



## Comparação de diferentes interfaces de programação paralela aplicadas a processamento de imagens

Luan Pereira Vargas, discente de graduação, Universidade Federal do Pampa, Campus Alegrete

Arthur Francisco Lorenzon, docente, Universidade Federal do Pampa

E-mail – [luanvargas.aluno@unipampa.edu.br](mailto:luanvargas.aluno@unipampa.edu.br)

Diversos problemas da área de processamento de imagens demandam um alto esforço computacional, como, por exemplo, os métodos de *inpainting* baseados na replicação de patches. Estes métodos viabilizam a solução de problemas reais, como a reconstrução de regiões sem conteúdo em imagens. Portanto, eles podem se beneficiar da exploração do paralelismo no nível de threads (TLP) através de interfaces de programação paralela (IPPs). No entanto, como cada IPP possui diferentes características com respeito ao gerenciamento de threads, escolher a ideal para implementar uma aplicação é importante para obter o melhor custo-benefício entre desempenho e consumo de energia, representado pelo *energy-delay product* (EDP). Considerando o exposto, neste trabalho, nós analisamos o potencial de exploração de paralelismo de um algoritmo de *inpainting* amplamente difundido na literatura com diferentes IPPs (PThreads, OpenMP, OmpSs-2 e OpenACC) e mostramos qual IPP proporciona o melhor desempenho, consumo de energia e EDP para três arquiteturas *multicore* e duas GPUs. Através de um conjunto de experimentos, os resultados mostram que OpenMP explorando TLP com laços paralelos é melhor para processadores AMD, enquanto que o OmpSs-2 apresenta melhores resultados nos processadores Intel.

O tempo de execução de cada aplicação foi obtido pela função *get\_time\_of\_day()*. Por outro lado, o consumo de energia foi obtido diretamente dos contadores de hardware presentes nos processadores modernos. No caso do processador Intel Xeon, a biblioteca *Running Average Power Limit* (RAPL) foi usada, enquanto que a biblioteca *Application Power Management* foi usada para o processador AMD Ryzen. Já o consumo energia de todo o sistema executando as aplicações na GPU foi obtido através do IPMI (*Intelligent Platform Management Interface*). O EDP da execução de cada aplicação foi obtido pela multiplicação do tempo de execução pelo consumo de energia.

Neste trabalho, nós analisamos o potencial de exploração de paralelismo de um algoritmo de *inpainting* altamente difundido na literatura e propôs implementações paralelas com diversas IPPs: PThreads, OpenMP, OmpSs-2 e OpenACC.

Através da execução em três arquiteturas *multicore* e duas GPUs, mostramos que OpenMP é melhor para processadores AMD enquanto que OmpSs-2 é melhor para

processadores Intel.

Adicionalmente, dada a natureza de dependências do algoritmo de *inpainting*, mostrou-se que sua implementação paralela não é eficiente em arquiteturas GPUs. Como trabalhos futuros, pretendemos expandir o número de aplicações reais utilizadas na área de processamento de imagem.

**Agradecimentos:** Este trabalho foi parcialmente financiado pela FAPERGS nos projetos 19/2551-0001224-1 e 19/2551-0001689-1 e PIBIC-CNPq

**Palavras-chave:** Programação paralela, processamento de imagens, IPPs, otimização.