



Cleomar Pereira da Silva

Computação de Alto Desempenho com Placas Gráficas para Acelerar o Processamento da Teoria do Funcional da Densidade

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Engenharia Elétrica do Departamento de Engenharia Elétrica da PUC-Rio.

Orientador: Prof. Marco Aurélio Cavalcanti Pacheco

Rio de Janeiro

Abril de 2010



Cleomar Pereira da Silva

Computação de Alto Desempenho com Placas Gráficas para Acelerar o Processamento da Teoria do Funcional da Densidade

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-Graduação em Engenharia Elétrica do Departamento de Engenharia Elétrica do Centro Técnico Científico da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Prof. Marco Aurélio Cavalcanti Pacheco
Orientador

Departamento de Engenharia Elétrica – PUC-Rio

Prof. Omar Paranaina Vilela Neto

Departamento de Engenharia Elétrica – PUC-Rio

Profa. Cristiana Bentes
UERJ

Prof. Ricardo Cordeiro de Farias
UFRJ

Prof. José Eugenio Leal
Coordenador Setorial do Centro
Técnico Científico – PUC-Rio

Rio de Janeiro, 13 de abril de 2010

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

Cleomar Pereira da Silva

Graduado Engenheiro Eletricista pela Universidade Federal de Santa Maria (UFSM) em 2008. Mestrado em Engenharia Elétrica, área de concentração Nanotecnologia, pela Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio) em 2010.

Ficha Catalográfica

Silva, Cleomar Pereira da

Computação de alto desempenho com placas gráficas para acelerar o processamento da teoria do funcional da densidade / Cleomar Pereira da Silva ; orientador: Marco Aurélio Cavalcanti Pacheco. – 2010.

87 f. ; 30 cm

Dissertação (mestrado)–Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica, 2010.

Inclui bibliografia

1. Engenharia elétrica – Teses. 2. Teoria do funcional de densidade. 3. SIESTA. 4. Computação de alto desempenho. 5. GPGPU. 6. CUDA. I. Pacheco, Marco Aurélio Cavalcanti. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Elétrica. III. Título.

CDD: 621.3

Aos Meus Pais Antonio Coraci e Maria Rosa.

Agradecimentos

À CAPES e à PUC-Rio pela concessão da bolsa e dos auxílios que tornaram possível a realização deste trabalho.

Ao meu orientador, Dr. Marco Aurélio Cavalcanti Pacheco, pelo imediato atendimento das solicitações realizadas durante o período do estudo.

À professora Dra. Marley M. B. R. Vellasco pelos conhecimentos transferidos através das disciplinas que ministrou.

Aos demais professores pelos ensinamentos.

Ao Dr. Omar Paranaíba V. Neto pela participação na elaboração das idéias envolvidas com o tema desta dissertação.

Aos amigos Douglas M. Dias, Leandro F. Cupertino, Iury S. O. Bezerra, Daniel S. Chevitarese, Dr. Juan L. Lazo e Dr. Renato B. Oliveira, pelo apoio e auxílio prestados no desenvolvimento deste trabalho.

Aos meus pais, pela educação, atenção e carinho.

Aos amigos do ICA por seu contínuo apoio e colaboração.

Resumo

Silva, Cleomar Pereira; Pacheco, Marco Aurélio Cavalcanti (Orientador). **Computação de Alto Desempenho com Placas Gráficas para Acelerar o Processamento da Teoria do Funcional da Densidade**. Rio de Janeiro, 2010. 87p. Dissertação de Mestrado - Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

As Unidades de Processamento Gráfico (GPUs), ou Placas Gráficas, são processadores que foram originalmente projetados para executar tarefas dedicadas às operações da computação gráfica. Porém, a NVIDIA desenvolveu uma extensão da linguagem C para programação de GPUs, chamada CUDA (*Compute Unified Device Architecture*). Isto permitiu utilizá-las, na Computação de Alto Desempenho, para processar dados genéricos. Já os sistemas físicos estudados pela Mecânica Quântica apresentam dimensões próximas da escala atômica, tais como moléculas, átomos, prótons e elétrons. A Teoria do Funcional da Densidade (DFT) é um dos métodos iterativos mais usados para encontrar uma solução aproximada para a equação de Schrödinger. Contudo, os cálculos realizados em DFT são computacionalmente intensos devido às integrais de troca e correlação eletrônica, integrais para o cálculo da energia de Hartree e energia cinética dos elétrons, as quais requerem maior esforço computacional à medida que o número de elétrons presentes na simulação aumenta. Esta pesquisa teve como objetivo estudar os cálculos do DFT e identificar partes do algoritmo que, se alteradas, apresentassem benefícios de desempenho ao serem executadas em GPU. Assim, funções computacionalmente intensas do método DFT do SIESTA (*Spanish Initiative for Electronic Simulations with Thousands of Atoms*) foram paralelizadas e usadas para calcular propriedades físicas de nanotubos e fulerenos. Verificou-se que a execução da versão paralela do SIESTA para GPU é capaz de atingir ganhos em desempenho, em funções individuais, de uma ou até duas ordens de grandeza, tornando promissor o emprego de GPUs em acelerar o processamento da Teoria do Funcional da Densidade.

