Práctica 4 - Ejercicios con MPI

Moreno González Gabriela.

3CV9 - Analysis and Design of Parallel Algorithms.

December 1, 2017

Ejercicios con MPI

Integral

El código para calcular una integral en MPI es el siguiente:

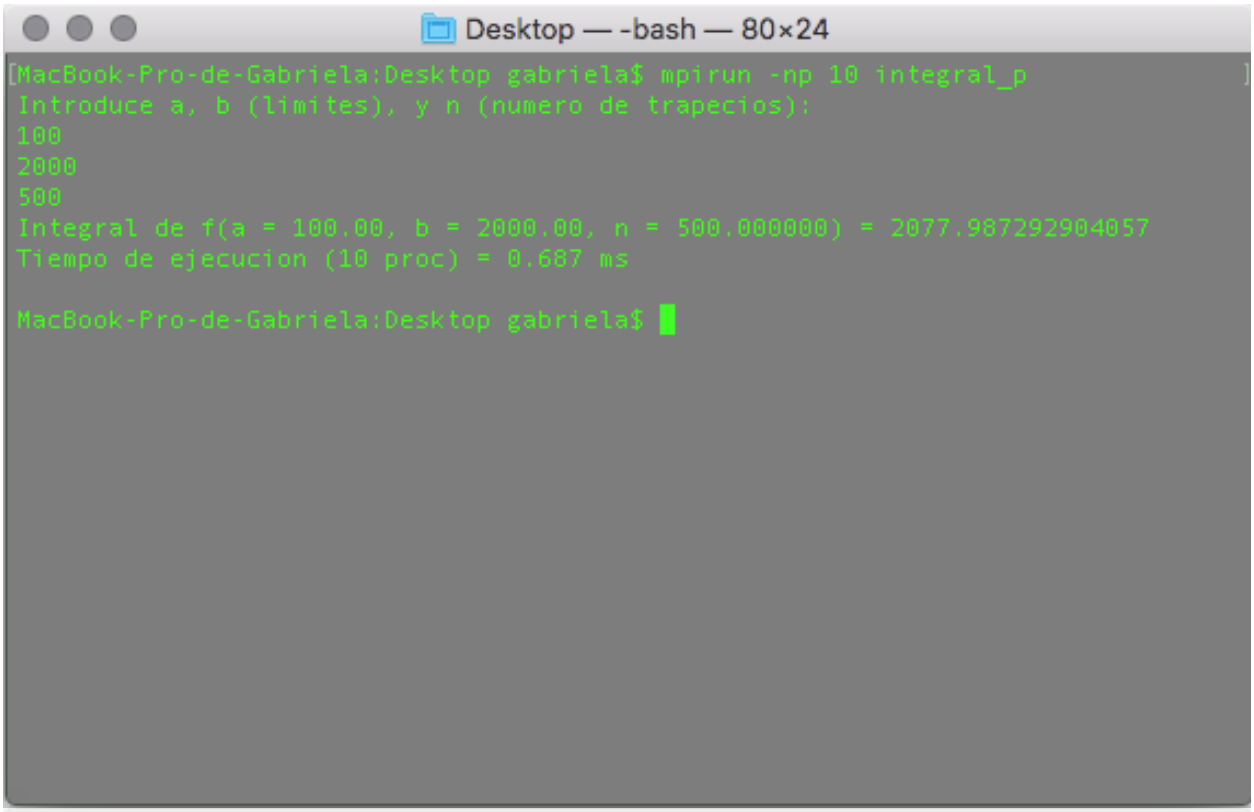
integral_p.cpp

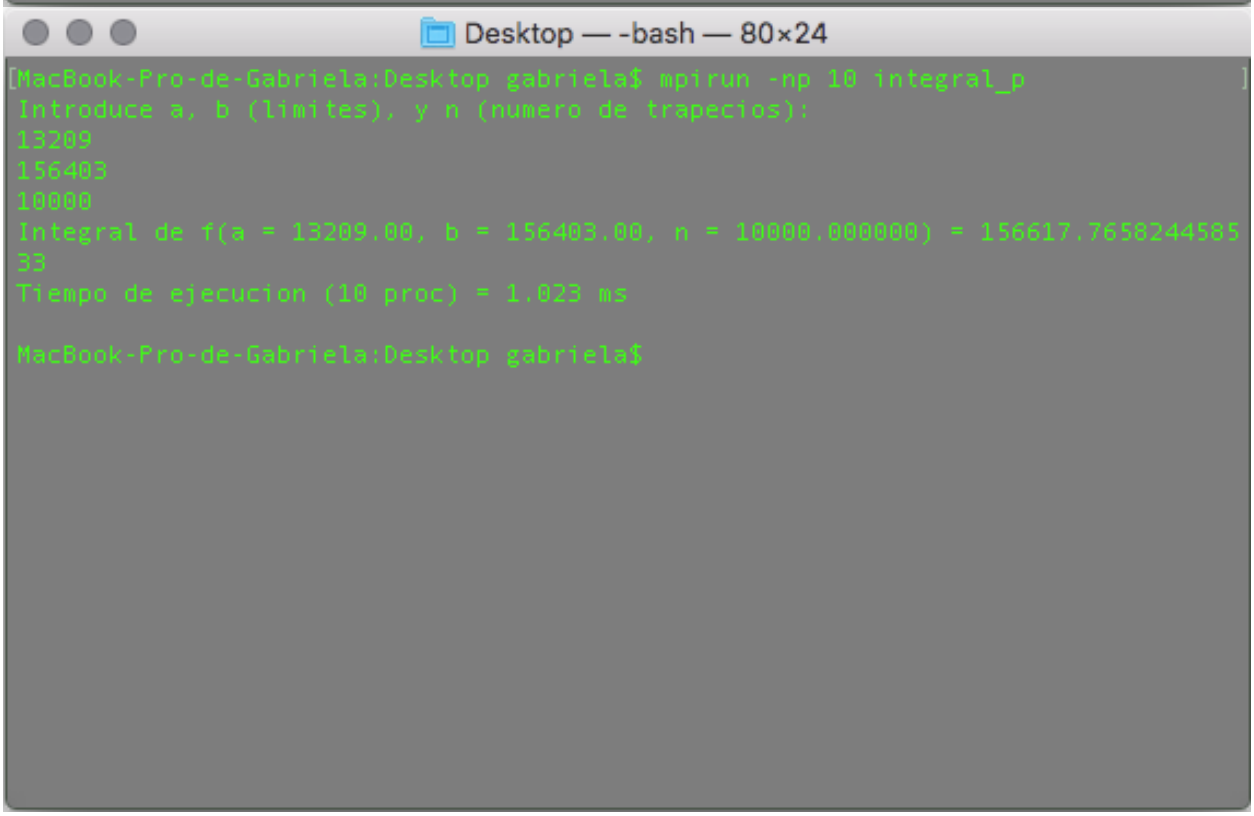
```
#include <stdio.h>
#include <math.h>
#include <mpi.h>

double t0, t1;

double f(double x)
{
    double y;

    y = 1.0 / (sin(x) + 2.0) + 1.0 / (sin(x) * cos(x) + 2.0);

    return y;
}

void leerDatos(double* a_ptr, double* b_ptr, long long* n_ptr, int pid)
{
    float aux_a, aux_b, aux_n;
    int root;

    if(pid == 0)
    {
        printf("Introduce a, b (limites), y n (numero de trapecios):\n");
        scanf("%f %f %f", &aux_a, &aux_b, &aux_n);
        t0 = MPI_Wtime();
    }

    (*a_ptr) = (double)aux_a;
    (*b_ptr) = (double)aux_b;
    (*n_ptr) = (long long)aux_n;

    root = 0;

    MPI_Bcast(a_ptr,1,MPI_DOUBLE,root,MPI_COMM_WORLD);
```

Así, ejecutando el programa de la integral nos arroja lo siguiente:

```
35      MPI_Bcast(b_ptr,1,MPI_DOUBLE,root,MPI_COMM_WORLD);
36      MPI_Bcast(n_ptr,1,MPI_LONG_LONG,root,MPI_COMM_WORLD);
37  }
38
39  double Integrar(double a_loc, double b_loc, long long n_loc, double w)
40  {
41      double resul_loc,x;
42      long long i;
43
44      resul_loc = (f(a_loc) + f(b_loc))/2.0;
45      x = a_loc;
46
47      for(i = 1; i<n_loc;i++)
48      {
49          x = x+ w;
50          resul_loc = resul_loc + f(x);
51      }
52      resul_loc = resul_loc * w;
53
54      return (resul_loc);
55  }
56
57  int main(int argc, char **argv)
