Modalidade do trabalho: Relatório técnico-científico
Evento: 2011 SIC - XIX Seminário de Iniciação Científica

EXECUÇÃO PARALELA DE SIMULAÇÕES DE SISTEMAS DINÂMICOS ATRAVÉS DA UTILIZAÇÃO DE GPU[1]

Marcos Batista Ketzer[2], Paulo Sérgio Sausen[3].

[1] Projeto de pesquisa desenvolvido no Departamento de Ciências Exatas e Engenharias, pertencente ao Grupo de Automação Industrial e Controle
[2] Estudante do Curso de Engenharia Elétrica do Departamento de Ciências Exatas e Engenharias, integrante do Grupo de Automação Industrial e Controle. E-mail: marcos.ketzer@unijui.edu.br
[3] Professor do Departamento de Ciências Exatas e Engenharias, participante do Grupo de Automação Industrial e Controle. E-mail: sausen@unijui.edu.br

Resumo

O estudo realizado parte da necessidade no desenvolvimento de aplicações que exploram a capacidade computacional disponibilizada pelas unidades de processamento gráfico (GPU). Tal recurso tem atratividade devido à arquitetura de processamento paralelo do dispositivo que permite que cálculos com grande quantidade de iterações sejam processados de forma distribuída, diminuindo assim, o tempo de execução. Através da tecnologia CUDA®, desenvolvida e disponibilizada pela NVIDIA®, é possível ter acessos a tais recursos de forma efetiva para o programador. Sendo assim, um problema derivado de varreduras paramétricas em simulação de sistemas dinâmicos foi implementado utilizando a tecnologia CUDA e o software MatLab®, mostrando sua viabilidade técnica no projeto e execução. Os resultados confirmam a eficiência da GPU frente à CPU, diminuindo os tempos de processamento para os diversos casos.

Palavras chave: Graphics Processing Unity, CUDA, NVIDIA.

Introdução

Os problemas de computação científica frequentemente são limitados devido ao tempo de processamento necessário para obtenção de resultados com precisão satisfatória. Neste contexto, diversas otimizações em compiladores e bibliotecas de algoritmo matemático, têm como objetivo tornar o processamento mais eficiente. Como solução alternativa a este problema, é utilizada computação paralela e distribuída, que permite que múltiplos núcleos de processamento, internos ou não a uma mesma unidade computacional, auxiliem na execução.

Por outro lado, estimulados pelo desenvolvimento de processamento gráfico, as unidades GPU tem constantes progressos tecnológicos significativos. Os modelos atuais de GPUs apresentam um elevado grau de paralelismo, uma maior largura de banda de memória e um alto poder computacional para cálculos aritméticos, superando assim, os processadores modernos em operações matemáticas intensas. Tendo em vista esse enorme poder computacional, as GPUs passaram a ser utilizadas em computação de propósito geral (GPGPU). Ainda, estas têm sido preferíveis ao processamento em clusters devido aos custos associados, que são muito inferiores.

Modalidade do trabalho: Relatório técnico-científico
Evento: 2011 SIC - XIX Seminário de Iniciação Científica

Lançada em novembro de 2006, a tecnologia CUDA (Compute Unified Device Architecture) é a arquitetura de hardware e software desenvolvida pela empresa NVIDIA que permite utilizar e gerenciar as capacidades computacionais paralelas das GPUs (NVIDIA, 2010). Usando CUDA, as últimas GPUs da NVIDIA se tornam acessíveis semelhante a programação em CPU.A programação em CUDA é originalmente em C, apesar de já existirem alguns softwares com capacidade de integração deste tecnologia, numa linguagem de mais alto nível.

Neste contexto, este trabalho consiste na exploração do poder computacional das unidades de processamento gráfico da NVIDIA, através do uso da tecnologia CUDA e do software MatLab. Neste é verificada a eficiência do processamento em GPU para um conjunto de simulação com objetivo da análise de varredura paramétrica. Neste caso é utilizada uma planta do sistema de distribuição disponibilizada pelo DEMEI. Os resultados obtidos em cada caso são comparados com sua execução e CPU, de forma a validar sua eficiência.

