

Práctica 2: Computación Paralela con Planetas

Mónica Chillarón

Computación Paralela (CPA)

Curso 2020/2021



DEPARTAMENTO DE SISTEMAS
INFORMÁTICOS Y COMPUTACIÓN



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

1. Recordatorios
2. Entrega de la práctica
3. Memoria de la práctica
4. Problema a resolver
5. Trabajo a realizar

- Este documento es únicamente para apoyar la explicación. El material completo de la práctica está en la tarea de PoliformaT. El **boletín** contiene toda la información necesaria y es el documento que se tiene que seguir para realizar el trabajo. Recomendable leerlo antes de cada sesión.
- Nunca utilizar espacios en blanco en los nombres de directorios o ficheros.
- El Front-end sirve para realizar comprobaciones cortas, nunca para medir tiempos o realizar ejecuciones costosas. Los tiempos se toman lanzando los trabajos al sistema de colas mediante qsub.

Entrega de la práctica

- La fecha de entrega de la práctica 2 es: **04-dic-2020 23:55**
- Los retrasos en la entrega penalizan
- Hay dos tareas en poliformaT: Una para el PDF de la memoria, y otra para código fuente
- Se recomienda realizar la práctica por parejas
- Las parejas tienen que ser del mismo grupo de prácticas
- Sólo un miembro de la pareja hará la entrega
- Cualquier copia detectada, ya sea en código o memoria, podrá suponer una calificación de 0

Memoria de la práctica

- La parte importante de la entrega es la memoria
- La memoria tiene que plasmar todo el trabajo realizado y el aprendizaje del alumno
- La extensión no es fija, depende del criterio personal
- Evitar repeticiones o relleno, pero no tener miedo de que ocupe mucho si se cree necesario
- **La presentación es muy importante**
- Cuidar los detalles, estructura, formato, etc.
- No es simplemente exponer resultados, sino explicarlos y analizarlos
- Para comentar el código, me es indiferente que se ponga como texto o con capturas de pantalla, siempre que se entienda bien

Problema a resolver

En esta práctica se resuelve el cálculo del centro de masas de un conjunto de planetas, y se calcula cuál es el planeta con más vecinos

- Un planeta está definido por sus coordenadas, y su masa
- Dos planetas son vecinos si su distancia es menor a una distancia dada: NGB_DST
- El procesamiento se hace en la función **process_data**
- El programa acepta como argumento el número de planetas a considerar (por defecto 10000)
- Los datos de los planetas se generan aleatoriamente. Aunque no cambian para el mismo número de planetas

```
$ ./planets 5000  
The center of mass is in ( 1180.98, 1184.01, 1175.36 ).  
The planet with most neighbors is 569 (15 neighbors).
```

Función process_data

- **Primer bucle i:** recorre los planetas, acumulando la aportación de cada uno de ellos al centro de masas del conjunto y obteniendo el número de vecinos de cada planeta
- **Bucle j:** cada iteración calcula la distancia entre los planetas i y j y, si los planetas se consideran vecinos, incrementa el número de vecinos tanto del planeta i como del j
- **Segundo bucle i:** se obtiene el planeta con mayor número de vecinos. Si hay varios planetas con el máximo número de vecinos, el programa se queda con el de índice menor

```
int process_data( int n, Planet p[], Point3D *pcm )
{ int i, j, ngbi, k, max;
  double dx, dy, dz, d, m;
  Point3D cm; // center of mass

  cm.x = cm.y = cm.z = 0; m = 0;
  for (i=0; i<n; i++) {
    // Accumulate mass and weighted position to compute the center of mass
    m += p[i].m;
    cm.x += p[i].m*p[i].p.x;
    cm.y += p[i].m*p[i].p.y;
    cm.z += p[i].m*p[i].p.z;

    // Update count of neighboring planets
    ngbi = 0;
    for (j=i+1; j<n; j++) {
      // Compute distance between planet i and planet j
      dx = p[i].p.x - p[j].p.x;
      dy = p[i].p.y - p[j].p.y;
      dz = p[i].p.z - p[j].p.z;
      d = sqrt(dx*dx + dy*dy + dz*dz);
      if ( d < NGB_DST ) {
        // Planets i and j are neighbors. Update number of neighbors
        // for both planets
        ngbi++;
        p[j].ngb++;
      }
    }
    p[i].ngb += ngbi;
  }

  // Finish computation of the center of mass
  if ( m > EPS ) { cm.x /= m; cm.y /= m; cm.z /= m; }

  // Find the planet with most neighbors
  k = 0; max = -1;
  for (i=0; i<n; i++) {
    if ( p[i].ngb > max ) {
      max = p[i].ngb; k = i;
    }
  }

  *pcm = cm;
  return k;
}
```

