An Analytical Model for a GPU Architecture with Memory-level and Thread-level Parallelism Awareness

Resumo de Artigo: S. Hong and H. Kim,"An Analytical Model for a GPU Architecture with Memory-level and Thread-level Parallelism Awareness", ISCA 2009, pp. 152-163

Giovani Chiachia, RA 098362, 25/04/2010

A importância das arquiteturas GPU tem crescido muito na era *multicore* em que nos encontramos. No entanto, a programação massiva de *threads* paralelas continua sendo um grande desafio para os engenheiros de software. Ainda mais difícil que a paralelização dos programas é o entendimento dos gargalos de desempenho desses programas nas arquiteturas GPU. As abordagens atuais baseiam-se no refinamento do programa por parte dos programadores através da exploração de cenários distintos em que as características da arquitetura são exploradas sem que haja um entendimento pleno do que está sendo feito.

Com o objetivo de melhorar a percepção sobre como os gargalos de desempenho afetam as aplicações em arquiteturas GPU, este trabalho propõe um modelo analítico simples para a estimativa do tempo de execução de programas massivamente paralelos. Uma das características-chave do modelo está no fato dele fundamentar-se no número máximo de requisições em paralelo à memória (designado "*Memory Warp Paralelism*", MWP) que o programa possibilita. Esta medida está relacionada com o número de *threads* que acessam simultaneamente a memória, com o grau de paralelismo oferecido pela memória e com a largura do barramento de memória da GPU. A idéia é que o tempo de execução das aplicações é dominado pela latência das instruções de memória.

Baseado no grau de MWP, o modelo possibilita estimar o custo das requisições à memória, o que, por sua vez, possibilita estimar o tempo total de execução do programa. Adicionalmente, ao oferecer o número estimado de ciclos por instrução (CPI) de um programa, o modelo também possibilita aos programadores e compiladores decidir quando é vantajoso aplicar determinadas otimizações.

Para validar a proposta, comparações entre o desempenho real e o previsto pelo modelo são realizadas em uma série de GPUs e programas distintos. Os experimentos foram realizados a partir de programas "*micro-benchmarks*" e de algumas aplicações do "*Merge Benchmarks*", todos escritos na linguagem de programação CUDA. Para os programas "micro-benchmarks", o erro absoluto médio obtido foi de 5.4% e para os programas "Merge Benchmarks" selecionados, o erro foi de 13.3%.

Apesar de algumas deficiências, tais como não considerar o custo de "*caches misses*" ou de instruções de desvio, acredita-se que este modelo possa oferecer uma orientação aos programadores sobre como eles devem melhorar suas aplicações. Até onde se sabe, não há outro modelo analítico para predizer em tempo de compilação o desempenho dos programas em GPU baseado apenas em informações estáticas sobre seu comportamento.