

Práctica 1.- Manejo de MPI

Arquitectura y Computación de Altas Prestaciones. Universidad de Granada. Curso 2023/24

Objetivos: En esta práctica se pretende que los alumnos se familiaricen con el diseño de soluciones MPI para la resolución de problemas y la medición de tiempos. Igualmente, tomarán contacto con el cluster de prácticas.

Bases: Esta práctica se realizará individualmente y los programas desarrollados estarán pensados para funcionar en un entorno Linux. Habrá que entregar los archivos de código fuente desarrollados y una memoria (en formato PDF) que explique el trabajo realizado. La entrega se realizará mediante la actividad que se creará en SWAD para tal efecto. Las entregas tardías supondrán una penalización en la nota (0.5 puntos por día hasta alcanzar el límite de 5).

Entorno: La mayor parte del trabajo se puede realizar en equipos personales. Sin embargo, según la actividad, habrá que acceder al cluster de prácticas ATCGrid (repasar el documento adicional *AC_Seminario0_entorno.pdf*). Cada alumno tendrá un usuario y contraseña en ATCGrid. Serán asignados por el profesor de forma individual e intransferible. Tampoco se podrán cambiar las claves recibidas. El acceso al cluster de prácticas puede requerir estar conectado por VPN a la red de la universidad.

Ejercicio 1 (1 punto)

Muestra cómo compilas el ejemplo 1 del seminario MPI en ATCGrid. Después, ejecútalo dentro de la cola de Slurm de forma asíncrona (comando *sbatch* con script inyectado *-wrap*) y dentro de la partición "acap". Juega con los parámetros *N* y *n* de forma que se ocupe primero sólo un nodo y luego dos. Pon además algún caso que Slurm vea inviable.

Ejercicio 2 (2 puntos)

El producto escalar de dos vectores, *u* y *v*, se define como la siguiente operación algebraica que devuelve un escalar: $u \cdot v = u_1 \cdot v_1 + u_2 \cdot v_2 + \dots + u_n \cdot v_n$.

Modifica el ejemplo 4 del seminario de MPI para convertirlo en un cálculo del producto escalar. Esta vez haz que todos los procesos trabajen, incluido el proceso 0 ó maestro. Explica los cambios y pon unos cuantos ejemplos de ejecución.

Ejercicio 3 (4 puntos)

Se va a crear un servicio distribuido mediante MPI. Haz por tanto un programa que sólo conciba su ejecución con 4 procesos, cada uno con una funcionalidad distinta. El proceso 0 centralizará toda la entrada/salida y la interacción con el usuario. En concreto, solicitará en

bucle infinito un comando, que será un número (en el rango [0,4]). Dependiendo del número indicado en el comando, se realizará una de estas actividades:

- Si el número es 0, se saldrá del programa correctamente, acabando todos los procesos su ejecución.
- Si el número es 1, se le pedirá al usuario la longitud de un vector, y el proceso 0 pedirá entonces al proceso 1 un vector de números decimales aleatorios entre 0 y 1 con la longitud indicada. Dicho vector será mostrado por consola.
- Si el número es 2, se le pedirá al usuario una palabra, y el proceso 0 la enviará entonces al proceso 2. Éste sumará el valor ASCII de cada carácter y se lo devolverá al proceso 0, que mostrará dicho resultado.
- Si el número es 3, se le pedirá al usuario una palabra, y el proceso 0 la enviará entonces al proceso 3. Éste convertirá la cadena de caracteres a mayúscula y se la devolverá al proceso 0, que la mostrará por consola.
- Si el número es 4, se harán todas las acciones anteriores (menos la 0).

Ejercicio 4 (3 puntos)

El número π es la relación entre la longitud de una circunferencia y su diámetro. Es un número irracional y una de las constantes más usada en Matemáticas, Física e Ingeniería. Hay distintos métodos para aproximar su valor, y uno de ellos es el siguiente:

Se puede demostrar que:

$$\int_0^1 \frac{4}{1+x^2} dx = \pi$$

A su vez, esa integral se puede aproximar como una suma del área de N rectángulos:

$$\sum_{i=0}^N F(x_i) \Delta x \cong \pi$$

Donde cada rectángulo tiene una base fija Δx y una altura dada por $F(x) = 4/(1+x^2)$. Se proporciona un programa C que implementa este cálculo. Se pide que paralelice dicho cálculo en MPI para un número cualquiera de procesos, asumiendo que sólo el proceso 0 recibe parámetros, balanceando la carga equitativamente, y minimizando la interacción entre procesos. Incluye una medición del tiempo real ("wall time") empleado donde estimes oportuno y que ésta se muestre por consola. Prueba algunos valores y compara los tiempos con respecto a la versión secuencial. ¿Crees que merece la pena la versión paralela?

Finalmente, echa un vistazo al siguiente enlace (presta atención especialmente a la respuesta dada por el usuario denominado "Hristo Iliev"):
<https://stackoverflow.com/questions/27679041/the-differences-in-the-accuracy-of-the-calculations-in-single-multi-threaded>

En general, explica cómo puede afectar la realidad que se describe en este y otros cálculos que se hagan a lo largo de la asignatura. ¿Eras consciente de este aspecto?