MPI - Programming assignments

Roque Sosa Owen Haziel

October 30, 2024

Ejercicio 3.1

Use MPI to implement the histogram program discussed in Section 2.7.1. Have process 0 read in the input data and distribute it among the processes. Also have process 0 print out the histogram.

Pseudocódigo

1. **Obtener las entradas del usuario**

   1.1 La cantidad de mediciones, *data count*.

   1.2 Un arreglo de flotantes de tamaño *data count*, *data*.

   1.3 El valor mínimo para el bin que contiene los valores más pequeños, *min meas*.

   1.4 El valor máximo para el bin que contiene los valores más grandes, *max meas*.

   1.5 La cantidad de bins, *bin count*.

2. **La salida será un arreglo que contiene el número de elementos de *data* que se encuentran en cada bin**, donde:

   - *bin maxes* es un arreglo de flotantes de tamaño *bin count*.
   - *bin counts* es un arreglo de enteros de tamaño *bin count*.

3. **El arreglo *bin maxes* almacenará el límite superior para cada bin, y *bin counts* almacenará el número de elementos de *data* en cada bin**, donde:

$$bin\ width = \frac{max\ meas - min\ meas}{bin\ count}$$

4. **Inicializar el arreglo *bin maxes***.

5. **Definir la convención de que el bin *b* incluirá todas las mediciones en el rango**:

$$bin\ maxes[b-1] \leq measurement < bin\ maxes[b]$$

6. **La función *Find bin* retorna el bin al que pertenece *data[i]***, lo que se puede lograr mediante una búsqueda lineal simple en *bin maxes* hasta encontrar un bin b que satisfaga:

$$bin\ maxes[b-1] \leq data[i] < bin\ maxes[b]$$

7. **Los elementos de *data* se asignan a los procesos/hilos para que cada proceso/hilo obtenga aproximadamente el mismo número de elementos.**

8. **Cada proceso/hilo es responsable de actualizar su arreglo *loc bin cts* en función de los elementos asignados.**

Resultado

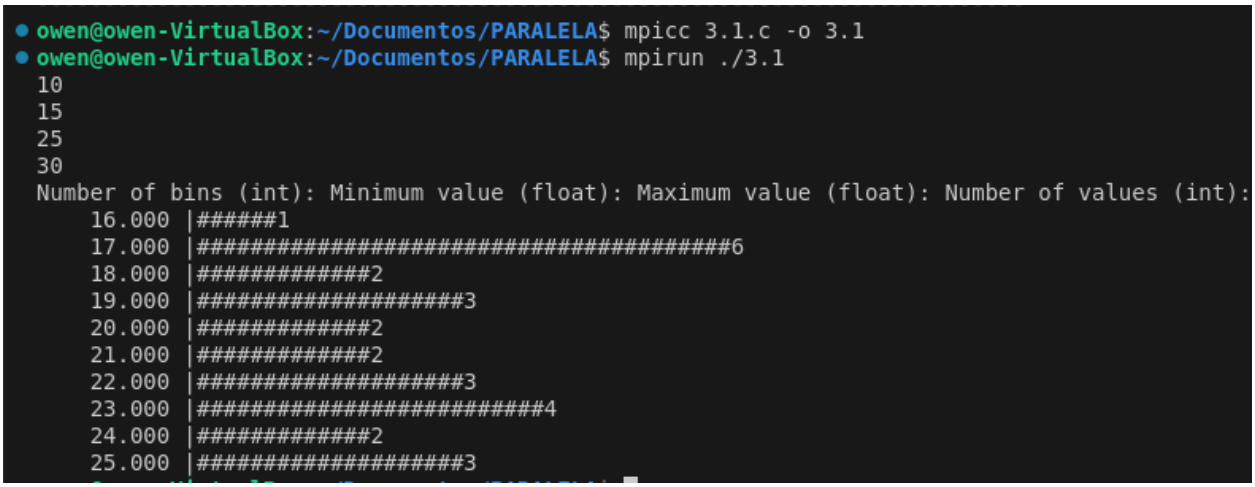

Figure 1: Resultado del ejercicio 3.1

Ejercicio 3.2

Write an MPI program that uses a Monte Carlo method to estimate π. Process 0 should read in the total number of tosses and broadcast it to the other processes. Use *MPI_Reduce* to find the global sum of the local variable *number_in_circle*, and have process 0 print the result. You may want to use long long ints for the number of hits in the circle and the number of tosses, since both may have to be very large to get a reasonable estimate of π.

