

- 1- En el programa de cálculo de fractales. ¿cómo indica el maestro a los trabajadores que ya no hay más trabajo?

Respuesta: Dándoles un número de fila a hacer mayor de lo esperado

- 2- En el último apartado de la práctica (sistemas de ecuaciones lineales), la distribución de la matriz más eficiente es:

Respuesta: Cíclica por filas, porque equilibra mejor la carga computacional

- 3- Indica cuál de las siguientes afirmaciones relativas a la práctica de sistema de ecuaciones lineales es correcta:

Respuesta: En el caso de la distribución **cíclica por filas**, un proceso puede averiguar el índice local de una fila de la matriz global de la que es propietario calculando el **cociente entre el índice de la fila global y el número de procesos**

Alternativas al ejercicio 3 (por si cambian las afirmaciones):

Respuesta: En el caso de la distribución **cíclica por filas**, para averiguar a qué proceso le corresponde una fila i de la matriz global A , hay que calcular el **módulo entre el índice de la fila global y el número de procesos**.

Respuesta: En el caso de la distribución por **bloques de fila**, para averiguar a qué proceso le corresponde una fila i de la matriz global A , hay que calcular el **cociente entre el índice de la fila global y el tamaño de bloque**.

Respuesta: En el caso de la distribución por **bloques de fila**, un proceso puede averiguar el índice local de una fila de la matriz global de la que es propietario calculando el **módulo entre el índice de la fila global y el tamaño de bloque**.

- 4- En el programa de cálculo de fractales sin el maestro trabajando, si ejecutamos con p procesos y la imagen tiene H filas, cada trabajador va a calcular este número de filas:

Respuesta: No se puede saber a priori el número de filas que hará cada uno.

- 5- Partiendo de un vector repartido por bloques de n/p componentes consecutivas en p procesos, una forma de obtenerlo completo y de forma replicada en todos los procesos sería:

Respuesta: Haciendo una operación gather seguida de una bcast

- 6- En el algoritmo paralelo de la práctica del producto matriz-vector, versión con distribución por bloques de filas, tras cada iteración del método, el vector x ...:

Respuesta: Debe quedar replicado completo en todos los procesos

- 7- Para juntar en el proceso 0 una matriz A de n filas y n columnas, con elementos de tipo double, previamente repartida por bloques de filas entre p procesos en una submatriz Aloc, y suponiendo n divisible entre p , utilizamos:

Respuesta: `MPI_Gather(Aloc, n*n/p, MPI_DOUBLE, A, n*n/p, MPI_DOUBLE, 0, MPI_COMM_WORLD);`

- 8- En la sustitución de la operación reducción: `MPI_Reduce(&mypi, &pi, 1, MPI_DOUBLE, MPI_SUM, 0, MPI_COMM_WORLD);`, mediante operaciones punto a punto del ejercicio del cálculo de PI, indica qué respuesta es **correcta**:

Respuesta: Los procesos con identificador diferente de 0 deberán ejecutar cada uno, una única operación de envío (`MPI_Send`) de la variable `mypi` al proceso 0.

Alternativas al ejercicio 8 (por si cambian las afirmaciones):

Respuesta: En el proceso 0, el valor recibido de cada uno de los procesos deberá ser acumulado en la variable `pi`.

Respuesta: En el proceso 0 deberá realizar mediante un bucle tantas operaciones de recepción (`MPI_Recv`) como procesos se han indicado en la ejecución **menos uno**.

Respuesta: Todos los procesos menos el 0 deberán enviar mediante la primitiva `MPI_Send` el contenido de la variable "`mypi`" al proceso 0.

Respuesta: El número de filas que le corresponderá a cada proceso **puede no ser** el mismo, usando una distribución por bloques de filas o cíclica por filas.

(diapo 15 párrafo 2)

- 9- Para repartir el proceso 0 una matriz A de n filas y n columnas con elementos de tipo double, por bloques de filas entre p procesos, quedando recibida en una matriz Aloc, y suponiendo n divisible entre p , utilizaremos:

Respuesta: `MPI_Scatter(A, n/p*n, MPI_DOUBLE, Aloc, n/p*n, MPI_DOUBLE, 0, MPI_COMM_WORLD)`

10- En el código original newton.c, indica la **respuesta falsa**:

Respuesta: El proceso maestro envía, mediante la variable num_row, el índice de fila que debe ser procesada por un proceso trabajador.

11- En el código original newton.c, indica la **respuesta correcta**:

Respuesta: Los procesos trabajadores almacenan las filas que calculan en el vector B.

Alternativas al ejercicio 11 (por si cambian las afirmaciones):

Respuesta: Cuando el proceso maestro recibe una fila de un proceso trabajador, el índice de fila queda almacenado en la variable status.

Respuesta: Los procesos trabajadores reciben en la variable num_row el índice de fila que deben procesar.