Metodologia

As GPUs foram originalmente desenvolvidas com a idéia de executar todo tipo de operação gráfica. Normalmente, na execução de aplicações gráficas, uma mesma operação é executada de forma repetitiva com dados diferentes, fazendo com que a arquitetura das GPUs fossem criadas para operar grandes conjuntos de instruções de maneira paralela. Desta forma, as GPUs conseguem manter altas frequências de computação em execuções paralelas de instruções (REIS *et al*, 2007).

A tecnologia CUDA permitiu que uma programação genérica, diminuindo o *overhead* encontrado na programação com os APIs gráficos. Para alcançar o maior número de desenvolvedores possíveis nesta plataforma, a NVIDIA utilizou o padrão C, adicionando poucas extensões da arquitetura CUDA. Segundo (SENDERS *et al*, 2010), o CUDA C, como é conhecido, se tornou a primeira linguagem especificamente desenvolvida por uma companhia de GPUs para facilitar uma computação não somente gráfica, mas de propósito geral.

A plataforma CUDA introduz dois novos conceitos para o escalonamento das *threads*: blocos e *grids*. É com esses conceitos que se organiza a repartição dos dados entre as *threads*, bem como sua organização e distribuição ao hardware. Cada *kernel* carregado no dispositivo, possui uma *grid* que contém encapsulados os blocos e *threads*. A Figura 1 apresenta as divisões dos processos na arquitetura CUDA.

Para utilizar CUDA de maneira a obter o melhor desempenho, programadores devem escolher a melhor maneira de dividir os processos de forma a manter a maior parte da GPU ocupada. Os fatores para análise incluem o tamanho do conjunto global de dados; a máxima quantidade de dados locais que um bloco de threads pode compartilhar; o número de multiprocessadores da GPU e o tamanho das memórias (DETOMINI, 2010).

O suporte da utilização da tecnologia CUDA pelo MatLab da MathWorks somente foi incorporada na segunda versão de 2010. A partir desta ferramenta é possível interagir com a plataforma CUDA usufruindo das bibliotecas de computação científica contidas no MatLab, agilizando o tempo de desenvolvimento da aplicação.

Modalidade do trabalho: Relatório técnico-científico
Evento: 2011 SIC - XIX Seminário de Iniciação Científica

Neste trabalho, o *software* MatLab em conjunto com a tecnologia CUDA são utilizados para implementar um algoritmo de simulação paralela de um sistema de distribuição de energia elétrica. O sistema utilizado esta localizado em Ijuí-RS Ijuí – RS Brasil, cuja concessão e operação são do Departamento Municipal de Energia de Ijuí (DEMEI). Maiores detalhes do sistema e sua modelagem são encontrados em (KETZER *et al*, 2010).

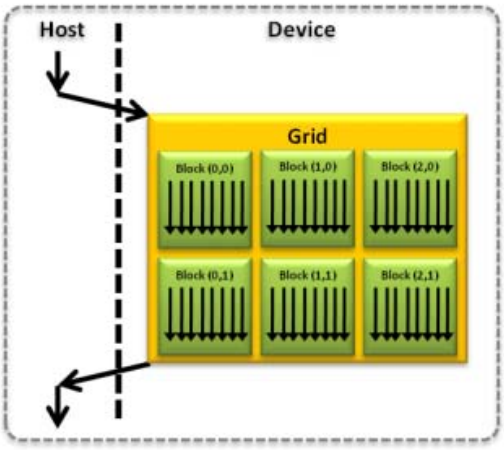

Figura 1. Hierarquia de processos em CUDA.

