



DEEC
DEPARTAMENTO DE ENGENHARIA
ELECTROTÉCNICA E DE COMPUTADORES

ARQUITETURA DE COMPUTADORES
2017/2018

Simulação de Gravitação com OpenMP
Exercício 15

AUTORIA:

Rui Barbosa
2014225254
uc2014225254@student.uc.pt

Rui Costa
2013156815
uc2013156815@student.uc.pt

Conteúdo

1	Introdução	2
2	Operações a ficheiros	2
3	Cálculos vetoriais	2
4	Aplicação global do algoritmo	2
5	Aplicação de OpenMP	3
6	Resultados obtidos	3

1 Introdução

Este relatório tem como objectivo descrever a elaboração do exercício 15 do Suporte às Aulas Laboratoriais.

Esta aula laboratorial é centrada à volta do OpenMP, que é uma biblioteca que facilita a programação multi-processador, através da utilização de algumas primitivas prédefinidas. Com este relatório queremos também documentar o ganho obtido com a utilização do OpenMP na simulação de uma interação gravitacional entre corpos.

2 Operações a ficheiros

Para iniciar a simulação foi necessário fazer uma leitura do ficheiro *particulas.dat* de forma a obter a posição inicial e massa de cada um dos corpos, tal como o tempo total da simulação e o intervalo de tempo entre cada calculo.

Para isso foi necessário primeiro fazer uma leitura do ficheiro, apenas para contar o numero de linhas, de tal forma a saber o numero de corpos que iriam existir na simulação. De seguida é feita uma leitura, linha a linha, aplicando um algoritmo de divisão de uma string por um divisor comum, neste caso o carácter de espaço. Para cada linha é corrido um ciclo que separa a *string* em duas strings resultantes, dividida pelo primeiro espaço encontrado, sendo que o resultado desta divisão são sempre duas *strings*, a primeira sendo o valor desejado, e a segunda a restante string. Assim, após termos os valores desejados, guardados numa estrutura do tipo *CORPO*, podemos aplicar o algoritmo da simulação.

Após o cálculo de cada posição é necessário também escrever essa nova posição para um novo ficheiro, localizado em *build/*, também chamado *partiles.bat*. Para isso foi criada também uma função para escrita de uma linha no ficheiro, com o formato de cada linha equivalente a "*x y z*" para cada corpo, e uma linha em branco no final de cada iteração temporal.

3 Cálculos vetoriais

De forma a facilitar a manipulação dos dados dos corpos, foram criadas as estruturas de dados *CORPO* e *VECTOR*, tal como as funções *get_vector_cartesian*, *get_vector_polar*, que devolvem um objecto do tipo *VECTOR*, com os parametros de entrada equivalentes às suas coordenadas, no formato cartesiano ou polar, respectivamente.

Foram também criadas funções para o calculo do versor de um vector, soma entre vetores, subtração de vetores, multiplicação de vetores e divisão de vectores por um valor escalar. As mesmas foram utilizadas para facilitar o calculo das equações que vão reger o "Universo" deliniado no ficheiro fornecido como suporte ao trabalho prático.

4 Aplicação global do algoritmo

O calculo do somatório de forças aplicado a cada corpo foi implementado através de um ciclo *for* visto que, para cada corpo, é necessário utilizar informação de todos os outros corpos.

$$\mathbf{f}_{i,j} = \frac{G.m_i.m_j}{||\mathbf{p}_j - \mathbf{p}_i||^2} \frac{\mathbf{p}_j - \mathbf{p}_i}{||\mathbf{p}_j - \mathbf{p}_i||}$$

Figura 1: Equação da força gravitica aplicada entre dois corpos.

$$\mathbf{f}_i = \sum_{j=0, j \neq i}^N \mathbf{f}_{i,j}.$$

Figura 2: Equação da força resultante da interação gravitacional entre todos os corpos.

$$a_x = f_x/m,$$

$$v_x(k) = v_x(k-1) + a_x \times \Delta t$$

$$p_x(k) = p_x(k-1) + v_x \times \Delta t$$

Figura 3: Equações para calculo da posição em x (aplicável de igual forma às restantes coordenadas)

As equações foram consideradas pela seguinte ordem : força gravitica entre dois corpos; somatorio das forças (depois de ter calculado todas as forças exercidas entre um corpo fixo e todos os outros corpos); aceleração (onde o uso do somatório era imprescindível); velocidade e posição.

Todas elas exigiam este encadeamento, já que elas recorrem a valores que foram calculados com o uso da equação anterior (da ordem apresentada). Durante todo o processo foi utilizado o multi-processamento, como foi referido anteriormente, para ser possível calcular as varias iterações temporais.

5 Aplicação de OpenMP

Inicialmente tinhamos como objectivo aplicar o paralelismo tanto com o *omp for*, como com *omp sections*, o que iria implicar utilizar o *for* no ciclo interior (de calculo do sumatório das forças), e o calculo de velocidade, aceleração e posição como *sections*, mas após comparação de resultados, concluímos que o uso de um *omp for* no ciclo exterior (que corre os corpos) dava resultados melhores.

6 Resultados obtidos

Inicialmente tentamos testar o algoritmo com 1001 corpos e 5 milhões de iterações temporais, porém o resultado era sempre *SIGKILL* a meio da execução do programa. A contagem do tempo real foi feita utilizando o comando *time* na execução do programa.

Resultados com a utilização de 1 processador:

525 iterações temporais - 1000 corpos - 795.147 segundos.

5256 iterações temporais - 100 corpos - 85.162 segundos.

52 iterações temporais - 1000 corpos - 80.198 segundos.

Resultados com a utilização de 2 processadores:

525 iterações temporais - 1000 corpos - 786.256 segundos*processadores - 538.94 segundos (real).

5256 iterações temporais - 100 corpos - 90.149 segundos*processadores = 88.81 segundos (real)

52 iterações temporais - 1000 corpos - 84.661 segundos*processadores = 50.29 segundos (real)

Resultados com a utilização de 4 processadores:

525 iterações temporais - 1000 corpos - 1023.637 segundos*processadores - 396.22 segundos (real).

5256 iterações temporais - 100 corpos - 104.734 segundos*processadores = 90.06 segundos (real)

52 iterações temporais - 1000 corpos - 103.089 segundos*processadores = 36.96 segundos (real)