Comparação da Desempenho da Divisão Tridimensional de uma Aplicação Stencil Desenvolvida com StarPU

Gabriel Freytag[1], **João Vicente Ferreira Lima**[1]

[1]Programa de Pós-Graduação em Informática – Universidade Federal de Santa Maria (UFSM)
97.105-900 – Santa Maria – RS – Brazil

{gfreytag,jvlima}@inf.ufsm.br

Resumo. *Parte do potencial que arquiteturas heterogêneas oferecem não é explorado por aplicações. Sistemas como StarPU possibilitam a exploração de todo o potencial com um alto nível de abstração e dinamicidade. Este trabalho busca avaliar a desempenho da adaptação de um algoritmo de Stencil tridimensional desenvolvido com StarPU e resultados parciais mostram que há ganhos em problemas maiores, mas há também um alto consumo de memória.*

1. Introdução

Dispositivos *multicore* e *manycore* se tornaram populares nos últimos anos. Computadores, consoles de vídeo game e, inclusive, *smartphones* atualmente possuem processadores com vários núcleos e GPUs com dezenas, centenas e até milhares de núcleos de processamento. A existência de vários núcleos possibilita a utilização do paralelismo para otimizar o desempenhos desses e outros dispositivos.

As GPUs foram desenvolvidas para prover eficiência energética e ganhos de desempenho brutos e, portanto, normalmente são responsáveis pelo processamento de partes de computação intensiva e de dados intensivo de uma aplicação, enquanto CPUs executam tarefas que não são *kernels*, ou seja, partes sequenciais de aplicações [Gaster et al. 2012]. CPUs normalmente são responsáveis pelo controle do processamento de aplicações e, dessa forma, enviam partes de computação e dados intensivo para as GPUs e então aguardam ociosas o retorno dos dados.

No entanto, sistemas como o StarPU [Augonnet et al. 2009b] possibilitam o processamento simultâneo de partes sequenciais e de processamento e dados intensivo, otimizando a execução de aplicações, pois segmenta aceleradores (GPUs, Cell's SPUs, etc.) e processadores *multicore* simultaneamente e de forma portátil [Augonnet et al. 2009a]. StarPU possui várias implementações de exemplo e uma delas é a de um padrão de computação paralela chamada Stencil.

Este trabalho tem como objetivo investigar os ganhos de desempenho da adaptação desse problema para a divisão não só do eixo Z em vários blocos para processamento paralelo, mas também dos eixos X e Y. Além disso, este trabalho também busca comparar o consumo de memória de ambas as abordagens. Na Seção 2 são apresentados alguns resultados parciais realizados em um servidor com Linux Debian 3.2.73-2+deb7u2 x86_-64, CPU Intel Xeon E5620 de 2.40GHz e 4 núcleos e 4 *threads*, 12GB de memória RAM e 2 GPUs Nvidia Tesla T20 e na Seção 3 algumas conclusões e trabalhos futuros.

2. Resultados

A implementação de Stencil presente no código fonte do sistema StarPU divide um problema tridimensional em vários blocos menores ao longo do eixo Z. Cada bloco armazena as coordenadas iniciais de cada bloco vizinho anterior e posterior. No entanto, a tese que baseia este trabalho é a de que a divisão em um número maior de blocos dividindo também ao longo dos eixos X e Y pode aumentar o paralelismo e a desempenho do algoritmo.

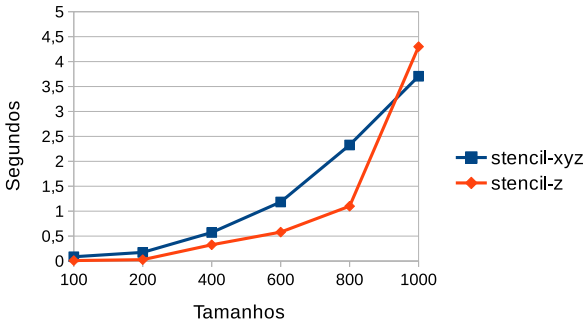

Figura 1. Tempo de computação

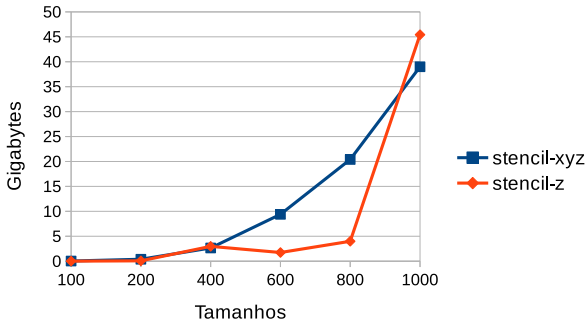

Figura 2. Transferência de dados

A Figura 1 mostra o tempo de execução da divisão de problemas com vários tamanhos em 4 blocos no eixo Z (implementação presente no código fonte do StarPU), nomeada stencil-z, e 4 blocos nos eixos X, Y e Z na implementação adaptada por este trabalho, nomeada stencil-xyz, somando 64 blocos. Cada configuração de tamanho do problema foi executado dez vezes e o tempo apresentado é a média dessas execuções.

Na Figura 2 é possível observar o consumo de memória por cada implementação. O consumo de memória é a soma dos bytes alocados para o processamento de cada bloco. É possível observar que há uma relação entre o consumo de memória e o tempo de computação. Isso se deve pelo fato da alocação de memória ser maior conforme o tamanho do problema aumenta.

3. Conclusão

O número relativamente maior de blocos da divisão nos três eixos aumentou o consumo de memória uma vez que cada bloco armazena as coordenadas iniciais de seis blocos vizinhos frente a apenas dois da implementação original. Como trabalho futuro é sugerido a realização de testes com problemas maiores e outros escalonadores de tarefas do StarPU.

Referências

Augonnet, C., Thibault, S., and Namyst, R. (2009a). Automatic Calibration of Performance Models on Heterogeneous Multicore Architectures. *European Conference on Parallel Processing*, (Hppc):56–65.

Augonnet, C., Thibault, S., Namyst, R., and Wacrenier, P.-A. (2009b). StarPU: A unified platform for task scheduling on heterogeneous multicore architectures. In *Euro-Par 2009 Parallel Processing*, pages 863–874.

Gaster, B., Howes, L., Kaeli, D., Mistry, P., and Schaa, D. (2012). *Heterogeneous Computing with OpenCL: Revised OpenCL 1*. Newnes.