Uso de GPUs para Simulações Hidrológicas de Águas Rasas em Pequenas Regiões

Adiel Seffrin Sarates Junior[1], **Philippe O. A. Navaux**[1]

[1]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

{assaratesj,navaux}@inf.ufrgs.br

Resumo. *Com o avanço das tecnologias heterogêneas, aplicações de simulações hidrológicas das mais variadas foram aceleradas. Com a introdução das GPU, metas de desempenho esperadas puderam ser alcançadas agilizando assim o resultado de eventos prováveis. Para simulações conhecidas, a introdução das GPUs possibilitou ganhos de 6x quando comparadas com versões paralelas em multi-cores.*

1. Introdução

A modelagem hidrológica tem sofrido grandes alterações tanto em suas equações governantes como nas tecnologias disponíveis. Nos últimos 40 anos, esta área evoluiu de um campo de pesquisa com poucos dados coletados, para um campo com uma vasta e abrangente fonte de recursos com a disponibilização de satélites para realizar o levantamento topográfico e a difusão facilitada desta informação.

A simulação voltada a pequenas áreas atende uma demanda para simulações específicas, como quebra de caixas d'água municipais ou rompimento de tubulações localizadas sobre a superfície para analisar seu impacto com alguma antecedência. Este trabalho visa mostrar que a utilização de placas gráficas para processamento genérico pode reduzir o tempo de simulação a um nivel satisfatório.

2. Trabalhos Relacionados

Desde os métodos de previsão de inidações propostos por [Zanobetti et al. 1970], a modelagem hidrológica se tornou cada vez mais popular. As mais recentes melhorias foram propostas por [Bates et al. 2010] onde uma nova proposta de resolução é apresentada. Com esta nova maneira, trocou-se o método de resolução das equações de um método difusivo para um método inercial [Hunter et al. 2008].

Além das melhorias do método de resolução, novas tecnologias passaram a ser usadas para acelerar tal processo. Recursos como máquinas com multi-processadores ou aceleradores externos passaram a ser estudados e, para dados tipos de modelos, grandes resultados puderam ser obtidos [Neal et al. 2010].

3. Processamento Heterogêneo

Com o desenvolvimento e utlização de novas tecnologias externas aos habituais processadores, o processamento heterogêneo ganhou mais espaço na área de simulações em geral. Neste trabalho foi utilizado a simulação publicada por [Bates et al. 2010], onde há uma entrada de um grande volume de água em um determinado ponto em uma rua

na região de Greensfiels em Glasgow, no Reino Unido. Nesta simulação é utilizada uma abrodagem puramente serial para reproduzir um evento com 2 horas de tempo simulado. Desta forma, foi obtido um resultado de 1,47 minutos e também é aprensetada uma previsão com uma expectativa de que com os recursos que estavam surgindo, seria possível executar tal simulação em 0,25 minutos utilizando técnicas paralelas com 8 cores para a execução utilizando a API OpenMP, dada a sua facilidade de implementação.

Com este tempo como meta, executamos o experimento paralelo utilizando OpenMP em uma máquina com 12 processadores, bem como uma versão apta a rodar em uma GPU foi desenvolvida, para que fosse possível comparar seus desempenhos neste tipo de simulação. Com a versão GPU implementada, foram analisados os impactos de utlização de GPUs para esta finalidade.

4. Resultados

Os teste utilizando OpenMP foram realizados em uma máquina Dell PowerEdge R720 com 2 processadores Xeon E5-2630 com 6 cores e 2 threads por core, enquanto a versão em GPU foi executada na mesma máquina juntamente com uma placa NVIDIA Telsa K20. Ambos casos são referentes à media de 20 execuções para cada versão.

Para a versão que utiliza OpenMP uma otimização de localidade de dados foi utilizada, de modo a facilitar o acesso à memória aos dados além de submetida a versões de 6, 12 e 24 threads. Para a versão que utiliza a GPU, testes mostraram que o acesso à memoria feito com o recurso de memória unificada, onde o gerenciamento dos dados é feito pela placa e tais dados estão alocados na região conhecida como *pinned memory* onde não é necessária realizar a paginação dos dados para o acesso, torna a comunicação entre CPU e GPU mais eficiente e assim melhorando o desempenho.

As versões executadas somente pela CPU tiveram seu melhor resultado quando executadas com todas as 24 threads resultando em um tempo de execução de 0,79 minutos ainda acima da expectativa relatada. Entretanto, a aplicação executada com o auxílio da placa GPU, obteve um tempo médio de 0,25 minutos. Com este resultado, o uso de GPUs para acelarar aplicações desse tipo deve ser analisado pela demanda maior de conhecimento técnico, quando comparado à utilização de OpenMP.

Referências

Bates, P. D., Horritt, M. S., and Fewtrell, T. J. (2010). A simple inertial formulation of the shallow water equations for efficient two-dimensional flood inundation modelling. *Journal of Hydrology*, 387(1):33–45.

Hunter, N., Bates, P., Neelz, S., Pender, G., Villanueva, I., Wright, N., Liang, D., Falconer, R., Lin, B., Waller, S., et al. (2008). Benchmarking 2d hydraulic models for urban flooding. *Proceedings of the ICE-Water Management*, 161(1):13–30.

Neal, J. C., Fewtrell, T. J., Bates, P. D., and Wright, N. G. (2010). A comparison of three parallelisation methods for 2d flood inundation models. *Environmental Modelling & Software*, 25(4):398–411.

Zanobetti, D., Lorgeré, H., Preissmann, A., and Cunge, J. A. (1970). Mekong delta mathematical model program construction. *Journal of the Waterways, Harbors and Coastal Engineering Division*, 96(2):181–199.