Mestrado em Engenharia Eletrotécnica e de Computadores

Computação Paralela e Distribuída

Projeto

Simulação de Partículas

Ion Ciobotari 84075

Pedro Fernandes 84168

Ivan Andrushka 86291

Parte II

17/05/2019

TÉCNICO LISBOA

Abordagem usada para a paralelização

O principal método de paralelização usado no projeto consiste na separação de centros de massa por todos os processadores através do uso da biblioteca OpenMPI. Em problemas com elevado número de partículas as mesmas vão estar mais uniformemente distribuídas que em problemas com menor número de partículas. Dado que nesta parte do projeto o objetivo do mesmo consiste no uso de um *cluster*, é evidente que se dará destaque a problemas com complexidade elevada, nomeadamente ao nível do número de partículas. Ao dividir os centros de massa é mais fácil calcular o valor dos mesmo e posteriormente das forças resultantes, velocidades e posições novas, do que por exemplo dividir as partículas pelos processadores, pelo que foi este o método usado.

Decomposição usada na paralelização

Apesar de os centros de massa estarem alocados num vetor, os mesmos podem ser agrupados como uma matriz. Apenas se usou um vetor para diminuir o tempo de acesso ao mesmo. Assim foi decidido usar a decomposição dos mesmos em formato xadrez (*checkerboard decomposition*), como observado na figura 1. Foi também alocado um anel em volta da matriz que recebe os centros de massa dos processos vizinho, como evidenciado na figura 2. Isto torna o problema mais escalável para elevados valores de processadores do que a decomposição em linhas ou colunas, por exemplo.

O algoritmo aplicado divide a matriz de centro de massas num número de blocos igual ao número de processadores solicitados para execução. Todos estes blocos têm as mesmas dimensões, exceto aqueles que contêm informação da coluna da direita e da linha de baixo da matriz original, que podem conter dimensões maiores. Esta desigualdade na divisão de memória deve-se ao facto do tamanho da matriz não ser possível dividir uniformemente pelos processadores.

No entanto nem sempre é possível dividir os centros de massa equitativamente pelos processadores. Por exemplo quando são usados um número primo de processadores, onde o algoritmo degenera numa decomposição de linhas, ou quando o número de processadores é superior ao número total de células, onde não é possível executar o programa.

Sincronização no acesso a variáveis

No que toca à estrutura principal do programa implementado o mesmo pode ser visto como um programa sequencial executado simultaneamente em vários processadores que possuem a sua memória privada e trocam informação entre si. Deste modo a maioria das funcionalidades do programa não criam *data races* ou regiões críticas no acesso à memória, pelo que não existe necessidade de sincronizar o acesso à memória.

Assim, a maior preocupação encontra-se nas comunicações entre processadores. No código do programa pode-se detetar três zonas de comunicação: inicialização de partículas; partilha de bordas; envio de partículas perdidas. A inicialização de partículas é realizada apenas por processador (processador com id 0), onde após a sua inicialização são enviadas a todos os outros processadores acontecendo apenas uma única vez. A partilha de bordas acontece em cada iteração de tempo onde cada processadore comunica com os 8 processadores que o circulam. Depois da atualização das partículas estas serão enviadas para outros processadores, no caso destas deixarem de pertencer ao seu espaço de centro de massas. Para enviar estas partículas é necessário efetuar dois MPI_Send, um para enviar o número de partículas que serão enviadas e outro para enviar as próprias partículas, e MPI_Recv, para receber as partículas para serem guardadas.

Distribuição de cargas pelas *threads*

Como referido anteriormente foi usada a *checkerboard decomposition* de modo a dividir o melhor possível o conjunto de centros de massa por processadores. Tendo em conta que as partículas se vão encontrar uniformemente distribuídas, ou perto disso, pelos centros de massa, ao dividir os centros de massa igualmente pelos processadores, o mesmo também se sucede na distribuição de partículas.

Resultados observados

Nos vários testes efetuados com vários números de processadores (1, 2, 4, 8, 16, 32 e 64), observou-se que o resultado obtido era igual, senão muito semelhante, ao resultado obtido na versão sequencial tal como na solução fornecida. Contudo, em vários casos, ao executar o mesmo teste repetidamente, o tempo de execução varia drasticamente, pelo que é difícil tirar conclusões a cerca da *performance* do algoritmo. Isto pode dever-se às escolhas de implementação do algoritmo ou simplesmente à sobrecarga do *cluster*.

No entanto, foi feita uma pequena análise à escalabilidade do algoritmo. Para o caso do algoritmo sequencial a complexidade é $O(n^2)$.

Para o algoritmo em paralelo temos de observar a complexidade de computação tal como a de comunicação. No nosso programa a criação e atualização do *checkerboard* corresponde à região com maior complexidade, onde é apenas efetuada comunicação, sendo esta a que vai ser analisada.

Para a comunicação obtemos uma complexidade de $O\left(\frac{n}{\sqrt{p}}\right)$.

Assim, o *overhead* do algoritmo em paralelo é de $O\left(n\sqrt{p}\right)$.

Como o requerimento de memória é dado por $M(n) = n^2$, a escalabilidade será dada por:

$$n^2 \geq Cn\sqrt{p} \Leftrightarrow n \geq C\sqrt{p}$$

$$\frac{M(C\sqrt{p})}{p} = \frac{C^2 p}{p} = C$$

Desta forma, concluímos que o sistema possui uma boa escalabilidade teórica.

Ao nível da análise experimental utilizou-se o teste com os parâmetros ./simpar 4 20 100000 5 e variou-se o valor de ncside. Observou-se que aumentando ncside, o tempo de execução aumentava pouco. Isto leva-nos a concluir que realmente temos uma boa escalabilidade, apesar da inconsistência nos resultados, como se verifica na tabela 1. É de notas que em alguns testes o programa é executado com erros (testes identificados com X), e noutros casos o teste chega ao fim mas tem erros na libertação de memória (testes identificados com -).

Anexo

Processos / ncside	1	2	4	8	16	32	64	128	256
20	0.574	0.542	0.473	0.601	0.664	X	X	X	X
100	0.576	X	0.483	0.608	0.816	0.690	1.769	-	-
500	2.003	X	X	X	0.750	0.773	1.779	3.223	4.297
1000	0.880	X	X	X	X	X	1.721	2.964	3.446

Tabela 1 – Análise da escalabilidade

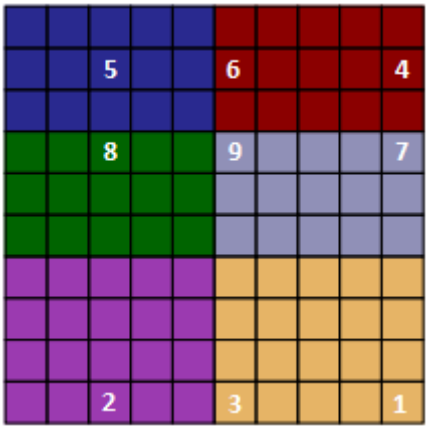

Figura 1 - Chekerboard decomposition

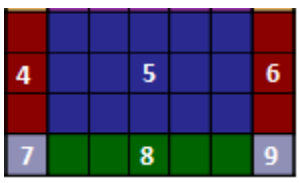

Figura 2 - Matriz de centros de massa alocada no processo azul