O processo paralelizado neste trabalho realiza a varredura de parâmetros para uma análise de sobreposição de frequências na rede, gerando a necessidade de múltiplas simulações. Esta análise permite verificar o comportamento do circuito de potência com sobreposição de harmônicos de tensão com freqüências acima da fundamental, cujo há interesses na área de detecção de falhas e comunicação em *Power Line Comunication (PLC).*

A simulação do sistema é realizada com sua representação em espaço de estados discreta. Devido a dimensões das matrizes, os cálculos em cada iteração podem ser facilmente paralelizados. Ainda, devido a não existência de dependência de dados entre cada simulação, o processo é facilmente paralelizado. A Figura 2 mostra a síntese do algoritmo implementado.

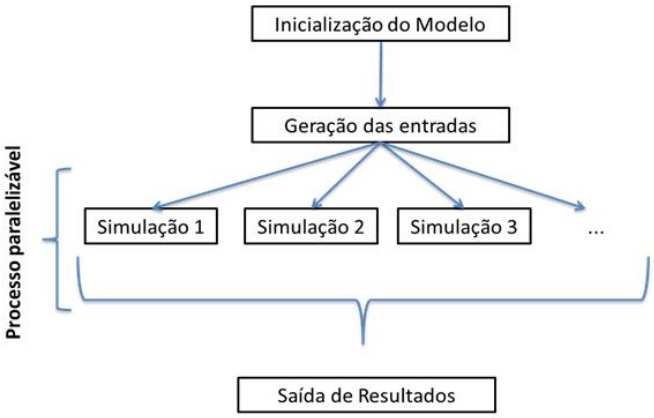

Figura 2. Algoritmo Paralelizado.

Resultados e Discussão

Para o desenvolvimento do código foi utilizado um hardware com as configurações contidas na tabela 1. O sistema operacional onde foram realizados os testes foi o Windows 7 - 32bits. Foi utilizada a versão MatLab 2010b, e o Cuda Toolkit 4.0.

Tabela 1. Especificações do computador utilizado.

	CPU	GPU
Processador	Intel Core i3 - 540	NVIDIA GeForce GTX 460
Frequência de Clock	3.06Ghz	625 MHz
Memória	3 GB DDR3	1.024 MB GDDR5
Largura de Banda Máxima	21 GB/s DD3	24.7 GB/s DDR3
Velocidade de Memória	667 MHz	800 MHz
Número de Núcleos	2 (4 threads)	48
Largura do Bus	64 bits	256 bits

Foram realizadas simulações com número diferente de varreduras paramétricas, o que significa diferença no nível de paralelismo no processamento. Os resultados obtidos são apresentados na Figura 3. Nesta é apresentado os tempos de processamento no CPU e na GPU obtidos a partir de uma média de 5 execuções cada. Os termos 4X, 8X, 12X, 16X e 24X se referem ao número de variações, ou simulações, realizadas em paralelo.

