Aula 04 – Intercomunicadores e Geração Dinâmica de Processos

Intercomunicadores e Intracomunicadores

Os intracomunicadores permitem a comunicação entre processos de um mesmo contexto de comunicação. Para que processos de contextos distintos se comuniquem são necessários intercomunicadores no MPI. Os intercomunicadores podem ser criados quando novos processos são gerados dinamicamente através de *MPI_Comm_spawn()*. Outra maneira de criar intercomunicadores no MPI é através da função *MPI_Intercomm_create()*, a qual cria um intercomunicador a partir de dois intracomunicadores já existentes. A função *MPI_Intercomm_create()* não será vista em nossas aulas.

A Fig. 4.1 ilustra dois intracomunicadores denominados MPI_COMM_WORLD, sendo um com cinco processos e outro com três. Os processos em cada grupo podem se comunicar localmente, mas não entre grupos distintos, i.e., cada intracomunicador determina um contexto local de comunicação. O intercomunicador *inter-comm* determina um novo contexto de comunicação com dois grupos de processos, e permite a comunicação entre processos existentes em grupos distintos de comunicação (dois MPI_COMM_WORLD no caso da Figura 4.1).

As primitivas coletivas de comunicação têm comportamento diferente quando são usados intercomunicadores. A principal diferença é que os processos transmissores ou receptores das comunicações coletivas podem não contribuir com dados durante a comunicação. No caso do *MPI_Gather* com intercomunicadores, por exemplo, o processo root não fornece uma porção dos dados como ocorre quando o *MPI_Gather* é usado com intracomunicadores. Outro exemplo é o MPI_Scatter que não recebe uma porção dos dados quando são usados intercomunicadores. As primitivas coletivas também precisam usar MPI_ROOT para determinar o processo da operação que receberá ou transmitirá os dados. Verifique a semântica esperada para cada função MPI antes de usar intercomunicadores.

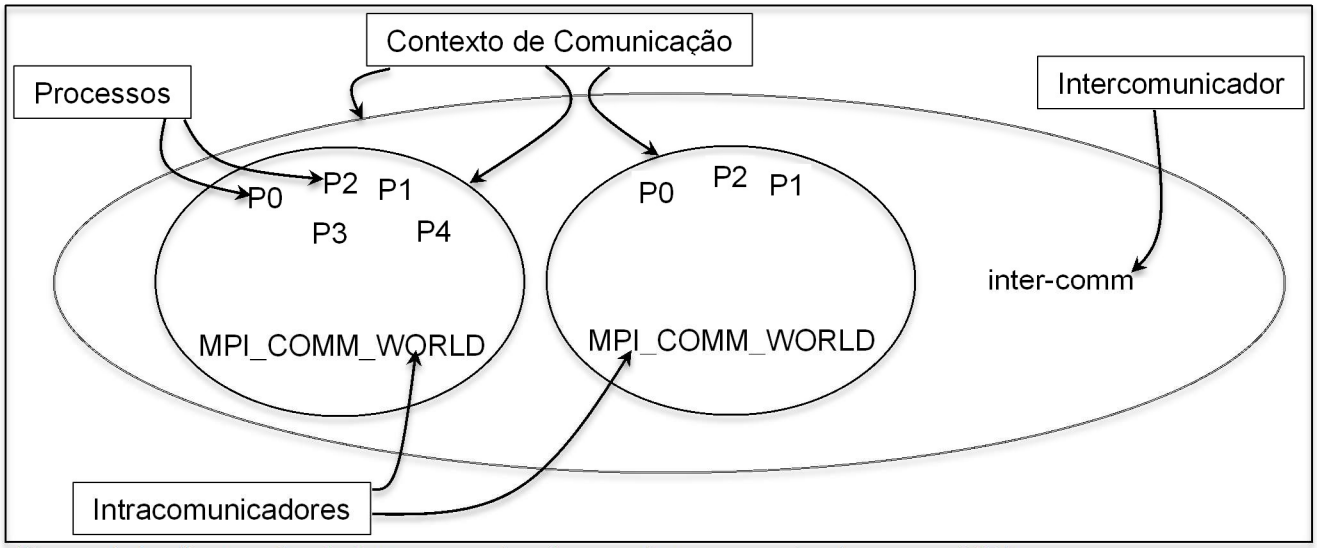

Figura 4.1 – Ilustração de Intracomunicadores e Intercomunicadores em MPI.

Geração dinâmica de processos & geração de processos remotamente no cluster

A geração dinâmica de processos no MPI é feita através da primitiva coletiva:

*int MPI_Comm_spawn (char *command, char *argv[], int maxprocs,MPI_Info info,*

*int root, MPI_Comm comm, MPI_Comm *intercomm, int errcodes[ ])*

O argumento de entrada ***command*** indica o nome do executável em cada processo. Em ****argv[ ]*** há os argumentos de entrada para o programa a ser carregado. O conteúdo de argv[0] não é o nome do programa em si, mas o primeiro argumento do programa. A macro MPI_ARGV_NULL determina que a lista de argumentos é vazia. O número de processos a serem criados é determinado por ***maxprocs***. O parâmetro ***Info*** determina parâmetros para a execução dos novos processos MPI, como por exemplo a rota no sistema de arquivos para encontrar o executável. A macro MPI_INFO_NULL pode ser usada quando não há argumentos a passar. O parâmetro ***root*** determina qual é o processo que realmente está gerando os novos processos, embora *todos os processos do grupo devam participar desta criação, pois trata-se de uma primitiva coletiva*. O parâmetro ***intercomm*** contém o ***intercommunicator*** *que será retornado* após a execução da primitiva. Um intercomunicador permite a comunicação entre os dois contextos distintos de processos (processos pais (original) e processos filhos (criados com o spawn). Ao criar os novos processos (filhos) com o MPI_Comm_spawn(), estes também formam um novo MPI_COMM_WORLD que permite a COMUNICAÇÃO LOCAL entre eles (filhos). O vetor apontado por ***errcodes*** contém o status do erro para cada processo an array with maxprocs entries in which the

Códigos exemplo:

Códigos do diretório 04a-spawn-coletivas. Os códigos estão numerados na sequência.

Exercícios solicitados:

Multiplicação de matrizes quadradas usando N^3 processos (N é a ordem das matrizes)
O código é lento e não visa ao desempenho. Visa mostrar o uso de várias primitivas MPI.
Código disponível em prog-MPI/mpi_examples/4b-matrizmult-spwn-collectives

Bibliografia:

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