

Grupo:

Tallys Assis de Souza

Eros Henrique Rodrigues Oliveira do Monte

Bryan Jonathan Melo de Oliveira

Daniel Soares França

| | Sequencial | Paralela |
|----------------------|------------|----------|
| Uso de CPU | 0,997 | 3,930 |
| Frontend cycles idle | - | 0 |
| Backend cycles idle | - | 0 |
| IPC | 0,81 | 0,37 |
| LL cache hits | 68,03% | 69,59% |
| Time elapsed | 0,670s | 0,386s |

Observa-se que a versão paralela utiliza mais CPU (3.930) do que a versão sequencial (0.997). Isso é uma indicação de que a versão paralela pode ser mais rápida do que a sequencial, mas não é possível afirmar que ela é mais eficiente. A versão paralela tem um IPC menor (0.37) do que a versão sequencial (0.81), o que indica que, em média, a versão sequencial executa mais instruções por ciclo do processador. Isso pode ser um indicativo de que a versão paralela tem gargalos.

A taxa de falta na cache L3 (LL-cache hits) é semelhante para ambas as versões dada uma margem de erro de 2%, com uma taxa um pouco maior para a versão paralela (69.59% contra 68.03%). Isso pode ser um indicativo de que a versão paralela está acessando mais frequentemente a cache L3.

O tempo total de execução da versão paralela é menor (0.386 s) do que o da versão sequencial (0.670 s). Isso indica que a versão paralela é mais rápida do que a sequencial.

Os gargalos da aplicação paralela podem ser identificados pela baixa utilização dos ciclos ociosos na ULA (Frontend e Backend). Isso indica que a aplicação paralela está sendo limitada por outras operações, como a busca de dados na memória ou a comunicação entre os núcleos. Uma otimização possível seria tentar reduzir a comunicação entre os núcleos, por exemplo, compartilhando dados através de variáveis compartilhadas em vez de copiá-los de um núcleo para outro, mesmo que isso use mais memória no geral. Além disso, pode-se tentar melhorar o algoritmo utilizado para a aplicação para tentar reduzir a carga de trabalho em cada núcleo e assim melhorar a utilização dos ciclos ociosos na ULA.