

COMPUTAÇÃO PARALELA DISTRIBUÍDA

Wolves & Squirrels

Grupo 17

Alunos:

Afonso Garcia, nº 70001

Duarte Patricio, nº 62564

Marcus Gomes, nº 69376

6 de Dezembro de 2013

Introdução

Este projecto teve como objectivo elaborar uma implementação paralela do problema Wolves and Squirrels. Para suportar o paralelismo será usada a *framework* MPI (Message Passing Interface). A seguir serão descritos alguns detalhes referentes a esta implementação, como a decomposição utilizada para o problema, o *load balancing* que foi feito entre as *threads* e os resultados de performance obtidos nos testes.

Decomposição do Problema

A decomposição utilizada para esta implementação do problema foi a distribuição dos blocos por linha. Assim cada processo fica responsável por processar um certo número de linhas. Para determinar o número de linhas que cada *thread* fica responsável por processar numa matriz de tamanho $N \times N$ utilizamos um tamanho fixo de linhas para cada *thread* denominado de *chunk*. O valor da *chunk* é obtido pela divisão do número de linhas da matriz pelo número de *threads* que estão a correr no programa, ou seja $N \div p$. Deste modo conseguimos garantir que cada *thread* fique responsável por processar o mesmo número de linhas, excepto no caso da última *thread* que pode ficar com um número maior ou menor de linhas para processar.

Load Balancing

O *load balancing* do programa esta distribuido de tal forma que a master *thread* tem uma carga de trabalho maior do que as outras *threads*. Isto acontece porque na nossa implementação esta *thread* é a responsável por fazer a leitura do *input* do programa e inicializar o mundo.

Para as demais *threads*, excepto a última, o *load balancing* é estático visto que todas elas tem o mesmo número de linhas para processar. Em relação a última *thread* pode acontecer que ela tenha uma carga de trabalho menor do que as demais, situação que pode ocorrer quando esta tem menos linhas para processar, ou pode acontecer que ela tenha uma carga de trabalho maior do que as outras, situação em que tem um número maior de linhas para processar do que as outras *threads*.

Resultados de Performance

Para avaliar a performance da solução implementada, foram realizados um conjunto de testes com os ficheiros disponibilizados pelo corpo docente. Estes testes foram executados em um cluster que possui máquinas com um processador Intel® i5 com 4 *cores*.

A seguir será feita uma análise dos resultados de performance obtidos pela implementação MPI do problema Wolves & Squirrels. Serão comparados os resultados obtidos pela versão serial e paralela de modo a obter os *speedups* alcançados. Para testar ambas versões do problema os exemplos em que a dimensão das matrizes varia entre 10×10 e 1000×1000 , e o número de gerações que o programa é executado varia entre 5 e 100000.

Implementação Sequencial

Os resultados de tempo de execução obtidos para a versão sequencial do programa são apresentados na tabela a seguir.

Dimensão	Iterações	Tempo
10x10	5	0.002s
10x10	1000000	6.692s
100x100	100000	49.639s
1000x1000	10000	51.066s

Implementação Paralela

Os testes com a implementação paralela do programa foram executados utilizando respectivamente 2, 4, 8, 16 e 32 processos. Devido ao facto de que estes testes foram executados num *cluster* partilhado, foi possível garantir que cada *core* do processador executasse apenas um processo de cada vez. Em seguida serão apresentados os resultados nos testes realizados.