

Projecto de Sistemas Operativos 2017-18

heatSim

Panorâmica

LEIC-A / LEIC-T / LETI
IST

Resumo

Este documento introduz o tema do projecto da cadeira de Sistemas Operativos 2017-18. É feita uma apresentação geral do problema a abordar assim como do faseamento do projecto. Cada fase terá um enunciado próprio, que complementa este documento.

1 Introdução

O projecto da disciplina de Sistemas Operativos pretende familiarizar os alunos com vários dos tópicos cobertos pela matéria, nomeadamente através da realização de programas que permitem aplicar na prática os conceitos expostos nas aulas teóricas. Em particular este projecto oferece aos alunos a oportunidade de:

- Realizar programas que exploram o paralelismo e que usam múltiplos fios-de-execução de forma concorrente.
- Experimentar os dois principais paradigmas de programação concorrente, nomeadamente a “troca de mensagens” e a “partilha de memória”.
- Ganhar prática na utilização de mecanismos de controlo de concorrência, nomeadamente na utilização de mutexes e variáveis de condição.
- Utilizar outros serviços fornecidos pelos sistemas operativos, como os sistemas de ficheiros e canais de comunicação, entre outros.

Para este fim, o projecto consiste na concretização de diversas versões de um programa para simular a propagação de calor numa superfície. Este é um exemplo clássico da programação científica que se adequa particularmente a ser paralelizado (isto é, o tempo de execução do programa, na resolução de problemas de grande dimensão, pode ser reduzido dividindo o trabalho por múltiplos fios de execução que se executam em paralelo).

O projeto de Sistemas Operativos 2017/18 está organizado em 5 exercícios encadeados (o primeiro exercício é preparatório e não conta para a avaliação). Este documento apresenta uma visão global do projeto e dos tópicos abordados em cada um dos exercícios. Cada exercício terá um enunciado próprio, que complementa este documento.

2 Problema e visão global do projeto

O projeto consiste em desenvolver um simulador chamado **heatSim**, que calcula a distribuição de calor numa superfície. O problema da computação da distribuição de calor é hoje em dia aplicado em diferentes domínios da Ciência e da Engenharia. Este problema coloca-se tanto em superfícies como em volumes mas, para simplificar o enunciado, consideraremos apenas superfícies quadradas.

Mais precisamente:

- A superfície quadrada é representada por uma matriz bidimensional de números. Cada valor $h_{i,j}$, onde $0 \leq i \leq N + 1$ e $0 \leq j \leq N + 1$, representa a temperatura desse ponto. Uma ilustração desta estrutura de dados é apresentada na Figura 1.
- Na matriz, podemos distinguir:
 - Os pontos interiores, dados por $h_{i,j}$ tais que $1 \leq i \leq N$ e $1 \leq j \leq N$. Ou seja, estes pontos estão dispostos numa matriz interior de $N \times N$ pontos.
 - Os pontos das arestas, dados por $h_{i,j}$ tais que $i = 0$ (aresta inferior) ou $i = N + 1$ (superior) ou $j = 0$ (esquerda) ou $j = N + 1$ (direita).

O problema consiste em calcular qual a temperatura de cada ponto da superfície, após o calor se ter propagado, sabendo qual a temperatura que é imposta em cada ponto das arestas.

Mais uma vez, para simplificar, vamos também assumir que cada aresta é sujeita a uma temperatura constante (embora arestas diferentes possam estar a temperaturas diferentes). Desta forma, a superfície pode ser sujeita, no máximo, a quatro temperaturas distintas nas arestas. Os valores respetivos são passados como argumentos ao simulador. Quanto aos pontos interiores, a sua temperatura inicial é um valor pré-definida; neste projecto, será sempre colocada a 0.

Note-se que, se todas as arestas forem sujeitas à mesma temperatura, todos os pontos interiores da superfície acabarão por ficar a essa temperatura. Adicionalmente, como os vértices da matriz não influenciam o algoritmo seguinte, estes podem ser preenchidos com qualquer valor.

A resolução deste problema passa, tipicamente, por uma solução iterativa.

- Em cada iteração, calcula-se uma nova temperatura de cada ponto interior com base na temperatura dos pontos vizinhos na iteração anterior. Para cada ponto, a sua nova temperatura é calculada pela média dos seus 4 pontos adjacentes. Ou seja:

$$h_{i,j} = \frac{(h_{i-1,j} + h_{i+1,j} + h_{i,j-1} + h_{i,j+1})}{4}$$

Note-se que os valores calculados na iteração i dependem dos valores da iteração $i - 1$. Desta forma é preciso manter pelo menos duas estruturas de dados para resolver este problema. Para poupar memória, não é criada uma nova estrutura de dados a cada iteração. Usam-se apenas duas estruturas: uma guarda os valores calculados na iteração anterior e outra que guarda os valores que estão a ser calculados na iteração actual. Para evitar copiar toda a matriz a cada iteração, o papel destas matrizes vai-se invertendo. Em pormenor, considere que possui a matriz M_0 e M_1 . Se na iteração i os valores da iteração $i - 1$ estiverem na matriz M_0 , os novos valores serão escritos na matriz M_1 . Na iteração seguinte, $i + 1$, usam-se como entrada os valores da matriz M_1 e escrevem-se os novos valores na matriz M_0 (e assim sucessivamente).