12- Dado el siguiente fichero de trabajo, calcularemos el valor de Pi empleando:

```
#!/bin/sh
#PBS -l nodes=4:ppn=8,walltime=00:10:00
#PBS -W x="NACCESSPOLICY:SINGLEJOB"
#PBS -q cpa
#PBS -d .
mpiexec ./mpi_pi
```

4 nodos * 8 procesos = 32

Respuesta: 32 procesos

13- En el programa para el producto matriz vector paralelo:

Respuesta: Al expandir la matriz M a n2 filas o columnas, hemos podido simplificar la distribución de dicha matriz entre los procesos pudiendo utilizar MPI_Scatter en lugar de MPI_Scatterv.

Alternativas al ejercicio 13 (por si cambian las afirmaciones):

Respuesta: Si no se expandiera la matriz M hasta n2 filas o columnas, habría que sustituir el MPI_Scatter por MPI_Scatterv para su distribución entre los procesos.

14- ¿Qué obtenemos al ejecutar el programa ping-pong (completado correctamente) usando el siguiente fichero de trabajo?:

```
#!/bin/sh
#PBS -l nodes=2,walltime=00:10:00
#PBS -q cpa
#PBS -d .
mpiexec ./ping-pong 10000
```

Respuesta: El tiempo de transmisión entre dos nodos del clúster de 100 mensajes de 10000 bytes.

15- Si enviamos el siguiente fichero de trabajo al sistema de colas:

```
#PBS -l nodes=3:ppn=4,walltime=00:10:00
#PBS -W x="NACCESSPOLICY:SINGLEJOB"
#PBS -q cpa
#PBS -d .
mpiexec ./mpi_pi
```

Respuesta: Emplearemos 12 procesos para calcular el valor de pi, usando 3 nodos del clúster y lanzando 4 procesos en cada nodo.

16- En el último apartado de la práctica (sistemas lineales), la distribución de la matriz más eficiente:

Respuesta: Cíclica por filas, porque todos los procesos tienen trabajo durante todo (o casi todo) el cálculo paralelo.

17- En el código original newton.c, cuando el proceso maestro recibe una fila de un proceso trabajador, ¿cómo sabe el proceso cuál es el índice de fila recibida?:

Respuesta: Mediante la variable status

18- En la práctica de fractales, se cambia el programa original para que el maestro...:

Respuesta: Haga recepciones no bloqueantes y así pueda calcular filas mientras espera resultados de los trabajadores.

19- En el programa facilitado para realizar la sesión 4, la matriz del sistema:

Respuesta: Se aumenta en tamaño para poder distribuirla mediante MPI_Scatter.

20- En el esquema mostrado en la figura 11, que corresponde a la paralelización de la descomposición LU (función lu):

Respuesta: En el cálculo de la columna k participan todos los procesos en paralelo.

21- Sea i un número de fila de 0 a $nN - 1$ perteneciente a una matriz distribuida por bloques de tamaño nBloque filas entre nP procesos, entonces:

Respuesta: El valor de $i/nBloque$ representa el proceso que posee dicha fila.

22- Dado el siguiente script para lanzar un trabajo en el cluster kahan:

```
#!/bin/sh
#PBS -l nodes=8,walltime=00:10:00
#PBS -q cpa
#PBS -d .
mpiexec miprograma
```

Respuesta: El programa no se podría ejecutar por no haber suficientes nodos en kahan.

23- En el programa facilitado para realizar la práctica, el vector de términos independientes...:

Respuesta: Es sobrescrito por el vector solución

24- En la función que realiza la factorización LU, el paralelismo se consigue haciendo que se calculen simultáneamente:

Respuesta: Distintas iteraciones del bucle i dentro de una misma iteración del bucle k.

25- Sea el siguiente fragmento de código, en el que id_pr contiene el identificador de proceso y np es el número de procesos:

```
if (id_pr == 0) {
    for (j = 0; j < nfil; j++) al[j] = a[j];
    for (p= 1; p < np; p++) {
        MPI_Send(&a[p*nfil], nfil, MPI_INT, p, 10, MPI_COMM_WORLD);
        MPI_Send(y, n, MPI_INT, p, 20, MPI_COMM_WORLD);
    }
}
else {
    MPI_Recv(al, nfil, MPI_INT, 0, 10, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
    MPI_Recv(y, n, MPI_INT, 0, 20, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
}
```

¿Cuál de los siguientes fragmentos de código es equivalente al anterior?

Respuesta: MPI_Scatter(a,nfil,MPI_INT,al,nfil,MPI_INT,0,MPI_COMM_WORLD);
MPI_Bcast(y,n,MPI_INT,0,MPI_COMM_WORLD);

26- En la práctica de dibujo de fractales mediante el esquema maestro-trabajadores, ¿cómo avisa el maestro a cada trabajador de que ya no tiene que hacer nada más?:

Respuesta: Enviándole un número de fila a hacer fuera de rango.

