

# Algoritmo Paralelo do Teorema de Convolução em Imagens em OpenMP para Java

Henrique Yoshikazu Shishido<sup>1</sup>, Alexandre Huff<sup>2</sup>

<sup>1</sup>Coordenação de Engenharia da Computação  
Universidade Tecnológica Federal do Paraná – Câmpus Cornélio Procopio

<sup>2</sup>Coordenação de Técnico em Informática  
Universidade Tecnológica Federal do Paraná – Câmpus Toledo

{shishido, alexandrehuff}@utfpr.edu.br

## 1. Introdução

As aplicações científicas e comerciais tem demandado poder computacional cada vez maior. Para atender essa demanda, o mercado de processadores vem criando tecnologias como processadores *multicore*. Porém, o uso de parte ou de todos os núcleos em uma aplicação não é trivial, pois é necessário criar múltiplos *threads*, cujo controle e ajustes de algoritmos nem sempre são de conhecimento dos programadores (GEER, 2005).

A linguagem Java possui suporte nativo a *threads*, porém necessita-se tempo e conhecimento apurado para extrair o paralelismo de códigos sequenciais. A partir dessas dificuldades e com base na biblioteca OpenMP, foi desenvolvida a biblioteca JOMP que visa explorar o paralelismo em Java com menor dificuldade (BULL *et al.* 2000).

O objetivo desse trabalho foi analisar o ganho de desempenho da paralelização da biblioteca JOMP em processadores *multicore*.

## 2. Materiais e Métodos

O hardware utilizado nos experimentos foi um processador Intel Xeon E5620 2.4Ghz com 8 núcleos, 12MB de memória cache e 8GB de memória RAM. O sistema operacional escolhido para a execução dos experimentos foi o Linux CentOS 5 com kernel 2.6.18 x86. Além disso, foi utilizada a versão 1.0 beta da biblioteca JOMP.

O algoritmo desta avaliação é fundamentado no processo de convolução aplicado em processamento de imagens (TURNER, 1989) (GONZALES e WOODS, 2002). Assim, foi utilizada a versão sequencial do algoritmo desenvolvido pelo Grupo de Pesquisa em Geodésia da Universidade Estadual de São Paulo (UNESP) – Campus Presidente Prudente. O experimento consiste na execução do algoritmo com 1 a 16 *threads* para explorar os oito núcleos do processador em questão.

## 3. Resultados e Discussão

O *speedup* obtido na execução desse experimento é apresentado na Figura 1. A execução sequencial com 1 *thread* levou aproximadamente 1699 segundos. Já a execução com 2 *threads* ocorreu em 824 segundos. O uso de 4 *threads* apresentou redução de tempo em aproximadamente 4.05 vezes se comparado com a versão sequencial. Da mesma forma, a utilização de 8 *threads* acelerou o tempo de execução em 7.68 vezes.

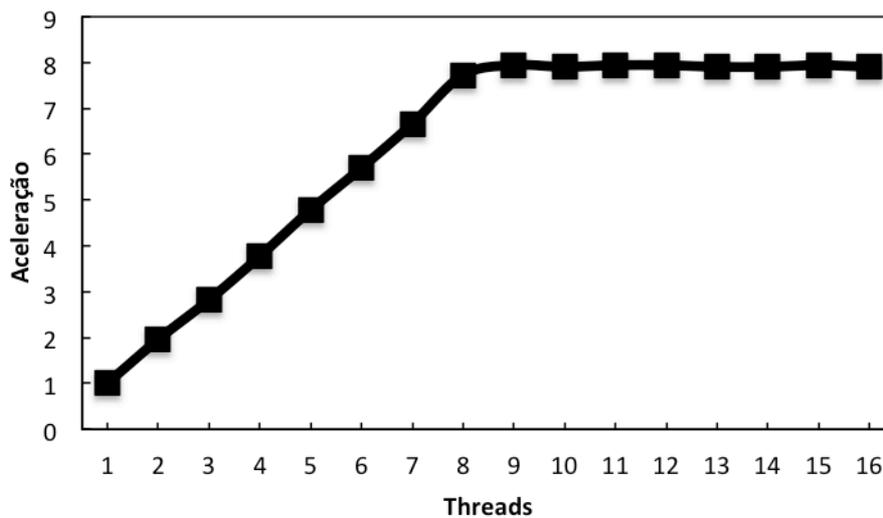


Figura 1. Speedup obtido na execução do algoritmo de convolução

De acordo com a Figura 1, nota-se que o ganho de desempenho aproxima-se proporcionalmente a adição de novas threads. Contudo, é importante destacar que a extrapolação do número de *threads* em relação à quantidade de núcleos deixa de ser vantajosa a partir da 8ª *thread*, devido à falta de núcleos para o escalonamento de algumas *threads*, refletindo negativamente no desempenho do algoritmo.

#### 4. Considerações Finais

O objetivo deste trabalho foi analisar desempenho oferecido pela biblioteca JOMP. O desempenho desta biblioteca mostrou-se muito satisfatório em um algoritmo que não possui dependência de dados. Para um processador com 8 núcleos pode-se atingir uma aceleração de até 7.68 vezes maior do que a versão sequencial do algoritmo. Entretanto, o número de *threads* inicializadas deve respeitar o número máximo de núcleos, evitando que ocorra um *overhead* no ambiente computacional. Da mesma forma, pretende-se posteriormente realizar a codificação e comparação de desempenho do mesmo algoritmo aplicado neste trabalho utilizando OpenMP em linguagem C/C++.

#### 5. Referências

- BULL, J. M.; KAMBITES, M. E. *JOMP: an OpenMP-like interface for Java*. In Proceedings of the ACM 2000 Java Grande Conference. pp.44–53, Junho, 2000.
- GEER, D. *Chip makers turn to multicore processors*. Computer, IEEE, v. 38, n. 5, pp. 11–13, 2005.
- GONZALES, R. C.; WOODS, R. E. *Digital Image Processing*. 2ed. Upper Saddle River, NJ: Prentice-Hall, 2002.
- TURNER, M. *Landscape ecology: the effect of pattern on process*. Annual Review of Ecology and Systematics, JSTOR, v.20, pp.171–197, 1989.