Aula 01- Int. à programação por passagem de mensagens

Assumem-se dois atributos chave:
- espaço de endereçamento de memória particionado
- suporta apenas a paralelização explícita

Comunicação e a sincronização são associadas

Primitivas básicas para realizar comunicação ponto-a-ponto:
- send(void *sendbuf, int nelems, int dest)

- receive(void *recevbuf, int nelems, int source)

- Detalhar o funcionamento de que cada

Há outras primitivas de comunicação:
- rendezvous
- comunicações coletivas
- Estruturas para comunicação de mais alto nível com passagem de mensagens:
  - RPC, RMI e web services.

Há uma visão lógica de processos cada um com seu próprio espaço de endereçamento
- Modelo voltado para clusters e outros multicomputadores com Mem Distribuída

Dados pertencem à memória local:
- Dados devem ser divididos e atribuídos adequadamente;

Interações bilaterais:
- Ambos os processos cooperam para a comunicação;

Programador tem condições de (deve) reduzir a comunicação da aplicação
- Otimização do desempenho;

Localidades espacial e temporal são determinantes para o desempenho;

Flexível por ser aplicado eficientemente em diferentes arquiteturas
- Alterando-se a granularidade

Programas tendem a ser mais complexos
- Quando comparados à programação via mem compartilhada

Programas com passagem de mensagens são considerados
- Assíncronos porque as tarefas concorrentes executam assincronamente;
- Fracamente sincronizados porque interagem menos que as threads;

Execução na CPU é não determinística

Não há compartilhamento de memória
- sem regiões críticas, exclusão mútua e condições de disputa

Programação pode ser MPMD ou SPMD
- MPMD é mais flexível e mais complexo. Pode dificultar a escalabilidade.
- SPMD é mais simples e voltado mais ao paralelismo de dados.

Introdução ao MPI

Apresentação e histórico
- Anteriormente: P4, Parmacs, Express, Linda, PVM e começo do MPI

Características das Principais Versões

MPI-1.3

Quando: 04 de setembro de 2008
Quantas funções: 129
Inclui Comunicação ponto-a-ponto e coletivas
Grupos de processos
Contexto de comunicação
Topologias de processos
Interfaces para Fortran 77 e C
Gerenciamento da plataforma e coleta de informações

MPI-2.2

Quando: 04 de setembro de 2009
Quantas funções: 300 funções
Inclui a mais que a versão anterior:
Tipos de dados
Objeto de informação (criado e associado a rotinas)
Criação e gerência de processos
Comunicação unilateral (One-sided)
Interfaces externas (criar novas op não bloq do usuário)
E/S paralela
Interfaces também para C++

MPI-3

Quando: 21 de setembro de 2012
Quantas funções: 437 funções
Inclui a mais que a versão anterior:
Extensão para mensagens coletivas não bloqueantes
Extensão para operações unilaterais
Interface para Fortran 2008
Removida a interface para C++

Elementos básicos do código, compilação, execução, e funcionamento:

Inserir no programa: #include <mpi.h>

Compilação: ***mpicc -o programa programa.c***

Execução: ***mpirun -np 4 executável [args]***

O parâmetro ***-np*** especifica a quantidade de processos a serem criados, em que cada um deles executa uma cópia do programa *executável*. É importante notar que o valor de ***-np*** depende da quantidade de *slots* disponíveis na arquitetura onde a aplicação será executada. Para o MPI, por padrão, os ***slots*** representam a quantidade de processadores físicos disponíveis.

Caso deseje-se utilizar também os processadores virtuais pode-se utilizar o parâmetro ***--use-hwthread-cpus***.

Para permitir que haja mais de um processo por ***slot***, *i.e.,* sobrescrever a limitação dos ***slots***, deve-se usar o parâmetro ***--oversubscribe***.

Um arquivo de hosts também pode ser utilizado, o qual constitui um arquivo de texto, onde cada linha contém o IP ou nome da máquina a ser utilizada. Para isto, deve-se usar o parâmetro ***--hostfile hostfile***, onde ***hostfile*** é o arquivo que contém os endereços ou nomes dos dispositivos. Os arquivos de ***hosts*** podem ter outras informações também: vide documentação do MPI para mais detalhes. Outros parâmetros do ***mpirun*** não serão discutidos. Consulte o documento https://www.open-mpi.org/doc/v4.0/man1/mpirun.1.php.

Funções básicas do MPI:

MPI_Init,
MPI_Comm_size,
MPI_Comm_rank,
MPI_Finalize,
MPI_Send
MPI_Receive

Conceito de contexto de comunicação e comunicadores para o MPI

MPI_COMM_WORLD

Tipos de dados do MPI:

Tipo do MPI	Tipo do C
MPI_CHAR	char
MPI_SHORT	short int
MPI_INT	int
MPI_LONG	long int
MPI_LONG_LONG_INT	long long int
MPI_UNSIGNED_CHAR	unsigned char
MPI_UNSIGNED_SHORT	unsigned short int
MPI_UNSIGNED	unsigned int
MPI_UNSIGNED_LONG	unsigned long int
MPI_UNSIGNED_LONG_LONG	unsigned long long int
MPI_FLOAT	float
MPI_DOUBLE	double
MPI_LONG_DOUBLE	long double
MPI_WCHAR	wide char
MPI_PACKED	special data type for packing
MPI_BYTE	single byte value

Tabela 1: Tipos de dados do MPI

Códigos vistos em sala de aula e solicitados como exercícios:

Execução em sala:
Hello World MPI

Exercícios solicitados:

1) Dado um vetor vet[TAM], determinar quantos nrs maiores que vet[K] existem em vet.

2) Faça um programa concorrente em C/MPI que implemente um Token Ring de processos, estes dispostos logicamente na forma de uma fila circular. O primeiro processo (de rank zero), deverá gerar um token = 0 (zero, um valor inteiro), passar este para o próximo processo (rank 1) e aguardar o recebimento do token do último processo (rank n-1). O processo zero, ao receber de volta o token, vindo deste último processo, imprime na tela o valor recebido e finaliza. Todos os processos devem receber um valor de outro processo, incrementá-lo e então repassar esse novo token ao próximo processo na fila circular. Após uma volta do token pelos processos, a aplicação finaliza.

Bibliografia:

Rauber, T., & Rünger, G. (2013). *Parallel Programming*. Springer. Second edition. Capítulo 5.
Pacheco, P. (2011). *An introduction to parallel programming*. Elsevier. Capítulo 3.
Barlas, G. (2014). *Multicore and GPU Programming: An integrated approach*. Elsevier. Capítulo 5.
Grama, A., Kumar, V., Gupta, A., & Karypis, G. (2003). Introduction to parallel computing. Pearson Education. Capítulo 6.
Apostila de Treinamento: Introdução ao MPI (Unicamp).
https://www.cenapad.unicamp.br/servicos/treinamentos/apostilas/apostila_MPI.pdf