# Escalonamento em Arquiteturas Heterogêneas - APUS

**Anderson Uilian Kauer, Mozart Lemos de Siqueira**

UNILASALLE – Centro Universitário La Salle
Av. Victor Barreto, 2288, Centro – 92010-000 – Canoas – RS – Brasil

andersonkauer@gmail.com, mozarts@unilasalle.edu.br

***Resumo:*** *Processadores com múltiplos núcleos de processamento têm se tornado popular na área de computação de alto desempenho. Processadores gráficos (GPUs) estão sendo cada vez mais utilizados para computação de propósito geral. Nesse trabalho, mostramos que a adequada coordenação de núcleos de processamento heterogêneos pode melhorar significativamente o desempenho das aplicações.*

## 1 Introdução

Com os avanços nas arquiteturas de computadores, principalmente a popularização dos processadores *multi-core* tem transformado os ambientes distribuídos em arquiteturas hierárquicas e muitas vezes heterogêneas, juntamente com a utilização de GPUs (*Graphics Processing Units*) como unidades de computação para propósito geral, em virtude de estas melhorarem significativamente o desempenho de aplicações em muitos cenários.

Nos últimos anos muitos trabalhos afirmam que GPUs atingem velocidade entre 10x e 1000x maiores comparados com CPUs *multi-core*. No entanto, recentes estudos baseados nestes resultados identificaram um aumento médio de 2,5x (LEE et al., 2011).

Portanto, para uma melhor utilização destas plataformas é preciso observar a adequação de cada uma das tarefas de computação envolvidas aos processadores disponíveis, de forma que sejam distribuídas de maneira eficiente em ambientes heterogêneos equipados com GPUs e CPUs *multi-core.*

Também é importante considerar gargalo de comunicação entre CPU e GPU, atualmente sendo provida pelo barramento PCIe, que tem velocidades de transferência inferior comparados com comunicação direta entre núcleos no mesmo chip.

No entanto novas arquiteturas heterogêneas tais como APU (*Accelerated Processing Units*) ou Unidades de Processamento Acelerado que consiste em dispor núcleos CPU e GPU no mesmo chip. Nestas arquiteturas toda a comunicação entre os processadores é feita através de em um barramento em comum. A computação heterogênea pode ser beneficiada ao utilizar essa abordagem, pois sugere que alguns dos problemas de comunicação sejam contornados.

O problema de escalonar aplicações em ambientes heterogêneos é difícil de ser resolvido por ao menos três razões: *(i)* em termos gerais é amplamente conhecido como NP-completo; *(ii)* as tarefas relacionadas à execução da aplicação são criadas em tempo de execução, assim, qualquer escalonamento estático é inviável; *(iii)* estimar o tempo de execução de aplicações/tarefas é um problema em aberto (TEODORO, 2010 apud Fernandez, 1989; Maheswaran et al., 1999; Fahringer, Zima, 1993; Kerbyson et al., 2001).

A contribuição deste trabalho está associada à investigação do escalonamento particionado em ambientes heterogêneos integrados (APUs) considerando as características das tarefas para os processadores envolvidos e as variações no tamanho do problema.

## 2 Arquiteturas Heterogêneas

Arquiteturas heterogêneas têm ganhado bastante espaço na área da computação de alto desempenho, isso se deve principalmente à popularização de processadores gráficos cada vez mais eficientes, que apesar de disponibilizarem frequências de ciclo relativamente inferiores aos processadores *multi-core* atuais, sua grande quantidade de núcleos simples podem executar centenas de cálculos ao mesmo tempo, fazendo com que, em alguns casos, possam apresentar desempenho muito superior aos processadores tradicionais e frequentemente consumindo uma menor quantidade de energia.

GPGPU (*General-Purpose computation on Graphics Processing Units*) é o conceito utilizado para designar a utilização de GPUs para computação de propósito geral. Inicialmente, GPGPU estava limitado às linguagens de programação proprietárias tais como CUDA. Posteriormente, OpenCL (Khronos Group, 2012) surgiu como um padrão para o desenvolvimento de aplicações utilizando recursos de arquiteturas heterogêneas de diversos fabricantes.

No entanto, o gargalo de comunicação para transferência de dados entre a GPU e CPU levou ao projeto de Unidade de Processamento Acelerado da AMD (APU), que combina a CPU e GPU em um único chip, formando um processador de computação heterogênea onde toda comunicação ocorre através do mesmo barramento.

