

Grado en Ingeniería Informática

Arquitectura de Computadores

Práctica 4

Contenido

Práctica 4: Resolución de problemas mediante el modelo de comunicación colectivo.....	3
Introducción.....	3
MPI_Allgather.....	3
MPI_Alltoall	3
Objetivos	4
Compilación y ejecución de programas MPI	4
Entregables.....	4
Ejercicio 1	4
Ejercicio 2	5
Entrega.....	6

Práctica 4: Resolución de problemas mediante el modelo de comunicación colectivo

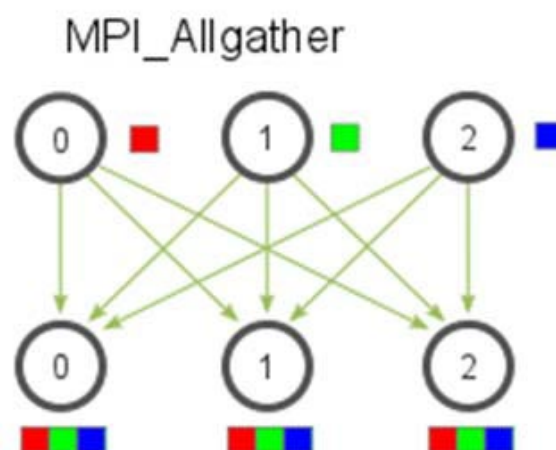
Introducción

La presente práctica continúa profundizando en el modelo de comunicación colectivo que fue presentado en la práctica anterior. Para ello se propone la resolución del problema de la transposición de matrices desde la perspectiva del paralelismo: El problema de la transposición de matrices mediante solución paralela: $B=A^T$

MPI_Allgather

Dado un conjunto de datos distribuidos entre los diferentes nodos del comunicador, todos los datos se recogerán en todos los nodos. MPI_Allgather es un MPI_Gather seguido por un MPI_Bcast.

```
MPI_Allgather( void* send_data, int send_count, MPI_Datatype send_datatype,
void* recv_data, int recv_count, MPI_Datatype recv_datatype, MPI_Comm
communicator)
```

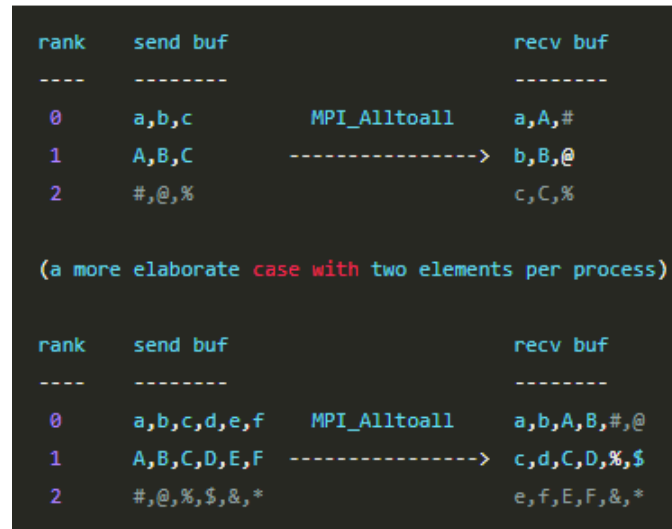


rank	send buf		recv buf
---	-----		-----
0	a,b,c	MPI_Allgather	a,b,c,A,B,C,#,@,%
1	A,B,C	----->	a,b,c,A,B,C,#,@,%
2	#,@,%		a,b,c,A,B,C,#,@,%

MPI_Alltoall

Funciona como MPI_Scatter y MPI_Gather combinados. El búfer de envío en cada proceso se divide como en MPI_Scatter y luego cada columna de fragmentos se recopila mediante el proceso respectivo, cuyo rango coincide con el número de la columna de fragmentos.

```
int MPI_Alltoall(const void *send_data, int send_count,
MPI_Datatype send_datatype, void *recv_data, int recv_count,
MPI_Datatype recv_datatype, MPI_Comm communicator)
```



Objetivos

En esta práctica se aprenderá el concepto de comunicación colectivo entre procesos de MPI.

