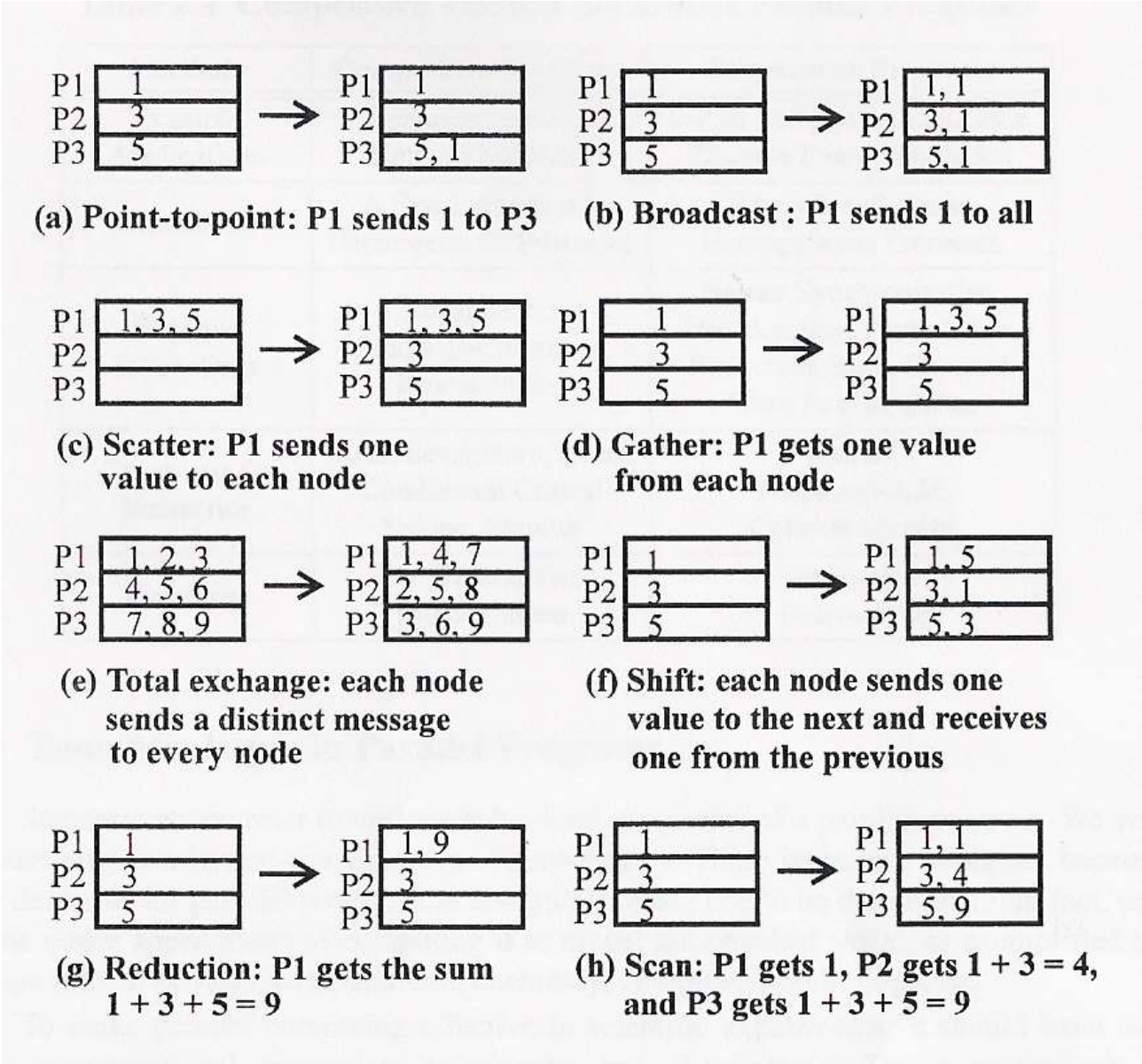
Comunicação coletiva

Um-para-um, Um-para-N, N-para-um e N-para-N



Livro Hwang pag 82 e 83 seção 2.4.3 Figura 2.10

1.1.1 Primitivas coletivas bloqueantes

Todas são feitas para os processos de um domínio de comunicação.