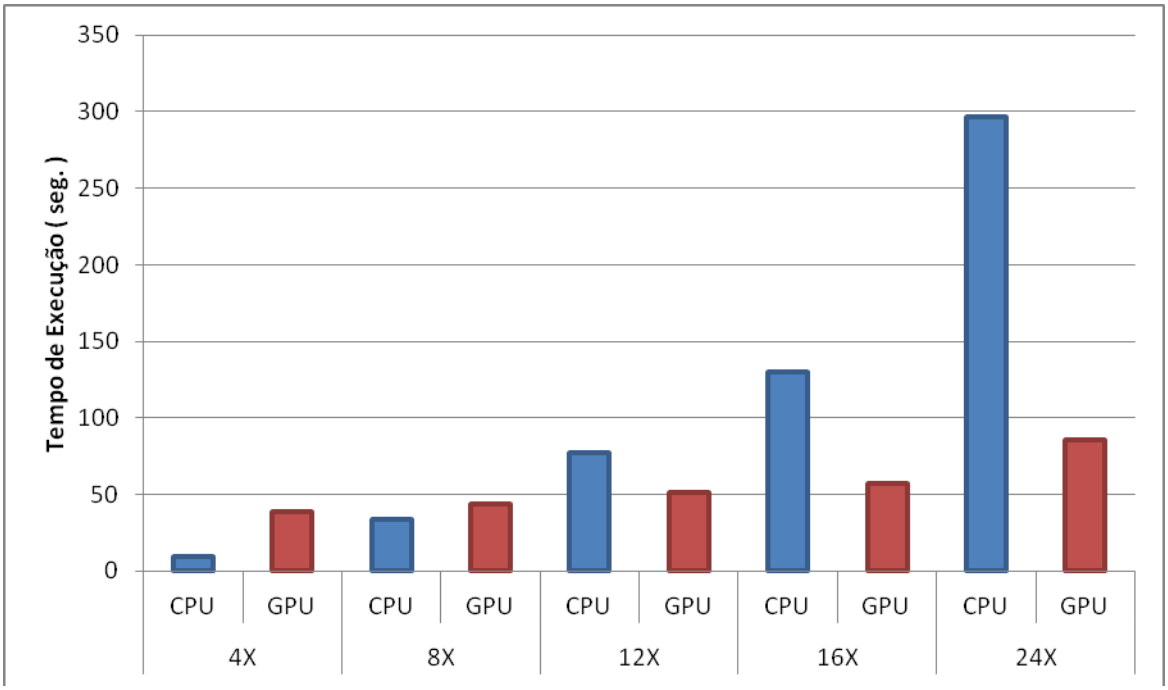

Figura 3. Subsistema de Monitoramento Móvel.

Pelos resultados é verificado que quando o processo é pouco paralelizável, o desempenho da GPU pode ser até mesmo inferior ao da CPU. Entretanto, quando o problema

paralelizado é maior, maior também é a vantagem da utilização da GPU. Pode ser visto que o tempo de processo na GPU é mais constante, enquanto que a CPU tem o tempo significativamente maior com o número de simulações. Para o caso mais extremo, tem-se que o tempo de processamento da GPU é quase quatro vezes o tempo em CPU. Sendo assim, para um caso em específico, deve ser analisada a parcela paralelizável do problema, para verificar se é ou não vantajosa o desenvolvimento do processo em GPU. Quando a parcela paralelizável, assim como a quantia de dados, é grande, a GPU será vantajosa.

Conclusões

Neste trabalho apresentou-se uma análise de desempenho da arquitetura na paralelização de simulações de sistemas dinâmicos com objetivo de varredura paramétrica, tomando como caso o sistema de distribuição de energia de Ijuí. Em ambos os casos, foi verificado que o tempo de processamento, principalmente quando as ordens das matrizes nos cálculos são elevadas, é menor na GPU, sendo vantajosa a aplicação desta nos problemas de computação científica.

Entretanto, a programação em CUDA requer uma série de conceitos novos não existentes na programação em CPU, tais como a complexa hierarquia de memória, sendo responsabilidade do programador, identificar e dividir as tarefas de maneira a melhor ocupar os recursos da GPU. Ainda, o programador deve tentar minimizar as iterações da CPU. Neste contexto, a utilização da plataforma MatLab é poderosa pois acelera o tempo de desenvolvimento dos programas paralelizados.

Agradecimentos

Os autores agradecem a Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul (FAPERGS) pela bolsa de Iniciação Científica disponibilizada para este Projeto de Pesquisa.

Referências

DETOMINI, Renan. C. **Exploração de Paralelismo em Criptografia Utilizando GPUs**. Monografia de Conclusão de Curso, Universidade Estadual Paulista, 2010.

KETZER, Marcos, CAMPOS, Maurício. (2010). **Caracterização e Modelagem de um Sistema de Distribuição de Energia Elétrico Realístico.** XVI SIC,UNIJUÍ, Ijuí.

REIS, David, CONTI, Ivan, VENETILLO, Jeronimo. (2007). **GPU - Graphics Processor Units**. Disponível em: <http://www.verlab.dcc.ufmg.br/_media/cursos/arquitetura/2007-1/grupo3/seminario_grupo3.pdf>, acessado em 15/08/2011.

SANDERS, Jason, KANDROT, Edward. **Cuda by Example**.US: NVIDIA, 2010. 311 p.