



Disciplina: EA876 – Introdução a Software de Sistema

Discentes: Guilherme B. Abreu RA: 173691

André M. Yoshida RA: 166241

Data: 08/06/2018

Análise de desempenho entre programas Multithread e Multiprocesso

Introdução

O objetivo do experimento realizado era o de comparar o desempenho de dois programas que realizassem computação paralela para cumprir uma mesma tarefa utilizando o mesmo padrão de design, porém, um deles seria paralelizado com threads e o outro com processos. Além disso, o único fator de desempenho considerado foi o tempo gasto pelo programa somente ao exercer o trabalho.

Para fazer a análise, a tarefa escolhida foi aplicar um filtro blur em uma imagem que pode ser representada por 3 matrizes (uma matriz para cada canal no sistema RGB). Em um filtro blur de tamanho N , novas matrizes são geradas, e para cada uma delas, o pixel (i, j) recebe o valor da média de todos os pixels que estão dentro do quadrado que vai de $(i - N, j - N)$ a $(i + N, j + N)$ da matriz original.

Método

O método utilizado para paralelizar os dois programas é baseado no padrão de design Thread Pool. Uma lista de tamanho igual ao número de pixels da imagem é compartilhada entre os processos (ou threads) e nela é guardada o estado de cada pixel, isto é, se o filtro já foi aplicado naquele pixel ou não. No início do programa esta lista é inicializada com nenhuma posição marcada. Cada processo percorre esta lista procurando por posições não marcadas. Quando um processo encontra uma dessas posições, ele aplica o filtro no pixel que está relacionado com aquela

posição, e então ela é marcada. Assim, a tarefa é realizada apenas uma vez em cada pixel.

Para realizar o experimento foi utilizado um computador com 4 processadores lógicos, operando em uma frequência base de 2.9GHz com arquitetura Intel i7-7500U em sistema operacional Ubuntu 16.04 LTS.

Resultados

Neste ambiente, para um filtro blur com N igual a 10, utilizando uma imagem com resolução 1920 x 1280, e com uma amostra de 100 testes, foram obtidos os seguintes valores de tempo médio de execução:

- Com 4 processos: 3.0094 ± 0.1697 [s]
- Com 4 threads: 2.8815 ± 0.1481 [s]

Discussão e Conclusão

Analisando os resultados, podemos observar que apesar do tempo médio obtido utilizando threads ter sido menor, não podemos inferir que os threads são mais rápidos que os processos. Isso porque considerando os desvios padrão, vemos que há uma faixa de valores onde os processos foram mais rápidos que os threads.

Portanto, podemos notar que threads e processos têm desempenhos muito próximos. Mas, mesmo que não sejamos capazes de garantir que um será mais rápido que o outro, sabemos que em média threads terão uma eficiência maior.