

Escalonamento baseado em roubo de tarefas em arquiteturas paralelas híbridas

Vinícius Garcia Pinto¹, João Vicente Ferreira Lima¹,
Stéfano Drimon Kurz Mór¹, Nicolas Maillard¹

¹Grupo de Processamento Paralelo e Distribuído (GPPD)
Instituto de Informática - Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 - 91.501-970 - Porto Alegre - RS - Brasil
{vgpinto, jvlima, sdkmor, nicolas}@inf.ufrgs.br

1. Introdução

Nos últimos anos têm-se observado o advento de processadores *multicore* e de coprocessadores especializados como meios de aumentar o desempenho dos sistemas computacionais. Os sistemas paralelos até então baseados em aglomerados de processadores homogêneos passaram a serem constituídos de forma híbrida, utilizando processadores de propósito geral e coprocessadores especializados [Asanovic et al. 2009]. GPUs são exemplos de coprocessadores especializados usados para acelerar cálculos de tarefas paralelas com poucas dependências ainda que originalmente tenham sido projetadas somente para processamento gráfico.

A programação em arquiteturas híbridas tem sido feita por meio de ferramentas específicas para cada recurso. Em CPUs *multicore* têm sido utilizadas ferramentas como OpenMP e Intel TBB. Já em GPUs, para a programação de propósito geral, têm sido utilizados os ambientes CUDA e OpenCL. Atualmente, a integração dessas ferramentas específicas para cada recurso computacional deve ser feita manualmente pelo programador. Este precisa considerar também outros pontos como a distribuição de tarefas entre os recursos e as transferências de dados entre as memórias de forma a extrair o desempenho esperado da arquitetura híbrida.

Ambientes de programação como o StarPU [Augonnet et al. 2009] e o XKaapi [Hermann et al. 2010], que oferecem suporte a arquiteturas compostas por processadores *multicore* e GPUs, podem extrair proveito tanto do processador de uso geral quanto do coprocessador especializado. Nesse cenário, as tarefas a serem executadas possuem implementações específicas para cada recurso da arquitetura híbrida, podendo ser alocadas entre os diferentes recursos de processamento, de forma que cada um dos recursos execute aquelas tarefas para as quais é mais adequado [Shan 2006, Hermann et al. 2010].

Um dos desafios em ambientes híbridos é prover o aproveitamento eficiente dos recursos computacionais heterogêneos, escalonando as tarefas de forma a manter a carga balanceada. O roubo de tarefas (*work stealing*) é um algoritmo de escalonamento onde um recurso de processamento que está com a lista de tarefas a executar vazia escolhe um outro recurso de processamento da qual roubará tarefas e então rouba a última tarefa inserida na lista de execução desta unidade [Blumofe and Leiserson 1994].

2. Escalonamento em arquiteturas híbridas CPU-GPU

Em publicações recentes várias abordagens para escalonamento em arquiteturas CPU-GPU têm sido avaliadas, como em [Cederman and Tsigas 2008] onde são implementados e comparados quatro algoritmos para distribuição dinâmica de carga em GPU. Os algoritmos são baseados em: lista de tarefas estática, fila de tarefas dinâmica e

bloqueante, fila de tarefas dinâmica e não-bloqueante e roubo de tarefas. Em outra abordagem [Toss et al. 2011] é apresentada uma implementação de um escalonador baseado em roubo de tarefas em GPUs, a qual é comparada com implementações baseadas em algoritmos de escalonamento estático e lista de tarefas. Em outra proposta [Augonnet et al. 2010], o ambiente de execução StarPU [Augonnet et al. 2009] é estendido para suportar de forma eficiente múltiplas GPUs em uma mesma máquina e em *clusters* de GPUs. As modificações introduzidas para melhorar a eficiência são baseadas no gerenciamento assíncrono dos dados por meio de pré-busca dos dados e sobreposição entre comunicações e computação. Suporte a múltiplas GPUs em uma mesma máquina também está presente em [Chen et al. 2010]. Nessa abordagem de distribuição de carga de trabalho é utilizado um esquema de filas de tarefas multi-nível, onde são mantidas uma fila global de tarefas e n filas de tarefas para as GPUs, onde n é o número de dispositivos GPU presentes no sistema.

3. Objetivos e Trabalhos Futuros

Este trabalho tem como objetivo explorar o escalonamento baseado na técnica de roubo de tarefas em sistemas paralelos híbridos com suporte simultâneo a recursos de processamento heterogeneos como CPUs *multicore* e GPUs. Como etapas futuras planeja-se modelar e implementar o escalonador e validar sua aplicabilidade em um ambiente heterogêneo.

Referências

- Asanovic, K., Bodik, R., Demmel, J., Keaveny, T., Keutzer, K., Kubitowicz, J., Morgan, N., Patterson, D., Sen, K., Wawrzynek, J., Wessel, D., and Yelick, K. (2009). A view of the parallel computing landscape. *Commun. ACM*, 52:56–67.
- Augonnet, C., Clet-Ortega, J., Thibault, S., and Namyst, R. (2010). Data-Aware Task Scheduling on Multi-accelerator Based Platforms. In *Parallel and Distributed Systems (ICPADS), 2010 IEEE 16th International Conference on*, pages 291–298.
- Augonnet, C., Thibault, S., Namyst, R., and Wacrenier, P. A. (2009). StarPU: a unified platform for task scheduling on heterogeneous multicore architectures. *Euro-Par 2009 Parallel Processing*, pages 863–874.
- Blumofe, R. and Leiserson, C. (1994). Scheduling multithreaded computations by work stealing. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 356–368. IEEE Comput. Soc. Press.
- Cederman, D. and Tsigas, P. (2008). On Dynamic Load Balancing on Graphics Processors. *Technology*, pages 57–64.
- Chen, L., Villa, O., Krishnamoorthy, S., and Gao, G. R. (2010). Dynamic load balancing on single- and multi-GPU systems. *2010 IEEE International Symposium on Parallel Distributed Processing IPDPS*, pages 1–12.
- Hermann, E., Raffin, B., Faure, F., Gautier, T., and Allard, J. (2010). Multi-GPU and Multi-CPU Parallelization for Interactive Physics Simulations. *Euro-Par 2010 Parallel Processing*, pages 235–246.
- Shan, A. (2006). Heterogeneous Processing: a Strategy for Augmenting Moore’s Law. *Linux Journal*, 2006(142):7.
- Toss, J., Gautier, T., and Danjean, V. (2011). Work Stealing Inside GPU. Technical report, LIG - Laboratory of Informatics of Grenoble.