

Relatório PPC – Atividade 3

Floyd-Warshall



Ciências
ULisboa

Henrique Barata fc54387

Foi utilizado o Google Colab para correr o programa com a GPU Tesla T4.

A description of the necessary modifications for GPU-execution

Foi alocada memória na GPU para uma variável onde de seguida foi copiado o graph gerado pelo CPU. Como o graph é uma matrix de inteiros de $\text{graph_size} \times \text{graph_size}$, foram reservados na memória $\text{sizeof(int)} \times \text{graph_size} \times \text{graph_size}$ bytes. Assim, com os dados na memória da GPU, foi possível resolver então o Floyd-Warshall algorithm.

Finalmente, foi copiado o resultado de volta para a memória do CPU e libertou-se o espaço alocado na memória da GPU.

The rationale of the choice of parameters for GPU-execution (blocks, threads) and memory usage or transfers.

Para paralelizar o processo, foi criada uma função para ser corrida pela GPU, onde cada thread trata de um par (i,j) , para cada valor de k . O par escolhido depende do bloco e da thread id em questão. Foi usado dim3 para 2D, pois o graph é representado por uma matrix, i por j , para passar a block dimension e o número de threads.

Com o graph_size de 2 000 são necessárias 4 000 000 de threads, e criar mais seria apenas inútil. Depois de testar vários valores como parâmetros, usando múltiplos de 2 para o número de threads, conclui que com 64 threads por bloco e 62 500 blocos, é o melhor valor, obtendo um tempo de 0.55s o que foi um melhoramento, visto que sequencialmente o programa demorou 28.6s, havendo um speedup de 52x.

An explanation of whether your program suffer from thread divergence or not.

O programa não sofre por thread divergence, pois cada thread corre o mesmo código independentemente, executando todas ao mesmo tempo e devem acabar também ao mesmo tempo, pois não há threads com mais trabalho que outras.