



Disciplina: EA876 – Introdução a Software de Sistema

Discentes: Guilherme B. Abreu RA: 173691

André M. Yoshida RA: 166241

Data: 12/06/2018

## **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, sendo que um deles seria paralelizado com threads e o outro com processos. Além disso, o único fator de desempenho que seria considerado era 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) onde cada posição da lista representa um pixel. Nela é guardado o estado dos pixels, isto é, se eles já foram filtrados ou não. Todos os processos percorrem a lista procurando por aqueles que não foram filtrados, quando eles os encontram os mesmos são processados, e concomitantemente, a lista é atualizada para demarcá-los como filtrados. Isto ocorre até que toda a imagem seja processada.

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. Neste ambiente, com um filtro blur de  $N$  igual a 10 foram realizados 100 testes em uma imagem com resolução 1920 x 1280.

### **Resultados**

Utilizando os métodos esclarecidos acima, foram obtidos os seguintes valores de tempo médio de execução com um único processo, com 4 processos e com 4 threads atuando, respectivamente:

- ❖  $6.6762 \pm 0.1577$  [s]
- ❖  $3.0094 \pm 0.1697$  [s]
- ❖  $2.8815 \pm 0.1481$  [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 foram 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. Além disso, fica claro que a aplicação do paralelismo nessa tarefa obteve um grande aumento de desempenho.

Assim, podemos notar que threads e processos tiveram desempenhos muito próximos, e que a aplicação que utilizou threads obteve uma eficiência maior, considerando a amostra e o ambiente em que foram testados.