**Resolución de la 1ra. Práctica Calificada del curso de Programación Paralela**

**Teoría**

1. **Explique con sus palabas ¿Qué es un proceso de una computadora?**

Es la ejecución de un programa que posee una secuencia de instrucciones, un estado actual y un conjunto de recursos asociados.

1. **Explique a que se refiere cuando hablamos de una comunicación punto a punto entre 2 procesos, proponer un ejemplo en código.**

Es la comunicación que se da entre 2 procesos, el envió de mensaje el cual lo hace el MPI\_Send () y el receptor el cual es el MPI\_Recv (),

Veamos un código:

…

If (my\_rank != 0){

sprintf (msg, “Saludos del proceso %d de %d!”, my\_rank, comm\_sz);

MPI\_Send(msg, strlen(msg) + 1, MPI\_CHAR, 0, 0, MPI\_COMM\_WORD);

} else {

printf (“Saludos del proceso %d de %d!\n, my\_rank, comm\_sz ”);

for ( int q=1; q < comm\_sz; q++) {

MPI\_Recv( msg, MAX\_STRING, MPI\_CHAR, q, 0, MPI\_COMM\_WORD, MPI\_STATUS\_IGNORE);

`printf (“%s\n”, msg);

}

}

….

1. **¿Qué es una memoria RAM (principal), Cache y Virtual? E indicar ¿Cómo funcionan?**

|  |  |  |
| --- | --- | --- |
| Memoria RAM | Cache | Memoria Virtual |
| Memoria RAM (Memoria de acceso aleatorio).  Es una colección de posiciones.  Es una memoria volátil.  Función:  Almacena información y luego es enviado las instrucciones y/o datos al procesador. | También llamada RAM caché  Es parte de la memoria RAM estática de alta velocidad(SRAM)  Función:  Se encarga de registrar las instrucciones comúnmente ejecutadas de una aplicación. | Es una simulación de la memoria RAM.  Se utiliza el principio de Localidad, swap- space y páginas.  Función:  Reserva un espacio en el disco duro para poder ser usados en el procesador. |

1. **¿En qué consiste la programación en Memoria Distribuida y la programación en Memoria Compartida?**

|  |  |
| --- | --- |
| Programación en  Memoria Distribuida | Programación en Memoria Compartida |
| Cada proceso tiene su propio espacio de memoria.  Los procesadores pueden compartir información solamente enviando mensajes. | Todos los procesos comparten una única memoria.  Cada proceso tiene acceso a toda la memoria. |

1. **Describa en 3 lineas como máximo e indicar los parámetros de los siguientes comandos del MPI:**
2. MPI\_Send () : Es aquel que envía mensajes a un proceso con parámetros

MPI\_Send (

void\* msg\_buf REFERENCIA

Int msg\_size CANTIDAD

MPI\_Datatype datatype TIPO DE DATO

Int dest RANK DESTINO

Int tag ETIQUETA

MPI\_Comm) communicator COMUNICADOR

1. MPI\_Recv (): Es aquel que capta el mensaje del proceso

MPI\_Recv (

void\* msg\_buf REFERENCIA DATO

Int buf\_size CANTIDAD

MPI\_Datatype buf\_type TIPO DE DATO

Int source RANK DESTINO

Int tag ETIQUETA

MPI\_Comm communicator COMUNICADOR

MPI\_Status) status ESTADO

1. MPI\_Reduce (): Toma una matriz de elementos de entrada en cada proceso y devuelve una matriz de salida.

MPI\_Reduce (

void\* imput\_data REFERENCIA DATO

void\* output\_data REREFERENCI LLEG

int count CANTIDAD / DATOS

MPI\_Datatype buf\_type TIPO DE MENSAJE

MPI\_Op operator OPERACION

Int dest\_proc DESTINO / PROCESO

MPI\_Comm) comm COMUNICADOR

1. MPI\_Allreduce (): Todos los procesos necesitan una reducción para poder continuar.

MPI\_Allreduce (

void\* imput\_data REFERENCIA DATO

void\* output\_data REFERENCIA LLEG

int count CANTIDAD / DATOS

MPI\_Datatype buf\_type TIPO DE MENSAJE

MPI\_Op operator OPERACION

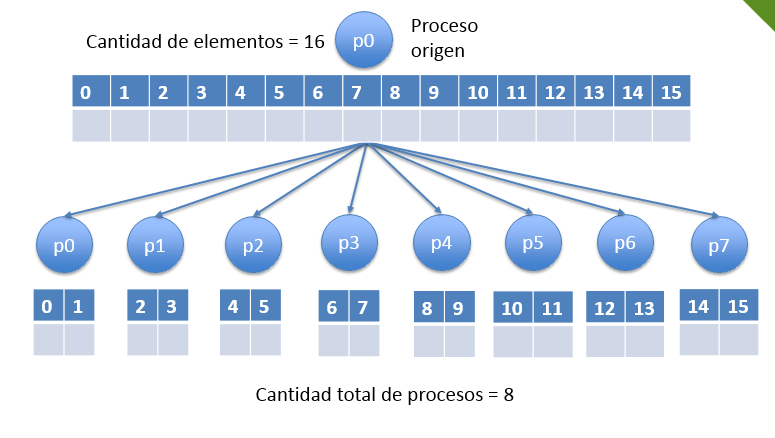
MPI\_Comm) comm COMUNICADOR

**Práctica**

**8) Suponga que comm\_sz=8 y la cantidad de elementos es n=16**

a) Diseñe un diagrama que explique cómo MPI\_Scatter puede ser implementado usando comunicaciones basadas en árboles. Puede suponer que el origen del Scatter es el proceso con Rank 0.

MPI\_Scatter (): Toma una lista y lo distribuye equitativamente a cada proceso incluyéndose



b) Hacer lo mismo para el MPI\_Gather, en este caso con el proceso 0 como destino.

MPI\_Gather (): Entrega el valor y lo concatena todos los elementos de los procesos, lo recibe en orden mientras lo une.