Barrier: Usada para a sincronização dos processos, onde cada processo de um domínio de comunicação aguarda a chegada dos demais processos para poder continuar (comunicação n-to-n).

int MPI_Barrier(MPI_Comm comm)

Broadcast: Envia a mesma mensagem para todos os demais processos do domínio de comunicação (1-to-n).

int MPI_Bcast(void *buf, int count, MPI_Datatype datatype, int source, MPI_Comm comm)

Gather: Coleta informações dispersas entre todos os processos e as reúne em um único processo (n-to-1).

int MPI_Gather(void *sendbuf, int sendcount, MPI_Datatype senddatatype, void *recvbuf, int recvcount, MPI_Datatype recvdatatype, int target, MPI_Comm comm)

Juntando a semantica da Gather com a Broadcast.
int MPI_Allgather(void *sendbuf, int sendcount, MPI_Datatype senddatatype, void *recvbuf, int recvcount, MPI_Datatype recvdatatype, MPI_Comm comm)

Scatter: Espalha informações concentradas em um processo sobre os demais processos de um domínio de comunicação (1-to-n)

int MPI_Scatter(void *sendbuf, int sendcount, MPI_Datatype senddatatype, void *recvbuf, int recvcount,
MPI_Datatype recvdatatype, int source, MPI_Comm comm)

A função abaixo envia quantidades diferentes de dados para cada processo destino.

int MPI_Scatterv(void *sendbuf, int *sendcounts, int *displs, MPI_Datatype senddatatype, void *recvbuf, int recvcount,
MPI_Datatype recvdatatype, int source, MPI_Comm comm)

All-to-All: União das semânticas do gather e scatter.

int MPI_Alltoall(void *sendbuf, int sendcount, MPI_Datatype senddatatype, void *recvbuf, int recvcount, MPI_Datatype recvdatatype, MPI_Comm comm)

Redução: Associa uma computação à comunicação. Algumas funções possíveis são: máximo, mínimo, soma, produto, E/OR/XOR lógico e E/OR/XOR bit-a-bit.
int MPI_Reduce(void *sendbuf, void *recvbuf, int count, MPI_Datatype datatype, MPI_Op op, int target, MPI_Comm comm)

Prefix:Faz uma operação de redução com todos os processos de um domínio de comunicação, no entanto, distribui resultados parciais entre todos os processos de acordo com o rank. As operações possíveis são as mesmas das operações de redução.

int MPI_Scan(void *sendbuf, void *recvbuf, int count, MPI_Datatype datatype, MPI_Op op, MPI_Comm comm)

1.1.2 Primitivas Coletivas não bloqueantes no MPI

Antes da versão 3 do MPI, todas as comunicações coletivas eram bloqueantes. Agora elas possuem funções análogas, porém, são não bloqueantes. Por exemplo, há uma barreira não bloqueante. Os processos retornam mesmo que os demais não tenham chegado à barreira. O

programador consegue recuperar esta situação e é permitido ao programa prosseguir. Cabe ao programador determinar o que pode ser feito enquanto a barreira aguarda a chegada dos demais processos.

int MPI_Ibarrier(MPI_Comm comm, MPI_Request *request)

As funções MPI_Test e MPI_Wait são usadas para verificar o fim das operações coletivas não bloqueantes.

Códigos exemplo: em 03a-primitivas-coletivas. Códigos numerados na sequência.

Exercícios solicitados: Números perfeitos usando apenas mensagens coletivas.
Código disponível em prog-MPI/mpi_examples/3c-numperf-coletivas