- À medida que cada iteração termina, as temperaturas dos pontos interiores convergem para aproximações de precisão cada vez maior. As iterações são repetidas até se atingir um número pré-definido de iterações ou até já não se verificarem alterações de relevo na temperatura dos pontos da superfície.

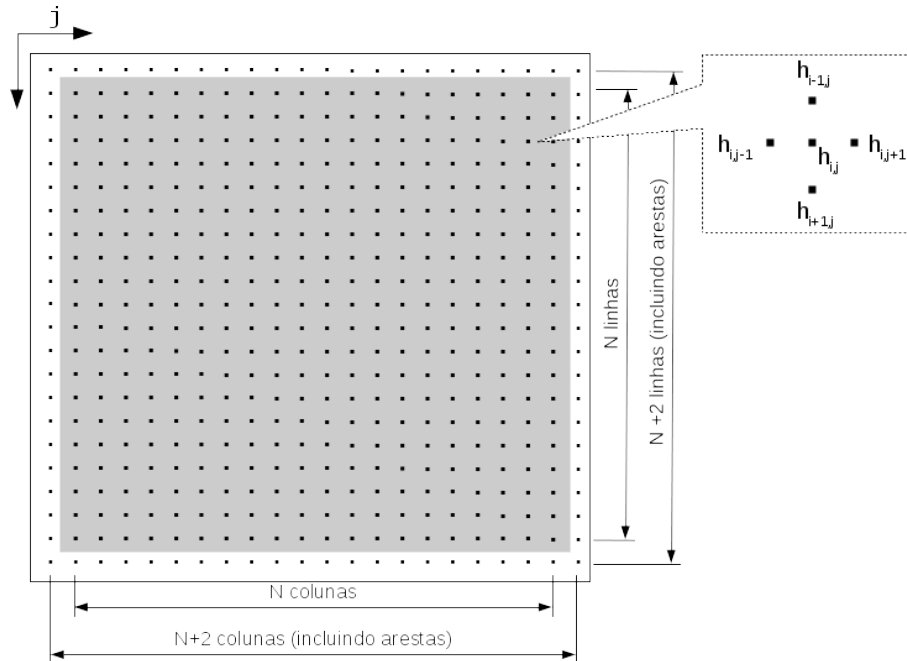


Figura 1: Representação da superfície em matriz de pontos. Os pontos sombreados correspondem aos pontos interiores, cuja temperatura deverá ser calculada. Os pontos sobre fundo branco correspondem às arestas, cuja temperatura é constante e fornecida como argumento do programa.

Para matrizes suficientemente grandes, este problema pode demorar muito tempo a ser resolvido de forma sequencial. Como tal, pretende-se que o `heatSim` aproveite a capacidade de processamento paralelo das arquiteturas multi-core modernas para obter um melhor desempenho.

3 Faseamento do Projecto

Os 5 exercícios que constituem o projeto abordam os tópicos seguintes (aqui apresentados de forma resumida):

- Exercício 0 - Este exercício é uma preparação para o projecto e não conta para a avaliação. Resume-se a fazer a versão sequencial do programa acima descrito a partir de um “esqueleto” que será fornecido pelo corpo docente.
- Exercício 1 - Consiste do desenvolvimento de uma versão paralela do problema recorrendo ao paradigma da troca de mensagens, recorrendo a uma condição simples de paragem. Para a resolução deste exercício, os alunos usarão uma biblioteca que concretiza a troca de mensagens.
- Exercício 2 - Estudo e melhoramento da biblioteca de troca de mensagens fornecida pelo corpo docente.
- Exercício 3 - Desenvolvimento de uma outra versão paralela, desta vez recorrendo ao paradigma da memória partilhada. Estenderá cada versão com uma condição de paragem dinâmica. Ambas as soluções (troca de mensagens e memória partilhada) poderão ser comparadas experimentalmente.

- Exercício 4 - Extensão do simulador com mecanismos de salvaguarda periódica do estado da simulação em memória persistente, que permitirão ao **heatSim** recuperar uma simulação após ter recebido ordem de suspensão ou ter sido interrompido devido a falha. Integração do simulador com um servidor que permite a visualização da matriz de forma mais intuitiva.