Palavras-chave

Teoria do Funcional da Densidade; SIESTA; Computação de Alto Desempenho; GPGPU; CUDA.

Abstract

Silva, Cleomar Pereira; Pacheco, Marco Aurélio Cavalcanti (Advisor). **High Performance Computing with Graphics Cards to Accelerate Processing Density Functional Theory**. Rio de Janeiro, 2010. 87p. MSc. Dissertation - Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

The graphics processing units (GPUs), or graphics cards, are processors that were originally designed to perform dedicated tasks to the computer graphics operations. However, NVIDIA has developed an extension of the C language for programming GPUs, called CUDA (Compute Unified Device Architecture). This allowed the use of GPUs, in High Performance Computing, for processing generic data. The physical systems studied by quantum mechanics have dimensions close to atomic scale, such as molecules, atoms, protons and electrons. The Density Functional Theory (DFT) is one of the most used interactive methods to find an approximated solution to the Schrödinger equation. However, the calculations in DFT are computationally intensive because of the exchange and correlation electronic integrals, integrals to calculate the Hartree energy and electrons kinetic energy, which requires greater computational effort as the number of electrons present in the simulation increases. This research aimed to study the DFT calculations and identify parts of the algorithm that, if changed, experience performance benefits from execution in GPU. Thus, computationally intensive DFT functions of the SIESTA method (Spanish Initiative for Electronic Simulations with Thousands of Atoms) were parallelized and used to calculate the physical properties of nanotubes and fullerenes. It was found that the implementation of SIESTA parallel version on the GPU is able to achieve gains in performance, in individual functions, of one or even two orders of magnitude, making it promising employment of GPUs to speed up the processing of Density Functional Theory.

Keywords

Density Functional Theory; SIESTA; High Performance Computing; GPGPU; CUDA.

Sumário

1	Introdução	14
1.1.	Motivação	16
1.2.	Objetivos	16
1.3.	Descrição do Trabalho	16
1.4.	Estrutura da Dissertação	18
2	Teoria do Funcional da Densidade	19
2.1.	Componentes da Energia Total	20
2.2.	Aproximação de Born-Oppenheimer	21
2.3.	Teoremas de Hohenberg e Kohn	23
2.4.	Equações de Kohn-Sham	24
2.5.	Aproximações dos Potenciais de Troca e Correlação	25
2.5.1.	Aproximação da Densidade Local	26
2.5.2.	Aproximação do Gradiente Generalizado	26
2.6.	Pseudopotencial	27
2.7.	Hamiltoniano Empregado no SIESTA	29
2.7.1.	Funções de Base	30
2.7.2.	Orbitais Atômicos Numéricos	31
3	Computação de Propósito Geral em Unidades de Processamento	
	Gráfico	32
3.1.	Programação Paralela	32
3.1.1.	Taxonomia de Flynn	33
3.1.2.	Lei de Amdahl	35
3.2.	<i>Compute Unified Device Architecture (CUDA)</i>	36
3.3.	Elementos de Hardware da GPU	39
3.3.1.	Processador e Multiprocessador de Fluxo	39
3.3.2.	Tipos de Memória da GPU	40
3.4.	Elementos de Software da GPU	41
3.4.1.	<i>Kernel</i>	42
3.4.2.	<i>Grid</i>	43

3.4.3. Bloco de <i>Threads</i>	44
3.4.4. <i>Warp</i>	44
3.4.5. Escalabilidade	45
3.5. Técnicas Otimizadas	46
3.5.1. Acesso à Memória Global	46
3.5.2. Acesso à Memória Compartilhada	51
3.5.3. Transferência de Dados Através do Barramento PCI Express	53
3.5.4. Nível de Ocupação do Multiprocessador	54
4 Descrição das Alterações para Aceleração do SIESTA por GPU	56
4.1. Identificação das Partes Adequadas ao Paralelismo de Dados	57
4.1.1. CUFFT	57
4.1.2. CUBLAS	58
4.1.3. MAGMA	58
4.1.4. CULA	59
4.2. Alteração de Partes do SIESTA	59
4.2.1. Potencial de Hartree, Energia de Hartree e Forças de Deslocamento Atômicas	60
4.2.2. Implementação Paralela para GPUs	61
4.2.3. Cálculo do Dipolo Elétrico	68
4.2.4. Reordenação de Dados	71
5 Estudo de Casos	73
5.1. Caso 1 – Cálculo da Energia e Estrutura de Bandas de Nanotubos	73
5.1.1. Estrutura do Nanotubo	74
5.1.2. Resultados e Testes de Desempenho	75
5.2. Caso 2 – Otimização da Geometria Estrutural de Fullereno	78
5.2.1. Estrutura do Fullereno	78
5.2.2. Resultados e Testes de Desempenho	79
6 Conclusões e Trabalhos Futuros	81
7 Referências Bibliográficas	84

