Relatório $EP2$ — MAC 0219

22 de junho de 2018

1 Alunos

- **André Luiz Abdalla Silveira** – 8030353
- **Mauricio Luiz Cardoso** – 6796479

2 Metodologia e Funções

Lista de funções importantes:

1. **os_menores** [1] – função responsável pelas comparações. Para cada posição especificada na variável index, a função compara o valor do elemento apontado com os elementos correspondentes na distância de `jump` [2]
2. **menorMatriz** – define os prâmetros e argumentos das chamadas de sistema para a GPU. Toma como argumentos a matriz no device e a quantidade de matrizes a levar em conta.
3. **encontraMenorMatriz** – função preparatória para a anterior. Copia as matrizes lidas do arquivo para um pedaço de matéria alocada na GPU e libera depois do fim da função

Nosso EP é implementado em múltiplas fases. Depois de alocar o espaço das matrizes, aglutinadas todas em um grande vetor, colocamos esse ponteiro para ser copiado na função 3.

O próximo passo é chamar a função 2 que é responsável por determinar as chamasdas de sistema a serem feitas à GPU. A ideia é aplicar a solução para todas as matrizes no primeiro momento, e para uma parte bem mais reduzida posteriormente, de modo que a resposta fique armazenada na primeira matriz que será exibida ao final da execução.

3 Viabilidade

Durante a implementação do EP, percebemos que operar a redução de S em relação a $\oplus$ na GPU é possível, mas sua implemetação fatalmenete será sequencial em algum ponto. Isto é, naturalmente há uma ordem na qual as operações deverão ser feitas/consideradas para que não haja condição de corrida ou que para garantir a correção do algoritmo.

No nosso caso, dentro da função 1, cada thread olha um punhado de elementos das matrizes, e uma thread não acessará uma posição de memória que foi ou será acessada por outra, isso garante que duas threads não competirão pela mesma memória que não compararão o mesmo elemento mais de uma vez. Mas inegavelmete trata-se de um trecho sequencial.

[1] GPU

[2] `numBlocks * numThreads`