

Laboratorio de Arquitectura e Ingeniería de
Computadores

PRÁCTICA V
COMPUTACIÓN
PARALELA
(Paso De Mensajes)

Ingeniería de Computadores

Pedro Barquín Ayuso
Miguel Ballesteros García

PRACTICA

Para esta práctica se va a realizar la implementación de un programa que utilice las funcionalidades MPI explicadas en clase para resolver un problema mediante el uso de este sistema el cual va a dividir el trabajo en varios procesos independiente y se van a comunicar mediante el paso de mensajes una vez resuelto su cometido.

El programa que se va a realizar es el cálculo de una integral la cual nos da un resultado de 2PI y por ello se van a realizar cálculos con distinto número de procesos y distintas áreas para ver como varían los resultados.

$$\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} e^{-(x^2+y^2)/2} dx dy \approx \int_{-x_N}^{+x_N} \int_{-y_M}^{+y_M} e^{-(x^2+y^2)/2} dx dy \approx \sum_{i=0}^N \sum_{j=0}^M h^2 e^{-(x^2+y^2)/2}$$

$$x_i = -x_N + ih, \quad i = 0, \dots, N.$$

$$y_j = -y_M + jh, \quad j = 0, \dots, M.$$

CONEXIÓN Y EJECUCION

Para esta práctica hemos de utilizar el súper ordenador de la universidad ya que en las maquinas normales no están instaladas las librerías necesarias para el uso de MPI y lo más sencillo será trabajar allí. Para poder trabajar con el hemos de seguir estos pasos:

- Abrir 2 terminales, en una trabajaremos con nuestro ordenador y en otra con el otro.
- Abrir conexión mediante ssh -> ssh [gic1@172.29.23.181](ssh:gic1@172.29.23.181)
- Insertar contraseña de usuario *****
- Ubicarnos en la ruta de trabajo
- Desde la otra terminal enviar los archivos al súper ordenador mediante -> scp nombre.c [gic1@172.29.23.181:/areinco/gic1/carpeta](ssh:gic1@172.29.23.181:/areinco/gic1/carpeta)
- Insertamos contraseña *****
- Serán necesarios los documentos .c y además un documento mpd.host con esta información (172.29.23.181:23)
- Una vez completados los pasos desde el súper ordenador compilamos el código -> mpicc -o nombre nombre.c -lm
- Ejecutamos el mpdboot -> mpdboot -n 1 -f mpd.host -r ssh
- Verificamos la conexión -> mpdtrace
- Ejecutamos -> mpiexec -np número de procesos ./nombre
- Antes de salir terminamos el mpi -> mpdallexit y exit
- **Para cambiar los intervalos de integración ha de realizarse desde el código**

