



Problema: Donat el següent programa paral·lel en MPI que calcula la integral d'una funció amb una precisió n i el resultat s'emmagatzema en tots els processos: (Dado el siguiente programa paralelo en MPI que calcula la integral de una función con una precisión n y el resultado se almacena en todos los procesos:)

```
int main(void) {
    int rank, sz, local_n, n;
    double a = 0.0, b = 5.0, h, local_a, local_b;
    double local, total;
    int source;
    MPI_Init(NULL, NULL);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);
    MPI_Comm_size(MPI_COMM_WORLD, &sz);
    if(rank==0){
        printf("Introduce la precisión del cálculo (n > 0): ");
        scanf("%d",&n);}
    MPI_Bcast(&n, 1, MPI_INT, 0, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
    h = (b-a)/n; /* h es el mismo para todos los procesos */ 2 FLOP
    local_n = n/sz; /* es el número de trapezoides */
    local_a = a + rank*local_n*h; 3 FLOP
    local_b = local_a + local_n*h; 2 FLOP
    local = Trap(local_a,local_b,local_n, h); //Coste= 5*local_n+2 FLOP
    if (rank == 0) {
        total = local;
        for (source = 1; source < sz; source++) {
            MPI_Recv(&local, 1, MPI_DOUBLE, source, 0, MPI_COMM_WORLD,
                MPI_STATUS_IGNORE);
            total += local; 1 FLOP
        }
    } else {
        MPI_Send(&local, 1, MPI_DOUBLE, 0, 0, MPI_COMM_WORLD);
    }
    if (rank == 0) //para el coste no se tiene en cuenta la E/S
        printf("La integral de %f a %f = %.15e\n", a, b, total);
    MPI_Bcast(&total, 1, MPI_DOUBLE, 0, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
    MPI_Finalize();
    return 0;
} /* main */
```

- a) Calcula el model de cost temporal aproximat d'este programa considerant per a la part computacional sols les operacions en coma flotant (cada operació costa t_a) i que per a les comunicacions es segueix el model $t_{com}(m) = t_{in} + m \cdot t_c$, on m està en bytes i $t_{in} = 50 \cdot t_c$. (Calcula el modelo de coste temporal aproximado de este programa considerando para la parte computacional sólo las operaciones en coma flotante (cada operación cuesta t_a) y que para las comunicaciones se sigue el modelo $t_{com}(m) = t_{in} + m \cdot t_c$, donde m está en bytes y $t_{in} = 50 \cdot t_c$.)

El coste temporal del programa viene dado por la suma de los costes de las operaciones en coma flotante y por las comunicaciones.

Coste de operaciones en coma flotante: Las 5 operaciones iniciales, más el cálculo de la función más la suma de los resultados parciales.

$(7 + (5(n/p) + 2 + (p-1)) \cdot t_a = (9 + 5(n/p) + (p-1)) \cdot t_a$, donde n es el tamaño del problema y p es el número de procesos.

Coste de comunicaciones: Considero las dos operaciones colectivas más las $(p-1)$ comunicaciones punto a punto,

MPI Bcast de n $\rightarrow (t_{in} + 4t_c) \log_2 p$
 MPI_Send/Recv $\rightarrow (t_{in} + 8t_c) (p-1)$
 MPI_Bcast de local $\rightarrow (t_{in} + 8t_c) \log_2 p$



La suma de lo cual nos da:

$$(54t_c + 58t_c) \log_2 p + 58t_c (p-1) = ((112) \log_2 p + 58(p-1)) t_c$$

Coste total

$$(9 + 5(n/p) + (p-1)) * t_a + ((112) \log_2 p + 58(p-1)) t_c$$

- b) Calcula el rendimiento en MFLOPS d'este programa a partir del model anterior, per a $n=1024$, si s'executa amb $p=8$ (processos=processadors) que funcionen a $F=4$ GHz ($t_a=1/F$) i la xarxa d'interconnexió té un ample de banda màxim de $BW=32$ Gb/s (velocitat efectiva, $t_c=1/BW$). ¿Quin serà el rendiment quan $n \rightarrow \infty$? (Calcula el rendimiento en MFLOPS de este programa a partir del modelo anterior, para $n=1024$, si se ejecuta con $p=8$ (procesos=procesadores) que funcionan a $F=4$ GHz ($t_a=1/F$) y la red de interconexión tiene un ancho de banda máximo de $BW=32$ Gb/s (velocidad efectiva, $t_c=1/BW$). ¿Cuál será el rendimiento cuando $n \rightarrow \infty$?)

