WORMS: Um Ambiente de Execução para Sistemas Multi-CPU e Multi-GPU

Vinícius Garcia Pinto, Nicolas Maillard

[1]Grupo de Processamento Paralelo e Distribuído - Instituto de Informática
Universidade Federal do Rio Grande do Sul (UFRGS) - Porto Alegre - RS - Brasil

{vgpinto, nicolas}@inf.ufrgs.br

Resumo. *Este artigo apresenta a avaliação de um ambiente de execução para sistemas de processamento de alto desempenho híbridos constituídos por CPUs e GPUs. Os resultados obtidos mostram que o ambiente avaliado teve melhor desempenho que ferramentas padrão para programação em CPUs e GPUs.*

1. Considerações Iniciais

Nos sistemas de alto desempenho atuais, uma das maneiras utilizadas para aumentar o poder de processamento tem sido o uso de arquiteturas híbridas. Frequentemente essas arquiteturas são compostas por processadores *multicore* de propósito geral e por aceleradores, como GPUs [Asanovic et al. 2009].

As ferramentas de programação clássicas para sistemas de alto desempenho não levam em conta as particularidades dos sistemas híbridos contemporâneos. Dessa forma, novas ferramentas de programação que sejam construídas considerando as particularidades de sistemas híbridos podem atingir melhores resultados tanto na produtividade e legibilidade do código quanto no desempenho atingido pela aplicação [Vandierendonck et al. 2011, Asanovic et al. 2009].

Neste trabalho será apresentado o ambiente de execução WORMS que oferece suporte para arquiteturas híbridas compostas por CPUs multicore e múltiplas GPUs. Posteriormente será apresentada uma avaliação experimental do desempenho do ambiente de execução em uma arquitetura composta por CPUs e GPUs.

2. Ambiente de Execução para Sistemas Multi-CPU e Multi-GPU

O ambiente de execução WORMS (WORk stealing scheduling for Multi-CPU/GPU Systems) fornece suporte a sistemas multi-CPU e multi-GPU utilizando o paradigma do paralelismo de tarefas. WORMS permite que uma mesma tarefa tenha implementação tanto para execução em CPUs *multicore* quanto em GPUs, decidindo em tempo de execução qual das implementações será executada de acordo com os recursos de hardware disponíveis. O escalonamento dessas tarefas é feito através de um algoritmo de roubo de tarefas. Além disso é permitido que uma tarefa em execução crie novas tarefas filhas dinamicamente, seguindo um modelo de dependências totalmente estritas (*fully strict*).

Em um trabalho prévio [Pinto and Maillard 2012], foi apresentada uma versão anterior do ambiente. Nessa versão, ao combinar a capacidade de processamento de CPUs e GPU o WORMS apresentou desempenho superior ao de ferramentas tradicionais tanto para programação em CPUs quanto para GPUs. Neste trabalho, será avaliada a versão atual do ambiente que incorpora modificações que reduzem o *overhead* no gerenciamento das tarefas, possibilitando a execução de aplicações que criem maior número de tarefas.

3. Avaliação Experimental e Considerações Finais

Os resultados aqui apresentados foram obtidos na plataforma Newton do Centro Nacional de Supercomputação da UFRGS utilizando dois processadores AMD Opteron 2427 2.2 GHz com seis núcleos cada e duas GPUs NVIDIA Tesla S1070. Como aplicação de teste foi utilizada uma multiplicação de matrizes por algoritmo de Strassen. Foram testadas três implementações: uma com o WORMS para execução em CPUs e GPUs, uma com Cilk para execução em CPUs e uma com CUBLAS (somente GPU). As execuções foram repetidas 50 vezes usando até 10 núcleos CPU e duas GPUs com matrizes de 8192 x 8192.

A Figura 1 apresenta o desempenho de pico obtido para a aplicação multiplicação de matrizes. Pode-se observar que o ambiente WORMS usando núcleos da CPU e as GPUs simultaneamente obteve desempenho superior tanto ao do Cilk quanto do CUBLAS. Além disso, na configuração utilizando somente CPUs o desempenho do WORMS foi equivalente ao da ferramenta Cilk.

Como trabalhos futuros, pretende-se implementar outras aplicações de teste e realizar testes com um número maior de GPUs, além de incrementar algumas funcionalidades do ambiente de execução.

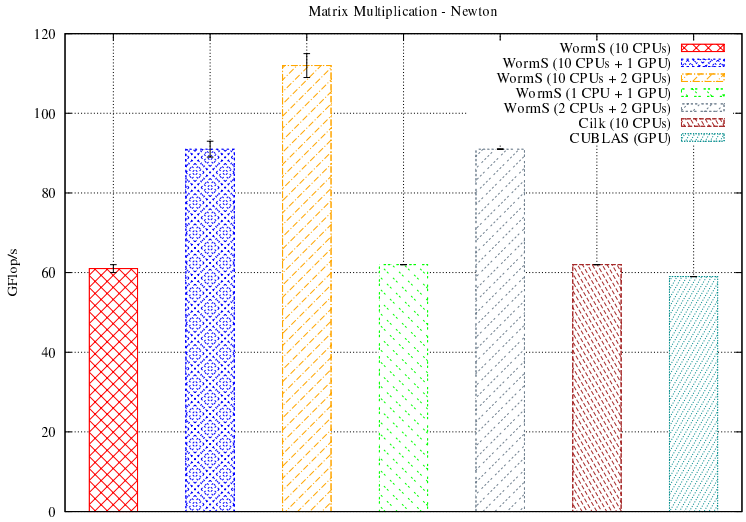

Figura 1. Desempenho de pico da aplicação Multiplicação de Matrizes no ambiente WORMS em comparação com Cilk e CUBLAS

Referências

Asanovic, K., Bodik, R., Demmel, J., Keaveny, T., Keutzer, K., Kubiatowicz, J., Morgan, N., Patterson, D., Sen, K., Wawrzynek, J., Wessel, D., and Yelick, K. (2009). A view of the parallel computing landscape. *Communications of the ACM*, 52(10):56–67.

Pinto, V. G. and Maillard, N. (2012). Work stealing on hybrid architectures. In *13th Symposium on Computer Systems (WSCAD-SSC 2012)*, pages 17 –24, Los Alamitos. IEEE Computer Society.

Vandierendonck, H., Pratikakis, P., and Nikolopoulos, D. S. (2011). Parallel programming of general-purpose programs using task-based programming models. In *Proceedings of HotPar'11*, Berkeley, CA, USA. USENIX Association.