CODIGO

```
#include <mpi.h>
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <time.h>

#define BUFSIZE 128           //Información y parámetros para las entradas y las
                              //llamadas MPI
#define TAG 0
#define INTERVALO 5.0        //Valor del intervalo
#define ValorPi 3.14159265358979323846 //Valor real de pi hasta el decimal 20

/*Realización de la integra para cada uno de los procesos en el rango establecido*/
double Trap(int mi_rango, int local_n, double h, int n) {
    double integral=0.0;    // Para almacenar el resultado
    double exp1=0.0;
    double return_val=0.0;
    double x = 0.0;        //Variables para los cálculos
    double y = 0.0;
    int i;                  //variable del bucle
    int j;

    for(i=local_n*mi_rango;i<local_n*(mi_rango+1);i++){
        x=(-INTERVALO+(i*h));
        for(j=0;j<n;j++){
            y=(-INTERVALO+(j*h));
            exp1 = -((x*x+y*y)/2);
            return_val = (h*h)*pow(M_E,exp1);
            integral = integral + return_val;
        }
    }
    return integral;
}

//función para calcular los parámetros para cada uno de los procesos
double calcula(int mi_rango, int numprocs) {

    double b= INTERVALO; /* Extremos de la integración */
    int n = 100; /* Numero de divisiones */
    double h; /* Base de cada trapecio */
    int local_n; /* Numero de trapecios para mi calculo */
    double integral; /* Resultado de la integral en mi intervalo */

    /* Obtener siguiente numero a comprobar */
    h = (2*b)/n; /* h es el mismo para todos los procesos, es el ancho de cada
    división*/

    local_n = n/numprocs; /* numero de intervalos para cada uno */

    /* Calculo la integral pasando el número del que lo realiza el número de
    intervalos anchura y numero de divisiones para el alto*/
    integral = Trap(mi_rango, local_n, h, n);

    /*Impresión de la información del calculo de cada uno de los procesos y su
    resultado*/
    printf("Integral del %d es = %f\n", mi_rango, integral);

    return integral;
}

/*Programa principal*/
int main(int argc, char *argv[]) {
```

```

int mi_rango = 0;
int numprocs = 0;
double total_parcial = 0.0;
double total = 0.0;
double tmpinic = 0.0;
double tmpfin = 0.0;

/*Obtención de la información de cada uno*/
MPI_Init(&argc, &argv); /* todos los programas MPI comienzan con MPI_Init; los
'N' procesos comienzan tras él */
MPI_Comm_size(MPI_COMM_WORLD, &numprocs); /* averigua como de grande es la
palabra SPMD */
MPI_Comm_rank(MPI_COMM_WORLD, &mi_rango); /* y los rangos del proceso */

/* indicamos si es la maquina cero y cogemos tiempo actual */
if (mi_rango==0) {
    fprintf(stdout,"Procesando...\n");
    tmpinic=MPI_Wtime();    //Inicializamos el contador global con 0 que es el
primero en empezar
}

//Calculamos el valor de la integra para cada uno
total_parcial = calcula(mi_rango,numprocs);

//Usamos el MPI_Reduce para recolectar la información cuando hemos terminado de
operar
MPI_Reduce(&total_parcial, &total, 1, MPI_DOUBLE, MPI_SUM, TAG, MPI_COMM_WORLD);

//Si soy el 0 Imprimo el valor de la cuenta total
if (mi_rango == 0){
    //Impresión de la información del calculo y la diferencia
    printf("Suma TOTAL es: %.20lf\n", total);
    printf("Error cometido en la resolución numérica: %.20lf\n", total-
(2*ValorPi));
    //Calculamos el tiempo final y lo imprimimos
    tmpfin=MPI_Wtime();
    printf("%.48lf Tiempo total en segundos para realizar el calculo.\n", tmpfin-
tmpinic);
}
printf("\n");
/*Finalizamos todos los MPI para evitar errores*/
MPI_Finalize();
return 0;
}

```

CAPTURAS

Nº Procesos 2 Intervalos 0.5-1-5

```
lgic1@perico5 Practica5]$mpirun -np 2 ./fuente
Procesando...
Integral del 0 es = 1.457384
Suma TOTAL es: 2.92823400981253456621
Error cometido en la resolucion numerica: -3.35495129736705166579
0.0012030601501464843750000000000000000000000000000 Tiempo total en segundos para realizar el calculo.
Integral del 1 es = 1.470850
```

```
lgic1@perico5 Practica5]$mplexec -np 2 ./fuente
Procesando...
Integral del 0 es = 2.820663
Suma TOTAL es: 5.72407467108253165122
Error cometido en la resolucion numerica: -0.55911063609705458077
0.0011849403381347656250000000000000000000000000000 Tiempo total en segundos para realizar el calculo.
Integral del 1 es = 2.903412
```

```
[gic1@perico5 Practica5]$mpirun -np 2 ./fuente
Procesando...
Integral del 0 es = 3.016258
Integral del 1 es = 3.266920

Suma TOTAL es: 6.28317794772152460325
Error cometido en la resoluci3n numerica: -0.00000735945806162874
0.001068830490112304687500000000000000000000000000000 Tienpo total en segundos para realizar el calculo.
```

