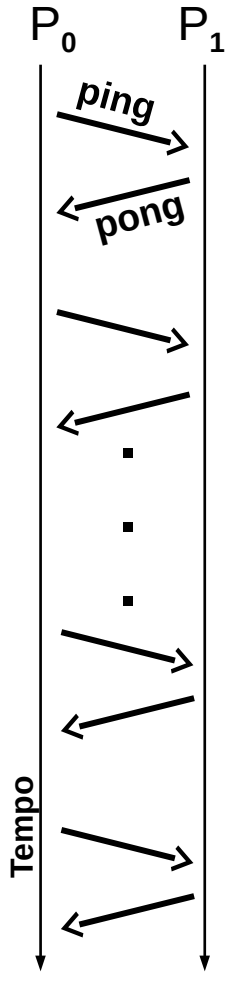
O trabalho será dividido em:

- parte 3a (mais simples, somente 2 processos MPI)
- parte 3b (um pouco mais elaborada, deve funcionar para qualquer número np de processos, sendo que o número de processos é fornecido na linha de comando de execução do programa MPI.
  - OBS: a especificação da parte 3b será colocada na UFPR virtual na semana que vem e teremos mais tempo para fazer a parte 3b.

Idéia da parte 3a:

Queremos medir o comportamento da latência e vazão de mensagens entre dois processos MPI.
Tanto as medidas de latência e vazão serão obtidas por experimentação. Para facilitar nossas medidas serão feitas com TODOS os processos MPI rodando no mesmo processador, ou seja, não teremos uma rede de interligação entre os processos/nodos MPI. Nesse cenário, nossas medidas serão dependentes da vazão e latência de memória RAM do nosso sistema onde executaremos os testes. No entanto, se colocarmos nosso programa MPI em um cluster essas medidas poderiam ser coletadas no contexto distribuído. Para esse trabalho (3a e 3b) faremos as medidas somente em uma CPU multicore com todos os processos rodando na mesma.

A idéia da parte 3a pode ser vista na fig abaixo:



- Todas as mensagens (ping ou pong) terão o mesmo tamanho
- o programa deve obter da linha de comando:
  - o número total de mensagens (**nmsg**)
    - o número de mensagens deve ser sempre par (i.e. múltiplo de 2), caso contrário o programa deve emitir msg de erro.
  - o tamanho de cada mensagem (**tmsg**, dado em bytes)
    - o número de bytes deve ser sempre múltiplo de sizeof(long) ou seja, múltiplo de 8, caso contrário o programa deve emitir msg de erro.
  - o número de processos MPI
    - para essa parte 3a, o número de processos MPI fornecido na linha de comando deve ser sempre 2 ou uma msg de erro deve ser emitida
  - o ÚLTIMO PARÂMETRO da linha de comando deve ser opcional. SE especificado, esse parâmetro pode ser: **-bl** OU **-nbl**

- SIGNIFICANDO:

  - **-bl** indica que o programa usará mensagens MPI send default (o mesmo acontecerá se nada for especificado aqui)

  - **-nbl** indica que o programa usará mensagens MPI **não bloqueantes** (o mesmo acontecerá se nada for especificado aqui)

- Como as mensagens são sempre compostas com uma sequência de números inteiros (long int). Portanto, se o tamanho de cada msg é tmsg, temos que, cada mensagem é um vetor de **ni = tmsg/8** inteiros long int:

  - no inicio de cada processo MPI os vetores devem ser inicializados com os números long int.

    - O processo P0 deve inicializar sua mensagem a ser enviada (inicializa UMA vez apenas) com valores long int no intervalo fechado **[1..ni]**

    - O processo P1 deve inicializar sua mensagem a ser enviada (inicializa UMA vez apenas) com valores long int no intervalo fechado **[(ni+1)..(2*ni)]**

- O sistema de processos deve enviar e receber as mensagens ping e pong, até o **total** de **nmsg** mensagens.

  - O processo P0 deve calcular o tempo medido para enviar e receber todas as msgs.

  - Ao final o processo P0 deve imprimir:

    - O tempo total para enviar/receber todas as mensagens.

    - A latencia de cada mensagem (tempo que a mensagem gasta para sair do nodo e ser recebida no outro nodo.

    - A vazão de transferência alcançada na experiência.

- Verificação de correção:

  - O professor vai fornecer uma função de verificação dos vetores recebidos. Seu programa deve usar a funçao de verificação. O processo P0 vai verificar se recebeu corretamente a msg com os valores vindos de P1, e vice-versa, O processo P1 vai verificar se recebeu corretamente os valores de P0. Essa verificação deve ser feita APENAS depois de coletadas as medidas.

  - As medidas de tempo DEVEM ser feitas com a nossa biblioteca chronos (a mesma usada no lab 2). OBS: NÃO usaremos as funções de tempo de MPI, usaremos a nossa chronos.

- SOBRE AS EXPERIÊNCIAS:

  - Voce deve medir 10 vezes cada experiência, para mensagens de tamanho:

    - 8 bytes, 1KB, 4KB, 16KB usando mensagens bloqueantes

    - 8 bytes, 1KB, 4KB, 16KB usando mensagens NÂO bloqueantes

  - Quantas mensagens deve enviar em cada experiência?

    - R: tente dimensionar o número de mensagens **nmsg** na linha de comando (para cada experiência) ou script, de modo que cada rodada (com um dado tamanho de mensagem) tome aproximadamente 1 segundo. Assim o tempo total para medir tudo não será muito

alto (aproximadamente 40 segundos para todas as experiências bloqueantes e 40 segundos para NÃO bloqueantes).

- SOBRE OS GRAFICOS DAS EXPERIÊNCIAS:
  - Fazer um gráfico de barras da vazão média obtida para mensagens (eixo y) sendo o eixo x os tamanhos das mensagens: 8 bytes, 1KB, 4KB, 16KB
  - Fazer outro gráfico de barras da latência média obtida para mensagens (eixo y) sendo o eixo x os tamanhos das mensagens: 8 bytes, 1KB, 4KB, 16KB
    - OBS: Faça os gráficos no mesmo tipo de planilha do trabalho 2 (mudando as tabelas e os calculos e os tipos dos gráficos)
  - REPETIR os dois gráficos acima para mensagens NÃO bloqueantes (total de 4 gráficos)

- SOBRE O RELATÓRIO do trabalho:
  - Fazer um relatório (PDF) com suas conclusões. Descreva as características importantes da sua CPU usada. Inclua em apendice ao relatório o texto da saida do comando lscpu bem como a figura do comando lstopo para sua CPU usada.