El Rendimiento en MFLOPS se define como la relación entre el coste de operaciones en coma flotante (FLOP) del algoritmo secuencial dividido por el coste temporal del algoritmo paralelo en microsegundos (10^{-3} segundos). Si se divide por nanosegundos serían GFLOPS, etc.:

Coste secuencial del algoritmo en FLOP = $(5n + 4)$ FLOP (no se tiene en cuenta el cálculo de las variables local_a y local_b, pues no se requieren en la versión secuencial)

Coste temporal del algoritmo paralelo:

$$(9 + 5(n/p) + (p-1)) * t_a + ((112) \log_2 p + 58(p-1)) t_c$$

De los datos se deduce que:

$$t_a = 1/4\text{GHz} = 0.25 \text{ ns}$$

$$t_c = 1\text{B}/32\text{Gb/s} = 8\text{b}/32 \text{ Gb/s} = 0.25 \text{ ns}$$

Luego para $n=1024$

$$\text{Rendimiento} = \frac{5n + 4}{(9 + 5(n/p) + (p-1)) * t_a + ((112) \log_2 p + 58(p-1)) t_c} = \frac{5 * 1024 + 4}{\left(\left(9 + 5 \left(\frac{1024}{8} \right) + (8-1) \right) + ((112) \log_2 8 + 58(8-1)) \right) * 0.25 \text{ ns}} =$$

$$\frac{5 * 1024 + 4}{((9 + 5 * 128 + 7) + ((112) * 3 + 58 * 7)) * 0.25 \text{ ns}} = \frac{5124}{(656 + 742) * 0.25 \text{ ns}} = \frac{5124}{(1398) * 0.25 \text{ ns}} = \mathbf{14.66 \text{ GFLOPS}}$$

Cuando $n \rightarrow \infty$:

$$\text{Rendimiento} = \frac{5n + 4}{(9 + 5(n/p) + (p-1)) * t_a + ((112) \log_2 p + 58(p-1)) t_c} = \frac{5n}{5 \left(\frac{n}{p} \right) * 0.25 \text{ ns}} = \mathbf{4p \text{ GFLOPS}}$$

- c) Proposa una modificació del codi per a millorar la relació R/C del programa. Indica amb claredat què part del codi es substitueix, presenta el codi alternatiu i justifica la millora de la relació R/C del nou codi respecte de l'anterior. (Propón una modificación del código para mejorar la relación R/C del programa. Indica con claridad qué parte del código se sustituye, presenta el código alternativo y justifica la mejora de la relación R/C del nuevo código respecto del anterior.)

Se propone una modificación en la que las comunicaciones punto a punto se sustituyen por una operación colectiva MPI_Reduce():

```
MPI_Reduce(&local, &total, 1, MPI_DOUBLE, MPI_SUM, 0, MPI_COMM_WORLD);
```

Como después se realiza un MPI_Bcast de la variable total, se podría unir en MPI_Reduce anterior con el MPI_Bcast y realizar un MPI_AllReduce:

```
MPI_AllReduce(&local, &total, 1, MPI_DOUBLE, MPI_SUM, MPI_COMM_WORLD);
```

La relación R/C del algoritmo paralelo inicial viene dada por la expresión:

$$R/C = \frac{(9 + 5(n/p) + (p-1)) * t_a}{((112) \log_2 p + 58(p-1)) t_c} = \frac{(A + (p-1)) t_a}{(B + 58(p-1)) t_c}$$



Donde las variables A y B sustituyen a las expresiones en **negrita**. Si asumimos que $n \gg p$, se cumple que en general $R/C > 0$.

En cualquier caso, la nueva relación R/C tras introducir Reduce vendría dada por la expresión:

$$R'/c' = \frac{(9+5(n/p) + \log_2 p) \cdot t_a}{((112) \log_2 p + 58 \log_2 p) t_c} = \frac{(A + \log_2 p) t_a}{(B + 58 \log_2 p) t_c}$$

Para comprobar que se produce una mejora se debe cumplir que $R'/c' > R/c$:

$$\frac{A + \log_2 p}{B + 58 \log_2 p} > \frac{A + (p - 1)}{B + 58(p - 1)}$$

Operando tenemos que

$$(A + \log_2 p)(B + 58(p - 1)) > (A + (p - 1))(B + 58 \log_2 p) \rightarrow$$

$$AB + 58A(p - 1) + B \log_2 p + 58(p - 1) \log_2 p > AB + 58A \log_2 p + (p - 1)B + 58(p - 1) \log_2 p \rightarrow$$

$$58A(p - 1) + B \log_2 p > 58A \log_2 p + (p - 1)B \rightarrow$$

$$58A((p - 1) - \log_2 p) > B((p - 1) - \log_2 p) \rightarrow$$

Dado que para $p > 2$ la expresión $(p - 1) - \log_2 p > 0$ podemos simplificar la ecuación, quedándonos que $58A > B$, y por la asunción inicial suponiendo que $n \gg p$, se debe cumplir que $58(9+5(n/p)) > 112 \log_2 p$ o lo que es lo mismo que para que haya una mejora se debe cumplir que $p > 2$ y n debe cumplir la desigualdad $n > ((112/58) \log_2 p - 9)p/5 = (1.94 \log_2 p - 9)p/5$

Nota: Indica explícitament les assumpcions que prengues. Es pot utilitzar qualsevol de les funcions MPI que s'indiquen en la taula adjunta. La puntuació dependrà de la correcció del programa i l'optimització de les comunicacions.

Nota: Indica explícitamente las asunciones que tomes. Se puede utilizar cualquiera de las funciones MPI que se indican en la tabla adjunta. La puntuación dependerá de la corrección del programa y la optimización de las comunicaciones.