- **Ejercicio 1:** Crea una copia del programa, llamada **planets1.c**, en la que añadas la toma de tiempo y muestres el número de planetas (n) considerado. Esta versión será la de referencia para el tiempo secuencial y la base para las demás.
- **Ejercicio 2**
 - Ejercicio 2.1: Crea la version **planets2i.c** en la que paralelices tanto el primer como el segundo bucle i .
 - Ejercicio 2.2: Crea la version **planets2j.c** en la que paralelices el bucle j junto con el segundo bucle i .
 - Si crees que algún bucle no se puede paralelizar, explica por qué
 - En ambas versiones se debe mostrar tiempo, número de hilos y número de planetas
 - Presta especial atención a la actualización del número de vecinos. Piensa si dos iteraciones de ese bucle pueden estar modificando el número de vecinos de un mismo planeta y si **genera concidiones de carrera**.

Trabajo a realizar

- **Ejercicio 3:** Para cada bucle paralelizado, ¿crees que la planificación utilizada afectará a las prestaciones obtenidas? Razona la respuesta.
- **Ejercicio 4**
 - Utiliza un número de planetas que consiga que el tiempo secuencial esté entre 30 y 90 segundos.
 - Toma de tiempos en los nodos de cálculo de Kahan
 - Prueba distintas planificaciones si crees que la planificación puede afectar
 - Utiliza potencias de 2 para el número de hilos (hasta el número que creas necesario, justificándolo)
 - Compara las prestaciones mediante tablas/gráficas del SpeedUp y Eficiencia obtenido con las distintas planificaciones. Realiza un análisis de los resultados
 - Compara las prestaciones mediante tablas/gráficas del SpeedUp y Eficiencia obtenido con las dos versiones paralelas (planets2i.c y planets2j.c). Realiza un análisis de los resultados que explique cuál es la mejor versión paralela
 - Incluye la explicación de cómo has lanzado los trabajos para la toma de tiempos

- **Ejercicio 5:** Haz una versión (**planets5.c**) basada en **planets2i.c**, en la que se añada/modifique lo necesario para que cada hilo muestre por pantalla cuántos planetas le ha tocado procesar (número de iteraciones del primer bucle *i* que ha realizado) y el centro de masas de estos planetas. Por lo demás, el programa debe seguir funcionando como antes. No se pide analizar las prestaciones de esta versión.

```
El hilo 1 ha procesado 10011 planetas. Centro de masas: (2380.53, 2369.02, 2392.19 ).
El hilo 3 ha procesado 9987 planetas. Centro de masas: (2384, 2392.55, 2378.22 ).
El hilo 0 ha procesado 9987 planetas. Centro de masas: (2370.97, 2395.42, 2375.13 ).
El hilo 2 ha procesado 10015 planetas. Centro de masas: (2408.05, 2366.17, 2381.7 ).
The center of mass is in ( 2385.9, 2380.7, 2381.86 ).
The planet with most neighbors is 14325 (17 neighbors).
Tiempo (40000 planetas, 4 hilos): 5.805158 segundos.
```