



ANÁLISE DE DESEMPENHO DE UM ALGORITMO DE ESQUELETIZAÇÃO DE IMAGENS EM ARQUITETURA NVIDIA CUDA

João E. Tozzi de Souza – Ifes Campus Serra - joatozzi@gmail.com

Flávio S. L. de Souza - Ifes Campus Serra – flaviolamas@ifes.edu.br

Jefferson O. Andrade - Ifes Campus Serra - joandrade@ifes.edu.br

Karin S. Komati - Ifes Campus Serra - kkomati@ifes.edu.br

Gestão, Tecnologia e Desenvolvimento de T.I / Ferramentas de Tecnologia da Informação

A utilização de Unidades de Processamento Gráfico para Propósito Geral (GPGPU) tem crescido muito nos últimos anos. Uma das arquiteturas que se utilizam desse conceito é a arquitetura CUDA da Nvidia, que consegue aumentos significativos de performance computacional repartindo o código em vários “pedaços” (*threads*) e o processando de forma paralela nas GPUs de uma placa de vídeo. Um de seus limitadores é a forma como o programador reparte o código em *threads* e como as organiza as mesmas em blocos, para aproveitar ao máximo o processamento das GPUs. Esse trabalho apresenta uma análise de desempenho de um algoritmo de esqueletização de imagens já implementado em CUDA, testando algumas formas de divisão dos *pixels* da imagem a ser processada por *thread* a fim de entender melhor o funcionamento da arquitetura CUDA e tentar encontrar um melhor desempenho para algoritmo em questão. O objetivo é gerar um gráfico de desempenho e através dele analisar a eficiência de um algoritmo de esqueletização que utiliza a arquitetura CUDA, a fim de encontrar a melhor configuração a ser utilizada pelo algoritmo no que diz respeito ao número de *threads*, blocos e *pixels* por *thread*. Nos testes, modificou-se o valor de três variáveis do código ligadas diretamente ao CUDA: **Blocksize**, que define a quantidade de *threads* de um bloco, **nBlocks**, que define o número de blocos e **Blockage** que define basicamente a quantidade de *pixels* que será processada por *thread*. Para cada configuração de **Blocksize**, foi testada a quantidade de 1 a 512 *pixels* da imagem por *thread* na variável **Blockage**, a fim de tentar identificar o efeito produzido pela quantidade de *pixels* por *thread* no desempenho do código. Já para calcular **nBlocks**, foi utilizada a seguinte fórmula: $nBlocks = \lceil \frac{\text{tamanho_da_imagem}}{(\text{Blocksize} * \text{Blockage})} \rceil$. Com base nos resultados, é possível chegar à conclusão de que a configuração mais eficiente em relação as demais foi com o **Blocksize** = 128. Só foi utilizado **Blocksize** onde os multiprocessadores pudessem trabalhar em sua carga total. O melhor resultado obtido foi alcançado distribuindo 6 *pixels* por *thread* e 128 *threads* por bloco, fazendo com que cada SM (*Streaming Multiprocessors*) trabalhasse por ciclo com 6 blocos.

Palavras-chave: CUDA, algoritmo de esqueletização de imagens, GPU

Instituição de fomento: Ifes