Los objetivos fijados son los siguientes:

- Compilación de programas MPI.
- Ejecución de programas en varios procesos de forma paralela.
- Estructura de un programa MPI.
 - Iniciar y finalizar el entorno MPI con MPI_Init y MPI_Finalize.
 - Identificador (rango) del proceso con MPI_Comm_rank.
 - Consultar el número de procesos lanzados con MPI_Comm_size.
- Aprender operaciones de comunicación colectivas:
 - MPI_Allgather
 - MPI_Alltoall

Compilación y ejecución de programas MPI

Compilación: `mpicc codigo_fuente.c -o ejecutable`

Ejecución: `mpirun -np <number> ejecutable`

Entregables

- Memoria
- Código fuente de los siguientes ejercicios:

Ejercicio 1

Implementar un programa que realice la transposición de la matriz inicial mostrada en la figura (parte izquierda).

Realizar el proceso en paralelo, distribuyendo la matriz entre 4 procesos.

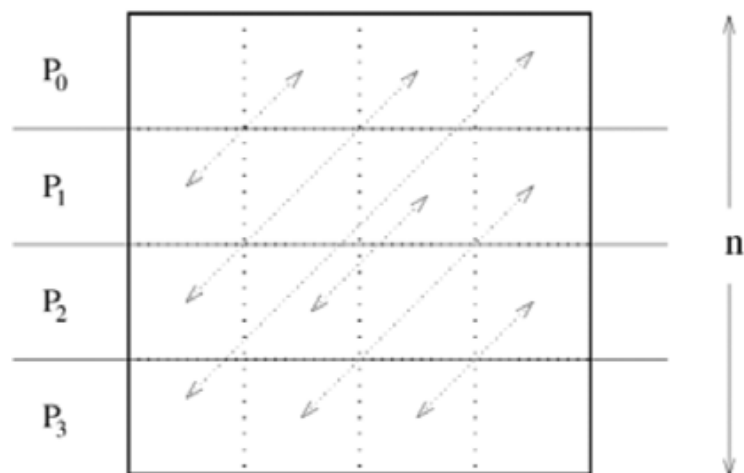
		Data partition			
		0	1	2	3
Process	0	1	2	3	4
	1	5	6	7	8
	2	9	10	11	12
	3	13	14	15	16

		Data partition			
		0	1	2	3
Process	0	1	5	9	13
	1	2	6	10	14
	2	3	7	11	15
	3	4	8	12	16

Ejercicio 2

Implementar un programa donde se resuelva el problema de la transposición de matrices cuadradas de **dimensión arbitraria** $N \times N$ (siendo N múltiplo del número de P de procesos). La matriz se dividirá entre P procesos ($p \ll N$) que la inicializarán, de tal modo que cada elemento sea único (para ello usar la posición de la fila y la columna y el número de rank, como por ejemplo: $1000 * i + j + n/p * rank$).

Para ello, distribuir A y B por columnas (o filas), entre los procesos y haciendo uso de las operaciones colectivas que se consideren oportunas llevar a cabo el proceso de transposición.



La siguiente función se puede utilizar para hacer la transposición de una matriz dada:

```
void transpuesta (double *a, int n) {
/* transpuesta de una matriz cuadrada a, de dimensión nxn */

    int i, j;
    int ij, ji, l;
    double tmp;
    ij = 0;
    l = -1;
    for (i = 0; i < n; i++) {
        l += n + 1;
        ji = l;

```

```

        ij += i + 1;
        for (j = i+1; j < n; j++){
            tmp = a[ij];
            a[ij] = a[ji];
            a[ji] = tmp;
            ij++;
            ji += n;
        }
    }
}

```

Entrega

Es imprescindible subir los entregables a la actividad del **campus** y enviármelos por **correo** (lrodriguezso@nebrija.es) antes del **9 de diciembre a las 7:59 am**.