Universidad Nacional de San Agustín

Escuela Profesional de Ciencia de la Computación

Comunicación Punto a Punto

Alumno:
Eddy Caceres Huacarpuma

5 de abril de 2017

Índice general

1. Introducción **2**
1.1. Modos de Comunicación 3
1.1.1. Blocking Synchronous 4
1.1.2. Blocking Ready 4
1.1.3. Buffered Send 5
1.1.4. Standard Send 5
1.2. Comunicación punto a punto no-bloqueante 6
1.2.1. Sintaxis 6

Capítulo 1

Introducción

Introducción

La comunicación punto a punto es la transferencia de un mensaje de un proceso específíco a otro . El patron punto a punto requiere acción tanto del proceso emisor y receptor.

En el presente documento se tocaran los tipos de comunicación existentes en MPI . Las diferencias entre estos tipos afectan directamente el rendimiento del programa.

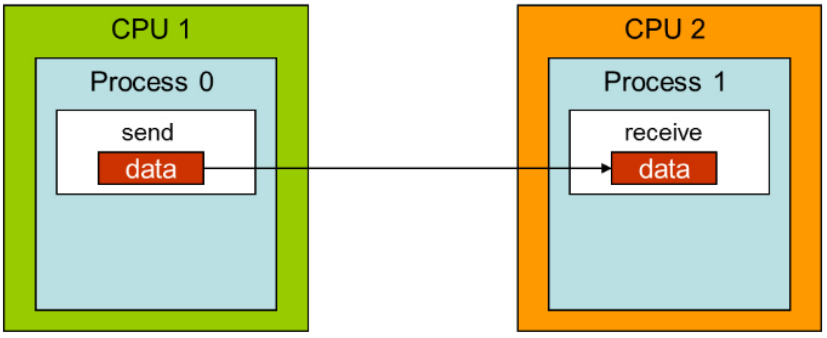


Figura 1.1: Ilustración de la comunicación

Comunicación Bloqueante

Tambien llamado seguro,

Tanto en el modo enviando(sending) y recibiendo (receiving), el buffer que se usa para contener el mensaje puede ser un recurso frecuentemente utilizado, y los datos podrían corromperse cuando se le utiliza antes que de una transacción en marcha sea completada, la comunicación bloqueante se asegura que esto nunca suceda.

Entonces la comuncion bloqueante suspende la ejecución hasta que el buffer de mensaje es seguro de usar.

Comunicación No-bloqueante

Una llamada no-bloqueante garantiza una interrupción cuando la transacción esta lista para continuar, permitiendo de esta manera al hilo original volver al proceso orientado al cálculo.

Entonces la comunicación no-bloqueante separa la comunicación del cálculo

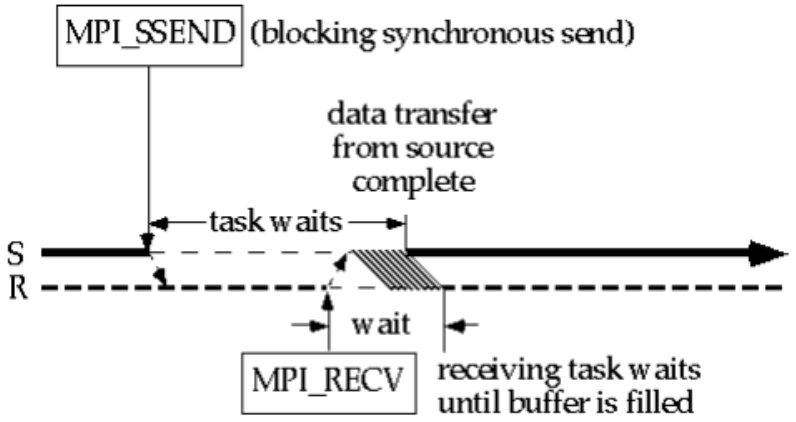
1.1. Modos de Comunicación

El modo de comunicación es seleccionado con la rutina de envio (send routine). Existen 4 rutinas bloqueantes y 4 rutinas no-bloqueantes. Las rutinas de recibo(recieve ruotine) no especifican el modo de la comunicación.

Communication Mode	Blocking Routines	Non-Blocking Routines
Synchronous	MPI_SSEND	MPI_ISSEND
Ready	MPI_RSEND	MPI_IRSEND
Buffered	MPI_BSEND	MPI_IBSEND
Standard	MPI_SEND	MPI_ISEND
	MPI_RECV	MPI_IRECV
	MPI_SENDRECV	
	MPI_SENDRECV_REPLACE	

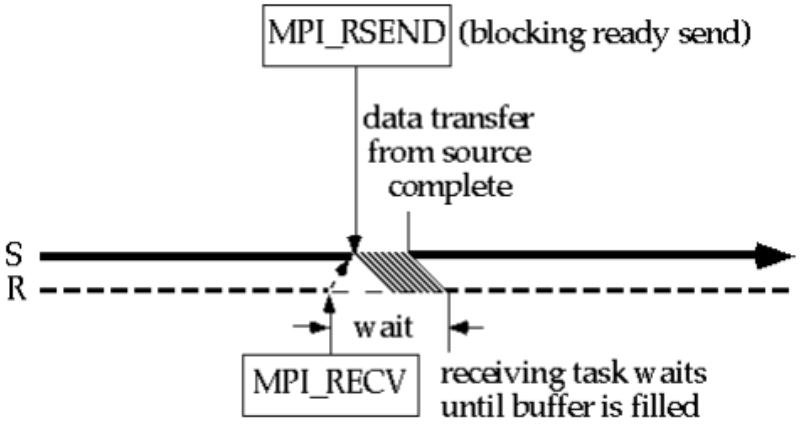
1.1.1. Blocking Synchronous

Cuando el Blocking Synchronous Send MPI_SSend se ejecuta, la tarea que hace el envío le indica al receptor que tiene un mensaje para él y espera a que el receptor le envíe un mensaje indicándole que está listo para recibir el mensaje. Entonces los datos son transferidos.



