
HIGH PERFORMANCE COMPUTING

RELATÓRIO DA SECCÇÃO DE *HPC* DA DISCIPLINA DE TEI

Bernardo Marques, 18373
Instituto Politécnico de Beja
Engenharia Informática

Gonçalo Amaro, 17440
Instituto Politécnico de Beja
Engenharia Informática

Hugo Silva, 16570
Instituto Politécnico de Beja
Engenharia Informática

12 de abril de 2021

RESUMO

Projecto que pretende demonstrar o uso da computação de alto desempenho, com um exemplo real e de um mini-projecto da detecção de contornos de *Sobel* via programação *multicore* em *OpenMP*.

Keywords Contornos de *Sobel* · *OpenMP*

1 Introdução

O objectivo deste trabalho é clarificar o uso e o potencial da computação de elevado desempenho.

Este divide-se em duas partes: uma meramente explicativa sobre um projecto real com o uso desta categoria de computação: O Projecto *Kaleidoscope* da *Repsol* e, seguidamente, uma apresentação clara e detalhada do mini-projecto pedido: Detecção dos contornos de *Sobel* numa imagem *4K* (3840 x 2160 pixels) com programação paralela.

2 Projecto *Kaleidoscope* da *Repsol*

Kaleidoscope é um projecto de colaboração tecnológico entre a *Emerson* e a *Repsol* projectado para trazer tecnologias avançadas de imagem sísmica de sub-superfície para a indústria de petróleo e gás. Como parte do projecto, a *Emerson* implementará e implantará soluções avançadas de imagem de sub-superfície com base nas tecnologias centrais da *Repsol*. Combinando o que há de mais moderno em visualização computacional de ponta e em computação de alto desempenho, as soluções estarão disponíveis para a comunidade de geociências da *Repsol* e para todas as empresas de petróleo e gás que optem por licenciar as tecnologias, para apoiar os seus processos de transformação digital.

2.0.1 A máquina onde se processa

A *Repsol* aliou-se ao centro de super-computação de Barcelona, Espanha, a qual disponibiliza tempo-*CPU* do seu supercomputador *MareNostrum*.

O *MareNostrum* é uma máquina com capacidade de processamento de 13.7 *petaflops* e de armazenamento de 14 *petabytes* e é o maior supercomputador da Península Ibérica.

Este divide-se em dois blocos: um de propósito geral e um de tecnologias emergentes.

O bloco principal tem 48 *racks* com 3456 *nodes* com um Lenovo ThinkSystem SD530, que é composto por dois Intel Xeon Platinum com 24 *cores* (48 *threads*) cada, um total de 165888 processadores e uma memória *RAM* de 390 *terabytes*.

O segundo bloco (de tecnologias emergentes) é composto de três *clusters*, dos quais:

- Um é composto por *racks* de *GPUs*, nomeadamente racks com *NVIDIA* Volta *GPUs* e outros com *IMP POWER9*. Este *cluster* tem uma capacidade computacional de 1.5 *petaflops*.

- Outro composto por tecnologias da *AMD*, com *CPUs AMD EPYC* de arquitectura *Rome* e *CPUs Radeon Instinct MI50*. Este pequeno *cluster* sozinho tem uma capacidade de 0.52 *petaflops*, equivalente ao supercomputador *Frontier* na *Oak Ridge National Laboratory*.
- O ultimo é composto por processadores *ARMv8* desenhados para alto desempenho, contrariamente aos *ARMv8* que encontramos nos nossos dispositivos *mobile* (que são desenhados para o menor custo energético e térmico). Com um poder total de 0.65 *petaflops*.

2.0.2 O porquê?

Um *survey* geológico a uma área geográfica, tem um *dataset* de entre 10 a 20 *terabytes*, uma pequena simulação de fluidos num computador *desktop* comum, já requer algum tempo de renderização incluindo com paralelismo auxiliado pela *GPU*, logo com um *dataset* dessa larga escala, necessita de um outro nível de paralelismo e auxilio a algoritmos eficientes.

2.1 Como funciona?

É um sistema de simulação de fluidos altamente paralelizado com recurso a *datasets* precisos e recentes de dados geológicos. Este divide a área de cálculo, quer o modelo seja bi ou tri-dimensional em células com percepção e comunicação entre si, onde os fluidos (cujo os mesmo são compostos de partículas) quando sua parte sai dos limites da célula e entra na célula adjacente é comunicado o seu estado e características (movimento, força, etc.).

3 Projecto Algoritmo Detecção de *Sobel* paralelizado

O filtro *Sobel* é uma operação utilizada no processamento de imagem, aplicado sobretudo em algoritmos de detecção de contornos. Matematicamente este operador utiliza duas matrizes 3×3 que são convolvidas com a imagem original para calcular aproximações das derivadas - uma para as variações horizontais e uma para as verticais [1].

-1	0	1
-2	0	2
-1	0	1

Figura 1: X - Direction Kernel

-1	-2	-1
0	0	0
1	2	1

Figura 2: Y - Direction Kernel

3.1 A máquina onde se processa

Lenovo Legion Y530

- CPU: Intel i5-8300H (4 Cores e 8 *Threads*)
- GPU: Intel UHD 630 (+ NVIDIA 1050 4GB, porém desativada, no *modeset*)
- RAM: 16GB DDR4 2666MHz
- SSD: 256GB NVMe TLC
- HDD: 1TB SATA

3.2 Como funciona?

O Algoritmo de Detecção de Sobel calcula o gradiente da intensidade da imagem em cada pixel, dando a direcção da maior variação de claro para escuro e a quantidade de variação nessa direcção. Assim, obtém-se uma noção de como varia a luminosidade em cada pixel, de forma mais suave ou abrupta.

Com isto consegue-se estimar a presença de uma transição claro-escuro e de qual a orientação desta. Como as variações claro-escuro intensas correspondem a fronteiras bem definidas entre objectos, consegue-se fazer a detecção de contornos [2].

3.2.1 Como funciona? (O projecto)

Para compilar os programas usamos o comando *make*, que executa o *makefile*. Para executar os nossos executáveis, damos primeiro permissões de execução com o comando *chmod +x nome_do_binario* e de seguida executamos com *./nome_do_binario*.

Para cronometrar as execuções usamos o comando *time ./nome_do_binario*.

3.3 Diferença nos tempos de execução

Graças ao uso de paralelismo recorrendo a biblioteca OpenMP, conseguimos acelerar o processamento da imagem, tal como podemos demonstrar nas cronometragem da execução dos binários.

```
01:40:33 gonca@LEGIONY530 TEI-ImageMP ±|master x|→ time ./sobelSingle
real    0m3,092s
user    0m1,857s
sys     0m0,284s
01:40:42 gonca@LEGIONY530 TEI-ImageMP ±|master x|→ time ./sobelMulti
real    0m2,096s
user    0m1,828s
sys     0m0,189s
01:40:50 gonca@LEGIONY530 TEI-ImageMP ±|master x|→
```

Figura 3: Diferença entre os timings

Referências

- [1] Sobel operator - Wikipedia, the free Encyclopedia https://en.wikipedia.org/wiki/Sobel_operator.
- [2] José Jasnaú Caeiro *Aulas online da disciplina*. (2021)