CAP-387(2016) – Tópicos Especiais em Computação Aplicada: Construção de Aplicações Massivamente Paralelas

Aula 28: Comunicação Coletiva em MPI

Celso L. Mendes, Stephan Stephany

LAC / INPE

Emails: celso.mendes@inpe.br, stephan.stephany@inpe.br

Comunicação Coletiva em MPI

- Comunicação ponto-a-ponto em MPI:
  - Realizada entre um par de ranks num grupo
- Comunicação coletiva em MPI:
  - Realizada entre todos os ranks do grupo
- MPI_COMM_WORLD
  - Define todos os ranks existentes no início da execução
  - Porém, com gerenciamento dinâmico, podem ser criados outros processos fora de MPI_COMM_WORLD!

Operações Coletivas

- Comunicação ponto-a-ponto em MPI:
  - Realizada entre um par de ranks num grupo
- Comunicação coletiva em MPI:
  - Realizada entre todos os ranks do grupo
- MPI_COMM_WORLD
  - Define todos os ranks existentes no início da execução
  - Porém, com gerenciamento dinâmico, podem ser criados outros processos fora de MPI_COMM_WORLD!

Operações Coletivas

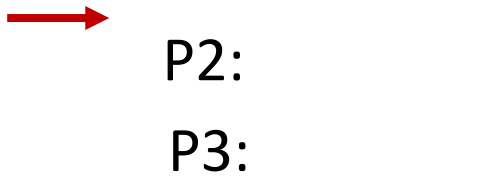
- Funções de coletivas podem ser com (versões mais antigas de MPI) ou sem bloqueio (versões recentes)
- Algoritmos coletivos eficientes podem ser complexos
  - Funções de MPI liberam o programador do problema
  - Bibliotecas são implementadas por especialistas
- Exemplo: *MPI_Bcast(void *buf, int count, MPI_Datatype datatype, int root, MPI_Comm comm)*
  - Algoritmo trivial: proc. root envia *buf* serialmente aos demais *P-1* procs
  - Otimização: distribuição de *buf* através de uma árvore binária, $log_2 P$ passos

Classes de Coletivas

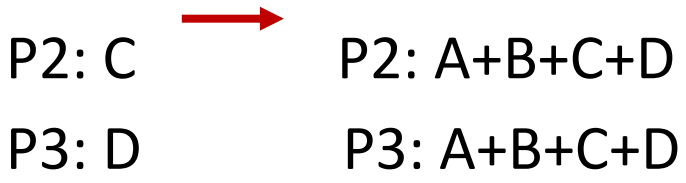
- Três classes de funções
  - sincronização
    - Ex: *MPI_Barrier(comm)*
  - movimentação de dados
    - Um para todos: *MPI_Bcast(…), MPI_Scatter(…)*
    - Todos para um: *MPI_Gather(…)*
    - Todos para todos: *MPI_Alltoall(…)*
  - computação coletiva: comunicação + computação
    - Todos para um, com combinação: *MPI_Reduce(…)*
    - Todos para todos, com combinação: MPI_Scan(…)
    - Todos paratodos, com combinação: MPI_Reduce_scatter(…)
- OBS: devem ocorrer na mesma ordem em todos os ranks

Computação Coletiva

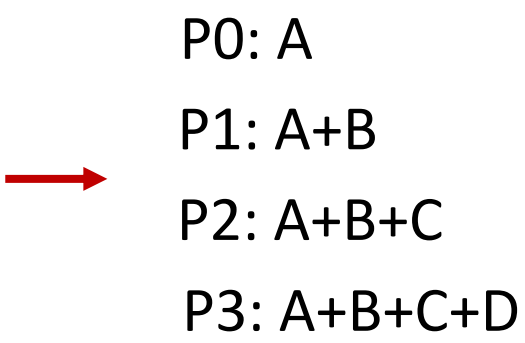
- Reduce:



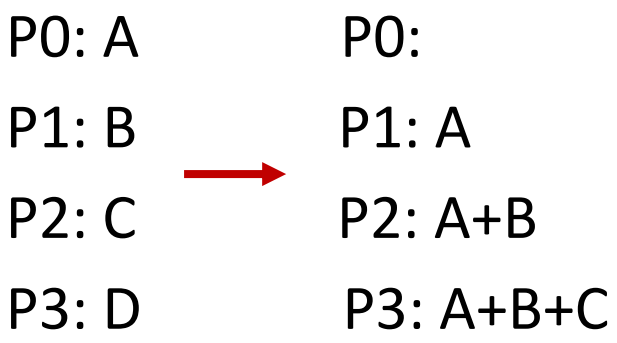
- Allreduce:



- Scan:



- Exscan:



Computações Disponíveis

MPI_MAX	Máximo
MPI_MIN	Mínimo
MPI_PROD	Produto
MPI_SUM	Soma
MPI_LAND	E Lógico
MPI_LOR	OU Lógico
MPI_LXOR	OU-Exclusivo Lógico
MPI_BAND	E bit-a-bit
MPI_BOR	OU bit-a-bit
MPI_BXOR	OU-Exclusivo bit-a-bit
MPI_MAXLOC	Máximo e localização
MPI_MINLOC	Mínimo e localização

Outras Computações

- Computação definida pelo programador

  MPI_Op_create (user_fcn, commutes, &op);

  MPI_Op_free (&op);

  user_fcn (invec, inoutvec, len, datatype)

onde *user_fcn* implementa

inoutvec[i] = invec[i] op inoutvec[i]; para *i=0,...,len-1*

e *commutes* indica se *user_fcn* é comutativa ou não

Coletivas sem Bloqueio

- MPI-3 define versões sem bloqueio das coletivas
  - Mesmo conceito de *MPI_Send/MPI_Isend/MPI_Wait*
  - Exemplo: Broadcast sem bloqueio

*MPI_Ibcast(void *buffer, int count, MPI_Datatype datatype, int root, MPI_comm comm, MPI_Request *request)*

  - Retorno da chamada é imediato
  - Conclusão da operação pode ser tratada com *MPI_Wait*, *MPI_Test*, etc.
  - Parâmetro *request* pode ser combinado com outras chamadas

Barreiras sem Bloqueio

- Também introduzidas em MPI-3

*MPI_Ibarrier(MPI_comm comm, MPI_Request *request)*

  - Retorno da chamada é imediato
  - Objetivo: informar aos demais ranks sobre a chegada à barreira, sem necessariamente ficar preso nela

Exemplo de Uso:

```
<...tarefa essencial...>
MPI_Ibarrier(...,&request);
flag=false;
while (flag==false) {
        <...tarefa não-essencial...>
        MPI_Test(&request, &flag, &status);
}
```