Trabalho Prático 1 - Computação Paralela

1st João Vitor Vieira
Universidade do Minho
Braga, Portugal
PG50501

2nd José Pedro Magalhães
Universidade do Minho
Braga, Portugal
PG50528

***Abstract*—Este trabalho tem por base a implementação do algoritmo k-means fazendo uso de técnicas de localidade espacial, temporal e vetorização de modo a minimizar o tempo de execução do algoritmo na máquina virtual disponibilizada pelos docentes.**

***Index Terms*—kmeans, algoritmo, localidade, temporal, espacial, vetorização**

I. Introduction

No ambito da disciplina de Computação Paralela foi desenvolvido o algoritmo k-means com base no algoritmo de Lloyd que é utilizado para particionar dados não classificados (N pontos com coordenadas cartesianas) em K grupos (clusters).

O objetivo deste trabalho prático é implementar e tentar minimizar o tempo de execução do algoritmo recorrendo a técnicas de localidade espacial e temporal, vetorização e armazenamento inteligente dos dados em estruturas de dados que favoreçam o tempo de execução.

II. Estruturas de dados

A. Criação das Estruturas de Dados

Relativamente à criação de estruturas de dados, foi necessária a implementação da estrutura Ponto e Cluster.

A estrutura Ponto é constituida por um float x e y, que são as suas coordenadas cartesianas e o número do seu cluster.

A estrutura Cluster é constituida por um inteiro size, relativa ao seu tamanho. Contém também 2 apontadores para 2 Pontos, o temp_centroid e o centroid. O temp_centroid é o Ponto que serve para calcular cada novo centroid e comparar com o anterior.

Por fim, também foram criadas duas variáveis globais, sendo uma delas uma lista, pointList, com todos os N pontos, e uma lista, clusterList, com todos os K clusters.

B. Inicialização das Estruturas de Dados

Para a realização do algoritmo, começou-se por inicializar todos os pontos e clusters, através da função init.

Esta função, tal como o nome indica, é a responsável por alocar memória para todas as variáveis que serão utilizadas no algoritmo. É constituida por 2 ciclos, um com N de iterações e o outro com K, para os pontos e os clusters, respetivamente.

Os valores dos pontos são aleatórios e encontram-se todos entre 0 e 1, fazendo o uso da função rand. Já os clusters, apenas é dado um valor à variável centroid, o centro geométrico do cluster, que inicialmente corresponde aos K primeiros pontos, isto é, o cluster número 1, terá como centroid o ponto número 1 e assim sucessivamente.

Identify applicable funding agency here. If none, delete this.

III. Implementação do Algoritmo

O algoritmo k-means começa por escolher K centroids aleatórios ainda na inicialização das variáveis, seguidamente são iterados todos os N pontos.

Para cada um destes ponto (ponto _i) são iterados todos os K clusters de modo a identificar qual deles se encontra mais próximo do ponto_i através da distância euclidiana.

Depois de identificado o cluster mais próximo do ponto_i, os valores das suas coordenadas são acumulados numa variável do cluster e o ponto_i é adicionado à sua lista de pontos.

Quando termina o ciclo e todos os pontos se encontram nos clusters mais próximos, estes são iterados novamente calculando-se o novo centroid. Todo este processo é repetido até que as coordenadas dos novos centroids sejam iguais às da iteração anterior.

IV. Otimização do Algoritmo

A. Loop Unrolling

A técnica de Loop Unrolling tem como objetivo diminuir o número de instruções utilizadas para controlo de ciclos, fazendo mais instruções de cada vez e incrementando a variável de controlo de ciclo em mais do que uma unidade, tornado o programa mais rápido à custa de tornar o programa mais pesado a nivel de memória.

O programa realizado é compilado com a flag -funroll-loops o que ativa as otimizações de loop unrolling automaticamente.

B. Localidade espacial

Localidade Espacial ocorre quando se usa elementos de dados ou instruções que estão relativamente próximos em locais de armazenamento. Ou seja, quando se executa uma instrução ou se recolhe informação, esta já se encontra em cache uma vez que foi trazida para a memória por uma instrução anterior.

O algoritmo k-means faz uso de localidade espacial, por exemplo no acesso às coordenadas dos pontos no seu ciclo for mais interno visto que o acesso à variável x traz para memória a variável y.

C. Localidade temporal

Localidade Temporal ocorre quando dados ou instruções que foram acedidos recentemente têm grande probabilidade de serem novamente acedidos.

O algoritmo k-means faz uso de localidade temporal, uma vez que os pontos são iterados no ciclo for mais externo é sempre acedido o mesmo ponto em todas as iterações do ciclo for mais interno.

D. Vetorização

A Vetorização Automática é uma otimização do compilador que ao invés de realizar operações a elementos individuais em loop, realiza operações a vários elementos ao mesmo tempo através de operações vetoriais.

O programa realizado é compilado com as flags -ftree-vectorize e -msse4 o que ativa as otimizações de vetorização automaticamente.

E. Otimizações Matemáticas

Os elementos do grupo repararam ainda que algumas otimizações podiam ser feitas de modo a tornar o algoritmo ainda mais eficiente.

A fórmula normal da distância euclidiana faz uso da função ”sqrt”, no entanto o objetivo do programa passa por identificar qual o cluster mais próximo de um certo ponto, não sendo necessário calcular o valor real da distância, ou seja, se sqrt(a) sqrt(b) então a b, o que nos permite retirar por completo a função ”sqrt” poupando bastante tempo de execução.

F. Inlining

Inlining permite evitar a chamada a uma função substituindo-a pelo seu conteúdo.

A função ”pow” utilizada no cálculo da distância euclidiana é dispendiosa uma vez que é utilizada quatro vezes, para além disto a função contém um ciclo for para poder executar a operação de exponenciação n vezes.

Assim, esta função foi substituída pelo uso da operação de multiplicação, permitindo assim uma execução mais rápida do programa.

V. Validação do resultado e output do programa

A seguinte tabela mostra os resultados obtidos em média em cinco execuções do programa.

Time Elapsed	4.765 seconds
Instructions	25,517,524,850
Cycles	15,023,380,322
Insn per Cycle	1.70
CPI	0.6
L1 Cache Misses	263,887,868

VI. Organização do código e Makefile

Foi decidido separar o trabalho em 2 pastas: source e bin. Também foi criada uma Makefile para correr o programa baseado no template providenciado pelos docentes que inclui as as flags: -std=c99 -O2 -fomit-frame-pointer -funroll-loops -ftree-vectorize -msse4

VII. Conclusão

O trabalho proporcionou uma melhoria bastante significativa em relação às alterações possíveis para melhorar a eficácia de uma aplicação.

A realização do trabalho foi bastante enriquecedora, pois permitiu aprofundar e consolidar conceitos aprendidos em contexto sala de aula e consolidar e a adquirir competências que serão uma mais valia para o futuro.

Em suma, consideramos que em relação ao pretendido pelos docentes da cadeira para este trabalho, conseguimos uma boa resolução e uma boa interpretação do enunciado.

Bibliography

https://en.wikipedia.org/wiki/K-means_clustering

https://en.wikipedia.org/wiki/Automatic_vectorization

https://en.wikipedia.org/wiki/Locality_of_reference

https://en.wikipedia.org/wiki/Loop_unrolling

https://acervolima.com/diferenca-entre-localidade-espacial-e-localidade-temporal/

https://www.ieee.org/conferences/publishing/templates.html