27- Indica qué operación colectiva de MPI se parece más a lo que realiza este código:

```
if ( yo == 0 ) {
    for ( proc = 1 ; proc < num_procs ; proc++ ) {
        MPI_Send(buf, count, MPI_DOUBLE, proc, 1234, MPI_COMM_WORLD);
    }
} else {
    MPI_Recv(buf, count, MPI_DOUBLE, 0, 1234, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
}
```

Respuesta: MPI_Bcast

28- El tipo de dato

MPI_Type_vector(nBloque, nN, nP*nN, MPI_DOUBLE, &NUEVO_TIPO);
podría ser usado en el programa de la práctica:

Respuesta: En el caso de distribución cíclica, para enviar a un proceso un mensaje con todas las filas de la matriz que le tocan.

29- En una distribución cíclica por filas de una matriz A de $n \times n$, dado un índice de fila global i , ¿cómo se calcularía el proceso propietario de esa fila y el índice local de la fila? Asumimos que nN es un múltiplo exacto de nP (el número de procesos):

Respuesta: Propietario = $i \% nP$, índice local = i / nP

30- Sea el siguiente fragmento de código:

```
dest = (rank+1) % p;  
src = (rank+p-1) % p;  
MPI_Send(&val, 1, MPI_DOUBLE, dest, 0, MPI_COMM_WORLD);  
MPI_Recv(&val, 1, MPI_DOUBLE, src, 0, MPI_COMM_WORLD, &status);
```

En donde rank es el identificador del proceso y p es una variable que indica el número de procesos. ¿Qué tipo de transferencia del dato val se realiza?

Respuesta: Se trata de un desplazamiento en anillo

31- Dado el siguiente fragmento de código, donde rango contiene el identificador de proceso y p el número de procesos

```
if (rango==p-1) {  
    for (i=0; i<p-1; i++) {  
        MPI_Send(&x[i*n/p], n/p, MPI_DOUBLE, i, tag, comm);  
    }  
    for (i=0; i<n/p; i++) xloc[i] = x[n-n/p+i];  
} else {  
    MPI_Recv(xloc, n/p, MPI_DOUBLE, p-1, tag, comm, &stat);
```

El resultado sería equivalente a la siguiente instrucción por parte de todos los procesos:

Respuesta: `MPI_Scatter(x, n/p, MPI_DOUBLE, xloc, n/p, MPI_DOUBLE, p-1, comm);`

32- El siguiente fragmento de código MPI, suponiendo que se han lanzado 2 procesos,

```
if (my_rank == 0) MPI_Bcast(buf, count, MPI_INT, 1, MPI_COMM_WORLD);  
else MPI_Bcast(buf, count, MPI_INT, 0, MPI_COMM_WORLD);
```

Respuesta: Se produce un interbloqueo, ya que los dos quedan esperando al otro en la operación colectiva

33- Se tienen 4 procesos MPI con las siguientes variables enteras en cada uno de los procesos:

```
rank=0, a=12, b=-1
rank=1, a=18, b=-1
rank=2, a=22, b=-1
rank=3, a=8, b=-1
```

¿Cuál es el resultado de la siguiente operación?

```
MPI_Reduce(&a, &b, 1, MPI_INT, MPI_MAX, 1, MPI_COMM_WORLD);
```

Respuesta:

```
rank=0, a=12, b=-1
rank=1, a=18, b=22
rank=2, a=22, b=-1
rank=3, a=8, b=-1
```

34- Dado el siguiente fragmento de código:

```
if (rank == 0) prev = MPI_PROC_NULL;
else prev = rank-1;
if (rank == p-1) next = MPI_PROC_NULL;
else next = rank+1;
```

```
int a = rank;
```

```
MPI_Ssend( &a, 1, MPI_INT, next, 111, MPI_COMM_WORLD );
```

```
MPI_Recv( &a, 1, MPI_INT, prev, 111, MPI_COMM_WORLD, &status );
```

```
printf("P%d: a = %d\n",rank,a);
```

y suponiendo que las variables están convenientemente declaradas, elegir la respuesta correcta suponiendo que se ejecuta el programa con 4 procesos.

Respuesta:

La salida será:

P0: a = 0

P1: a = 0

P2: a = 1

P3: a = 2

35- El siguiente fragmento de código MPI, suponiendo que han lanzado 10 procesos:

```
if (my_rank < 5)
    MPI_Bcast(buf, count, MPI_INT, 0, MPI_COMM_WORLD);
```

Respuesta: No funcionará correctamente, ya que no todos los procesos ejecutan la operación colectiva.

36- En la práctica de dibujo de fractales mediante el esquema maestro-trabajadores, el trabajo a realizar ha sido:

Respuesta: Cambiar las recepciones del maestro a no bloqueantes para poder ir trabajando en el maestro mientras se reciben datos.

37- Dado el siguiente script para lanzar un trabajo en el cluster kahan:

```
#!/bin/sh
#PBS -l nodes=1:ppn=8,walltime=00:10:00
#PBS -q cpa
#PBS -d .
mpiexec miprograma
```

Respuesta: Faltaría indicarle la opción -W x="NACCESSPOLICY:SINGLEJOB".