Pseudocódigo

Resultado

Ejercicio 3.3

Write an MPI program that computes a tree-structured global sum. First write your program for the special case in which *comm_sz* is a power of two. Then, after you've gotten this version working, modify your program so that it can handle any *comm_sz*

Resultado

* Solamente la version con comm_sz elevado al cuadrado

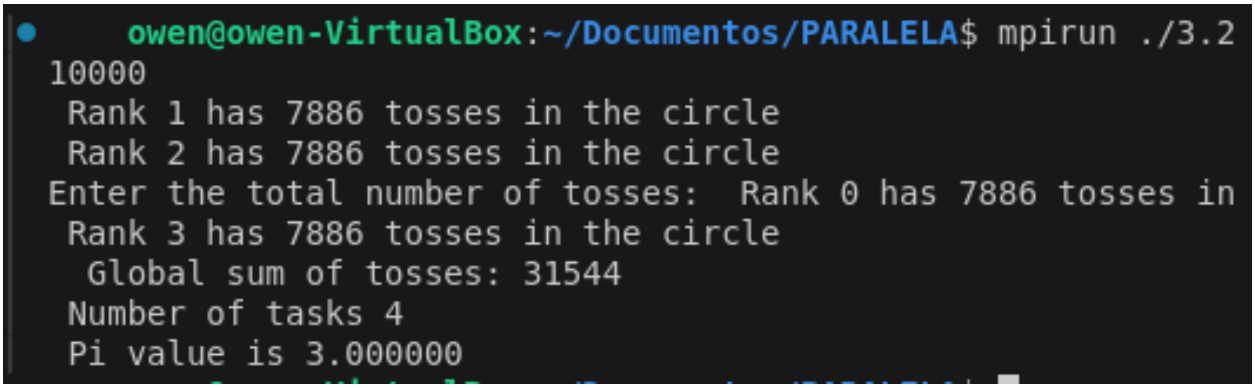

Figure 2: Resultado del ejercicio 3.2

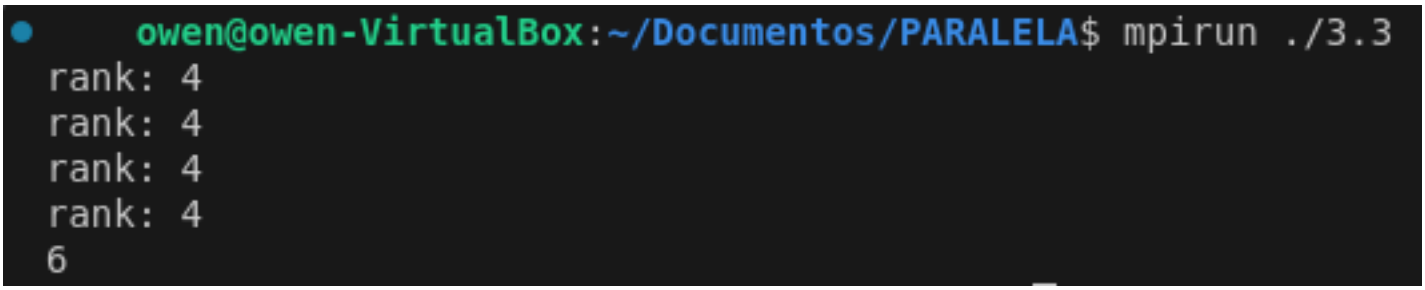

Figure 3: Resultado del ejercicio 3.3

1 Repositorio

https://github.com/OwenRoque/PARALELA