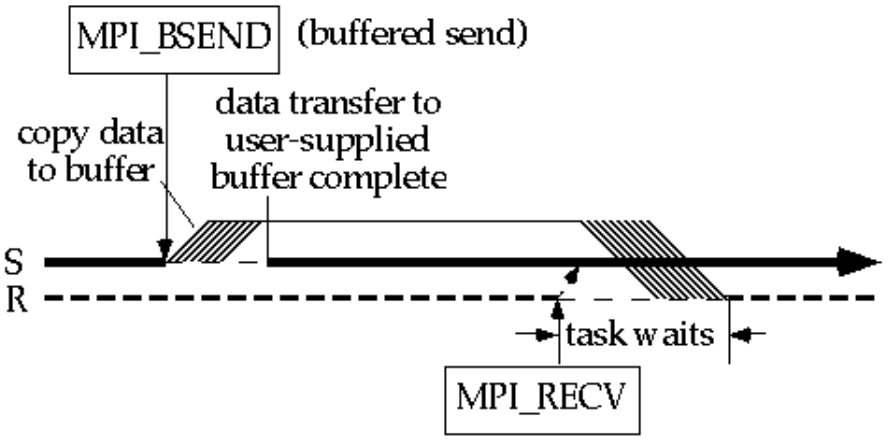
1.1.2. Blocking Ready

Este modelo busca reducir los costos de sistema y sincronización asumiendo que el mensaje de "listo para recibir"(ready-to-receive) ya ha llegado. Si esa notificación no ha llegado entonces ocurre un error.



1.1.3. Buffered Send

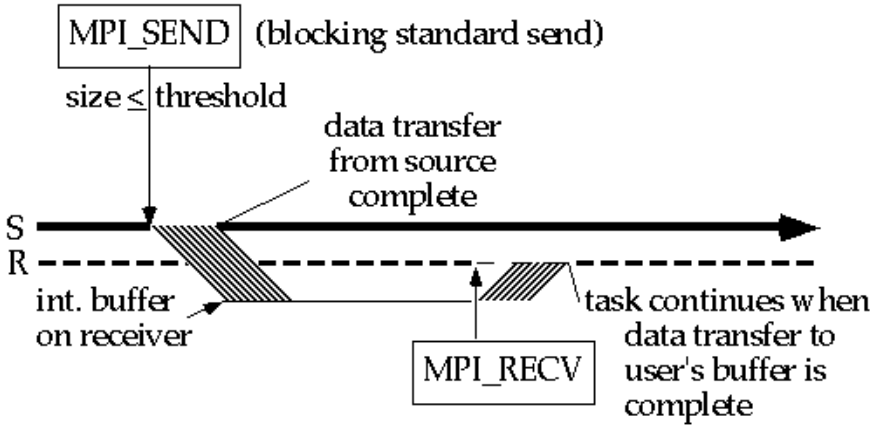
El Blocking buffered send MPI_BSend copia los datos del buffer de mensajes a un buffer suministrado por el usuario, y entonces retorna el control. Los datos pueden ser copiados desde este buffer suministrado por el usuario en red una vez que la notificación "listo para recibir "ha llegado.



1.1.4. Standard Send

Este modo de comunicacion es el más dificil de definir. Podriamos decir siplemente que es más optimo ya que trata de sacar ventaja de específicas optimizaciones que pueden estar disponibles a traves de mejoras del sistema.

La estrategia es tratar largos y pequeños mensajes de forma diferente. Si los mensajes son largos, la penalidad de tener copias extra en un bufer puede ser execesiva y podria matar el programa. Sin embargo, el almacenamiento en búfer puede llegar a ser beneficioso para los mensajes pequeños . Este modo de comunicación trata de encontrar un balance.



1.2. Comunicación punto a punto no-bloqueante

Un send/receive bloqueante suspende la ejecución del programa hasta que el buffer que se está enviando/recibiendo es seguro de usar. En el caso de un send bloqueante, esto significa que los datos a ser enviados han sido copiados al buffer de envío (no necesariamente han sido recibidos por la tarea que recibe).

El estado del dato trasnferido y el exito de la comunicacion debe ser verificda en la ultima parte del programa usando las funciones MPI_Wait or MPI_Test.

Las llamadas no bloqueantes terminan inmediatamente después de ser iniciada la comunicación.

1.2.1. Sintaxis

Las llamadas no-bloqueantes tienen la misma sintaxis que las boqueantes solo con dos exepciones:

- Se antepone la letra " I " eg MPI_IRecv
- El arguemento final es identificador de un obejto de peticion que tiene información detallada de la transacción.

Conclusion

A continuación se muestra un cuadro de comparación que resume todo lo explicado

Mode	Advantages	Disadvantages
Synchronous	- Safest, therefore most portable - No need for extra buffer space - SEND/RECV order not critical	- Can incur substantial synchronization overhead
Ready	- Lowest total overhead - No need for extra buffer space - SEND/RECV handshake not required	- RECV *must* precede SEND
Buffered	- Decouples SEND from RECV - no sync overhead on SEND - Programmer can control size of buffer space - SEND/RECV order irrelevant	- Copying to buffer incurs additional system overhead
Standard	- Good for many cases - Compromise position	- Protocol is determined by MPI implementation