

UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN
MATERIA: FUNDAMENTOS DE PROGRAMACIÓN PARALELA

TAREA FINAL

EXPLICACIÓN

El problema de los N cuerpos (no confundir con el problema de los 3 cuerpos) es una situación muy común en muchas áreas de la física y la química. Consiste en que, dadas unas condiciones iniciales de posición y velocidad de partículas, se calcula los cambios de movimientos de acuerdo a las atracciones mutuas de los cuerpos.

Se tiene el código que se usó de prueba para el curso de CUDA, donde un determinado número de partículas se mueven en un espacio 3D a diferentes velocidades. Este problema es N^3 en condiciones normales cuando se tiene una secuencia discreta de medidas de tiempo Δt

El problema a resolver es determinar el estado final de una simulación simple para un número alto de partículas, 65536 (2^{15}) usando MPI y OpenMP

Tenga en cuenta las condiciones de borde, ya que las posiciones de cada partícula cambian en cada Δt .

ENTRADA

Hay dos archivos de entrada, uno con 4096 partículas y otra con 65536, en formato binario y CSV

SALIDA

Un archivo de salida con la posición final de las partículas

CONDICIONES

Debe entregar un análisis de la solución diseñada y pruebas de ejecución. Dado que no hay un cluster MPI operativo, no se exigirán pruebas de rendimiento ni medidas de eficiencia. El problema puede ser resuelto en parejas

FECHA

Todo debe estar entregado para el día **VIERNES 21 DE MARZO** a las 12:00 del mediodía. Se ejecutarán los programas en el laboratorio del CCPD y cerraremos el semestre como es debido

ANEXO

Explicación de Copilot para el código nBodies:

```
for (int i = 0; i < n; ++i) {  
    float Fx = 0.0f; float Fy = 0.0f; float Fz = 0.0f;  
  
    for (int j = 0; j < n; j++) {  
        float dx = p[j].x - p[i].x;  
        float dy = p[j].y - p[i].y;  
        float dz = p[j].z - p[i].z;  
        float distSqr = dx*dx + dy*dy + dz*dz + SOFTENING;  
        float invDist = rsqrtf(distSqr);  
        float invDist3 = invDist * invDist * invDist;  
  
        Fx += dx * invDist3; Fy += dy * invDist3; Fz += dz * invDist3;  
    }  
  
    p[i].vx += dt*Fx; p[i].vy += dt*Fy; p[i].vz += dt*Fz;  
}  
}
```

Este código define una función llamada `bodyForce` que calcula las fuerzas entre cuerpos y actualiza sus velocidades en función de esas fuerzas. Aquí está el desglose de lo que hace:

1. Inicialización de fuerzas: Para cada cuerpo i , se inicializan las fuerzas F_x , F_y y F_z a cero.
2. Cálculo de fuerzas: Para cada par de cuerpos i y j , se calculan las diferencias en las posiciones dx , dy y dz . Luego, se calcula la distancia al cuadrado $distSqr$ y su inverso $invDist$ utilizando la función `rsqrtf` (que calcula la raíz cuadrada inversa). $invDist3$ es el inverso de la distancia al cubo.
3. Acumulación de fuerzas: Las fuerzas en las direcciones x , y y z se acumulan en F_x , F_y y F_z respectivamente, multiplicando las diferencias de posición por $invDist3$.
4. Actualización de velocidades: Finalmente, las velocidades vx , vy y vz del cuerpo i se actualizan sumando las fuerzas multiplicadas por el intervalo de tiempo dt .

Este código es útil para simulaciones de cuerpos en movimiento bajo la influencia de fuerzas, como en simulaciones de gravedad o interacciones entre partículas.

¿No se usa la masa de los cuerpos en estas fórmulas? ¿cual es el sustento físico?

Correcto, en el código que has compartido, la masa de los cuerpos no se utiliza explícitamente. Esto simplifica el cálculo de las fuerzas, pero en una simulación física más realista, la masa sería un factor importante. Aquí está el sustento físico detrás de las fórmulas utilizadas:

Sustento Físico

1. Ley de la Gravitación Universal: La fuerza gravitacional entre dos cuerpos se calcula como:

$$F = \frac{G \cdot m_1 \cdot m_2}{r^2}$$

donde (G) es la constante de gravitación universal, (m1) y (m2) son las masas de los cuerpos, y (r) es la distancia entre ellos.

2. Simplificación en el Código: En el código, la fuerza se calcula sin considerar las masas, lo que implica que se está utilizando una forma simplificada de la interacción gravitacional. La fórmula utilizada es:

$$F = \frac{1}{r^3}$$

Esto se puede interpretar como una fuerza proporcional a la inversa del cubo de la distancia, lo cual es una aproximación que puede ser útil en ciertas simulaciones donde las masas son iguales o se normalizan.

3. Actualización de Velocidades: Las velocidades se actualizan en función de las fuerzas calculadas y el intervalo de tiempo:

$$dt: v_x + = dt \cdot F_x \quad v_y + = dt \cdot F_y \quad v_z + = dt \cdot F_z$$

Consideraciones

Para una simulación más precisa, deberías incluir la masa de los cuerpos en los cálculos de fuerza. Esto se puede hacer modificando el código para incluir la masa en la estructura Body y ajustando las fórmulas de fuerza en consecuencia.