Essa mudança para um ambiente fortemente integrado implica diretamente na maneira de desenvolver os algoritmos, explorar seus recursos computacionais e compartilhamento de memória, fazendo com que estas arquiteturas tenham um novo nível de poder computacional para o público em massa que antes seria limitado a processadores *multi-core*.

## 3 Arquitetura desenvolvida

A arquitetura deste trabalho é composta por três processos principais e dois dispositivos de processamento (GPU e CPU). O desenvolvimento da arquitetura heterogênea tem como objetivo ilustrar o relacionamento lógico entre estes processos.

O primeiro processo é o Particionador, este processo recebe os dados de entrada, consulta Estimador de Desempenho e divide o trabalho entre os processadores. O Estimador de Desempenho recebe os parâmetros de entrada e estima o desempenho relativo dos processadores. O processo sincronizador organiza os dados processados para formar a saída.

Os detalhes arquitetura podem ser observados na Figura 1 e o funcionamento dos processos serão descritos a seguir.

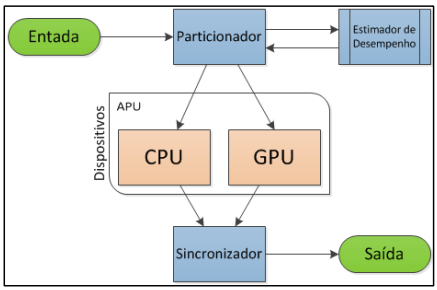



**Figura 1: Arquitetura desenvolvida**

O Particionador é o processo responsável por particionar o trabalho em sub-tarefas paralelas e distribuí-las aos processadores. Este processo foi desenvolvido baseando-se na

premissa de que há variações de desempenho entre os processadores e que tais variações podem ser previstas baseadas nos seus parâmetros.

Assim, este processo não irá atribuir todas as tarefas ao processador mais adequado, mas sim distribuir o trabalho entre todos os processadores, de forma que a diferença de tempo de execução prevista seja a mais próxima de zero. Dessa forma, é possível explorar a heterogeneidade da arquitetura para todas as aplicações, todavia, o processador que obter o melhor desempenho relativo irá executar uma carga de trabalho maior.

O Estimador de desempenho será responsável por automatizar a tarefa de previsão de desempenho relativo entre os processadores, baseando-se nos parâmetros da aplicação, que no contexto deste trabalho são o tamanho do problema e o ID da aplicação.

Este processo consiste em duas etapas. Na primeira, quando uma nova aplicação é implementada, são realizadas um conjunto de execuções da aplicação utilizando uma carga de trabalho igual para cada dispositivo (CPU e GPU). Na segunda etapa, os dados coletados na primeira etapa são utilizados como parâmetros para o cálculo do desempenho relativo entre os dispositivos para uma nova execução utilizando ambos os dispositivos cooperativamente.

O processo Sincronizador é responsável pela sincronização e por receber os dados processados pela CPU e GPU. Este processo foi desenvolvido partindo-se da premissa de que os processadores irão concluir sua carga de trabalho em tempos diferentes, por isso é necessário que haja uma operação bloqueante para que possa ocorrer esta sincronização.

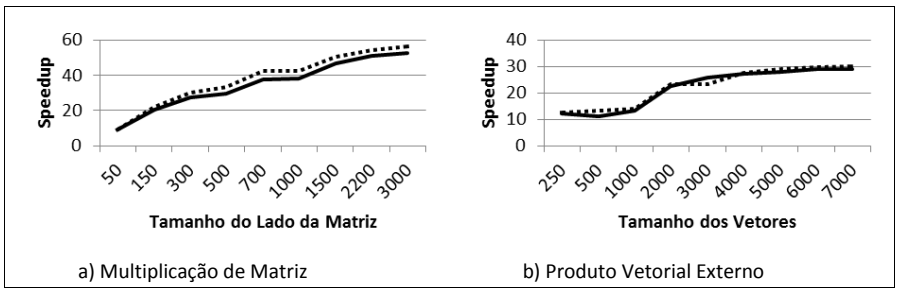
## 4 Resultados Experimentais

Para a execução dos experimentos foram utilizadas quatro aplicações paralelas disponíveis em versões para CPU e GPU em uma APU (*A8-3870K*) de primeira geração, utilizando a ferramenta de desenvolvimento OpenCL 1.2 em um ambiente local.