58  {
59      int pid,npr,root;
60      double a,b,w,a_loc,b_loc;
61      long long n,n_loc;
62      double resul_loc, resul;
63
64      MPI_Init(&argc,&argv);
65      MPI_Comm_rank(MPI_COMM_WORLD,&pid);
66      MPI_Comm_size(MPI_COMM_WORLD,&npr);
67
68      leerDatos(&a,&b,&n,pid);
```

```
69
70      w = (b-a)/n;
71
72      n_loc = (pid+1)*n/npr - pid*n/npr;
73      a_loc = a + (pid)*n/npr * w;
74      b_loc = a + (pid+1)*n/npr * w;
75
76      resul_loc = Integrar(a_loc,b_loc,n_loc,w);
77      resul = 0.0;
78      root = 0;
79
80      MPI_Reduce(&resul_loc,&resul,1,MPI_DOUBLE,MPI_SUM,root,MPI_COMM_WORLD);
81
82      if(pid == 0)
83      {
84          t1 = MPI_Wtime();
85          printf("Integral de f(a = %1.2f, b = %1.2f, n = %f) = %1.12f\n",a,b,(double)n,resul);
86          printf("Tiempo de ejecucion (%d proc) = %1.3f ms\n\n",npr,(t1-t0)*1000);
87      }
88
89      MPI_Finalize();
90      return 0;
91  }
92
93
94
Line 64, Column 13
```


Desktop — -bash — 80×24
[MacBook-Pro-de-Gabriela:Desktop gabriela$ mpirun -np 10 integral_p
Introduce a, b (limites), y n (numero de trapecios):
100
2000
500
Integral de f(a = 100.00, b = 2000.00, n = 500.000000) = 2077.987292904057
Tiempo de ejecucion (10 proc) = 0.687 ms
MacBook-Pro-de-Gabriela:Desktop gabriela$


Desktop — -bash — 80×24
[MacBook-Pro-de-Gabriela:Desktop gabriela$ mpirun -np 10 integral_p
Introduce a, b (limites), y n (numero de trapecios):
13209
156403
10000
Integral de f(a = 13209.00, b = 156403.00, n = 10000.000000) = 156617.765824458533
Tiempo de ejecucion (10 proc) = 1.023 ms
MacBook-Pro-de-Gabriela:Desktop gabriela$