Nº Procesos 4 Intervalos 0.5-1-5

```
[gic1@perico5 Practica5]$mpirun -np 4 ./fuente
Procesando...
Integral del 0 es = 0.792599
Integral del 1 es = 2.028064

Integral del 2 es = 2.065719

Suma TOTAL es: 5.72407467108254053301
Error cometido en la resolucion numerica: -0.55911063609704569899
0.0011010169982910156250000000000000000000000000000 Tiempo total en segundos para realizar el calculo.

Integral del 3 es = 0.837693
```

```
[gic1@perico5 Practica5]$mpilexec -np 4 ./fuente
Procesando...
Integral del 1 es = 0.819216
Integral del 0 es = 0.638168
Integral del 2 es = 0.823238
Suma TOTAL es: 2.92823400981254344799
Error cometido en la resolucion numerica: -3.35495129736704278400
0.000941038131713867187500000000000000000000000000000 Tiempo total en segundos para realizar el calculo.
Integral del 3 es = 0.647612
```

```
[gic1@perico5 Practica5]$ mplexec -np 4 ./fuente
Procesando...
Integral del 0 es = 0.033738
Integral del 1 es = 2.982520

Integral del 3 es = 0.044750
Suma TOTAL es: 6.28317794772151660965
Error cometido en la resolucion numerica: -0.00000735945806962235
0.0010831356048583984375000000000000000000000000000 Tiempo total en segundos para realizar el calculo.

Integral del 2 es = 3.222170
```

Nº Procesos 8 Intervalos 0.5-1-5

```
[gic1@perico5 Practica5]$mpixexec -np 8 ./fuente
Procesando...
Integral del 0 es = 0.276246
Integral del 1 es = 0.332026

Integral del 2 es = 0.376835

Integral del 3 es = 0.403863

Integral del 4 es = 0.408715

Integral del 5 es = 0.390581

Integral del 6 es = 0.352456
Suma TOTAL es: 2.84105601123522699680
Error cometido en la resolucion numerica: -3.44212929594435923519
0.0013508796691894531250000000000000000000000000000 Tiempo total en segundos para realizar el calculo.
```

```
Integral del 7 es = 0.300333
```

```
[gic1@perico5 Practicas]$mpixec -np 8 ./fuente
Procesando...
Integral del 0 es = 0.240382
Integral del 1 es = 0.496493

Integral del 2 es = 0.817935

Integral del 3 es = 1.074835

Integral del 4 es = 1.126662

Suma TOTAL es: 5.66094951663957512977
Error cometido en la resolucio numerica: -0.62223579054001110222
0.0009508132934570312500000000000000000000000000000 Tiempo total en segundos para realizar el calculo.

Integral del 6 es = 0.628321

Integral del 5 es = 0.942055

Integral del 7 es = 0.334267
```

```
[gic1@perico5 Practica5]$mpicc -o fuente fuente.c -lm
[gic1@perico5 Practica5]$mpiexec -np 8 ./fuente
Procesando...
Integral del 0 es = 0.000367
Integral del 1 es = 0.024836

Integral del 4 es = 2.687889
Integral del 3 es = 2.059679
Integral del 2 es = 0.436264

Integral del 5 es = 0.975233

Suma TOTAL es: 6.28316355236362333869
Error cometido en la resolucio numerica: -0.00002175481596289330
0.0009989738464355468750000000000000000000000000000 Tiempo total en segundos para realizar el calculo.

Integral del 6 es = 0.096385

Integral del 7 es = 0.002509
```

CONCLUSIÓN Y RESULTADOS

Procesos\Intervalos	Tiempo	Aproximación
2\0,5	0,00120306	-3,354951297
2\1	0,00118494	-0,559110636
2\5	0,00106883	-7,35946E-06
4\0,5	0,000941038	-3,354951297
4\1	0,001101017	-0,559110636
4\5	0,001083136	-7,35946E-06
8\0,5	0,00135088	-3,442129296
8\1	0,000950813	-0,622235791
8\5	0,000998974	-2,17548E-05

Como podemos ver en la tabla de resultados los tiempos van subiendo progresivamente según vamos aumentando el número de procesos e intervalos, pues al aumentar el intervalo se van realizando más operaciones.

Si nos fijamos en los errores de aproximación vemos como con 0,5 y 1 tenemos bastante error porque al ser el intervalo tan estrecho no puede calcular el resultado con exactitud. Con 5, al ser el intervalo mucho más ancho que los anteriores, vemos como el error es mucho más pequeño y el resultado obtenido es el esperado.

OTROS

Para el paso de mensajes intentamos realizarlo mediante MPI_Recv y MPI_Send para así evitar que se mostrara el resultado final antes de terminar de mostrar el resultado parcial de cada uno de los cálculos pero no conseguimos resolverlo por lo que se mantuvo el uso de MPI_Reduce ya que es más simple y fue la primera opción que usamos.