Considerando o nível de sincronização dos processadores, observa-se que a metodologia utilizada obteve melhores resultados que se diferenciam entre as aplicações avaliadas. De maneira geral, o Fator de Balanço de Carga médio se manteve em 0.09, o que representa um bom índice de sincronização entre os processadores. Portanto, com estes resultados pode-se afirmar que o algoritmo de particionamento obteve boa precisão ao estimar as cargas de trabalho para os processadores.

### 4.1 Desempenho da metodologia proposta

O objetivo desta abordagem é otimizar o tempo de processamento final, particularmente, melhorar o desempenho do melhor processador, utilizando todos os recursos possíveis do processador mais lento, sem que este influencie negativamente no tempo de execução final. Os resultados podem ser observados na Figura 2.




a) Multiplicação de Matriz b) Produto Vetorial Externo

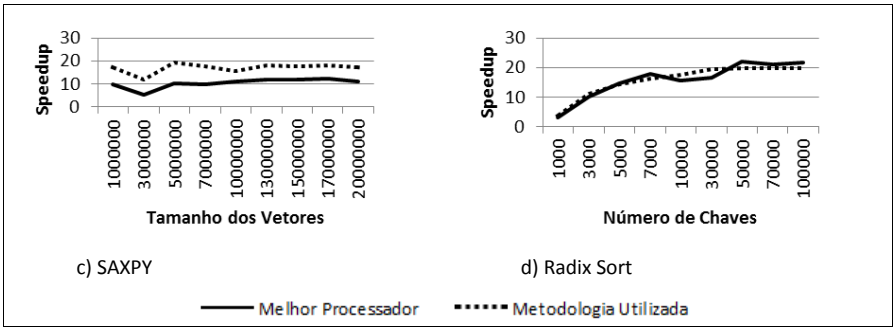



**Figura 2: Desempenho da metodologia utilizada**

Com estes resultados, observam-se ganhos que são inversamente proporcionais à diferença de desempenho relativo entre os processadores, ou seja, quanto menor a diferença no desempenho entre os processadores (CPU e GPU), maiores foram os ganhos sobre as técnicas mais estáticas.

Além disso, os ganhos de *speedup* foram diretamente dependentes da precisão da estimativa do desempenho relativo, ou seja, para situações onde o processador mais lento foi superestimado, os resultados apresentaram perdas significativas, que degradaram o desempenho final da aplicação. Entretanto, ao superestimar o processador mais rápido foram obtidos resultados que, no pior caso, se iguala ao desempenho exclusivo do melhor processador.

## 5 Conclusões

Neste trabalho foi estudado o problema de escalonamento de tarefas em ambientes heterogêneos utilizando CPUs e GPUs. O objetivo deste trabalho está diretamente relacionado com a investigação de uma abordagem de distribuição de tarefas, buscando reduzir o tempo ocioso dos processadores, especialmente o mais lento.

Os resultados obtidos demonstraram que a utilização conjunta dos processadores alcançou melhor desempenho em todos os casos onde o processador mais lento não foi superestimado. Todavia, o pior caso ocorreu quando o tempo máximo foi definido pelo processador mais lento. Esta situação implicou em pelo menos dois efeitos negativos: menores ganhos no desempenho final e ociosidade do melhor processador.

No entanto, para os melhores casos, foram obtidos resultados que superam as técnicas estáticas atuais. Estas situações foram favorecidas quando há pouca diferença entre o desempenho médio dos processadores. Esta situação afetou positivamente a eficiência da metodologia proposta, onde os resultados percentuais médios foram 20% mais eficientes e cerca de cinco vezes mais eficaz do que simplesmente particionar as tarefas sem algum critério.

## 6 Referências

Khronos Group. (2012). Acesso em 4 de Agosto de 2012, disponível em OpenCL: http://www.khronos.org/opencl

Lee, V., Kim, C., Chhugani, J., Deisher, M., Kim, D., Nguyen, A. D., et al. (2011). Debunking the 100X GPU vs. CPU Myth: An Evaluation of Throughput Computing on CPU and GPU. *SIGARCH Comput. Archit. News, 38*(3), 451-460.

Teodoro, G. L. (2010). *Computação em Fluxos de Dados para Ambientes Paralelos e Heterogêneos.* Belo Horizonte: Universidade Federal de Minas Gerais.