Lista de Figuras

Figura 1 – Função de onda de todos os elétrons normalizada (AE) e a pseudofunção de onda de valência normalizada (PS).	28
Figura 2 – Arquitetura SISD.	34
Figura 3 – Arquitetura SIMD.	34
Figura 4 – Arquitetura MISD.	35
Figura 5 – Arquitetura MIMD.	35
Figura 6 – Efeito da Lei de Amdahl sobre o ganho de velocidade de algoritmo rodando em múltiplos processadores.	36
Figura 7 – Operações de Ponto Flutuante por Segundo (NVIDIA, 2009b).	37
Figura 8 – Largura de Banda (NVIDIA, 2009b).	38
Figura 9 – A GPU dedica mais transistores para o processamento de dados (NVIDIA, 2009b).	38
Figura 10 – Tipos de memória das GPUs da NVIDIA (NVIDIA, 2009b).	41
Figura 11 – Fluxo de processamento de um software empregando GPGPU.	42
Figura 12 – <i>Grid</i> e Blocos de <i>Threads</i> (NVIDIA, 2009b).	43
Figura 13 – Modelo de Escalabilidade da GPU (NVIDIA, 2009b).	46
Figura 14 – Segmentos Alinhados de Memória e <i>threads</i> de um <i>half-warp</i> .	47
Figura 15 – As <i>threads</i> do <i>half-warp</i> acessam elementos da memória global dentro de um mesmo segmento alinhado de 16 palavras (NVIDIA, 2009a).	47
Figura 16 – Elementos não alinhados com o bloco de 16 palavras residem dentro do mesmo bloco de 128 bytes (NVIDIA, 2009a).	48
Figura 17 – Elementos não alinhados com o bloco de 16 palavras residem dentro de dois blocos diferentes de 128 bytes (NVIDIA, 2009a).	48
Figura 18 – Largura de banda efetiva para padrões de acesso à memória com deslocamento (NVIDIA, 2009a).	49
Figura 19 – Acesso a elementos da memória com intervalos entre os elementos acessados (NVIDIA, 2009a).	49
Figura 20 – Desempenho da largura de banda para padrões de acesso com intervalos entre os elementos lidos (NVIDIA, 2009a).	50

Figura 21 – Leitura, com deslocamentos, na memória global e na memória de textura (NVIDIA, 2009a).	50
Figura 22 – Padrões de acesso à memória compartilhada sem conflitos de banco (NVIDIA, 2009b).	52
Figura 23 – Padrões de acesso à memória compartilhada com conflitos de banco (NVIDIA, 2009b).	53
Figura 24 – Redução do tempo total através da sobreposição dos tempos de transferência e de execução (NVIDIA, 2009a).	54
Figura 25 – Interface entre Fortran e CUDA.	59
Figura 26 – As equações diferenciais são transformadas em equações algébricas.	60
Figura 27 – Somatório paralelo realizado na memória compartilhada da GPU para um único vetor de dados.	61
Figura 28 – Solução no domínio da frequência.	62
Figura 29 – Uma <i>thread</i> para cada elemento da matriz de densidades.	63
Figura 30 – Nível de Ocupação do Multiprocessador para primeira versão do <i>kernel</i> da Equação de Poisson.	64
Figura 31 – Uma <i>thread</i> processa um vetor de dados da matriz de densidades	64
Figura 32 – Nível de Ocupação do Multiprocessador para segunda versão do <i>kernel</i> da Equação de Poisson.	65
Figura 33 – Nível de Ocupação do Multiprocessador para o <i>kernel</i> dos somatórios da energia de Hartree e das forças de deslocamento atômicas.	66
Figura 34 – Somatório paralelo de sete vetores na memória compartilhada da GPU.	67
Figura 35 – Padrões usados de acesso à memória compartilhada, sem conflitos de banco.	68
Figura 36 – Nível de Ocupação do Multiprocessador para o <i>kernel</i> de Cálculo do Dipolo Elétrico.	69
Figura 37 – Somatório paralelo de três vetores na memória compartilhada da GPU.	70
Figura 38 – Nível de Ocupação do Multiprocessador para o <i>kernel</i> da Reordenação de Dados.	72

Figura 39 – Estrutura de bandas de nanotubos (a) <i>armchair</i> (5, 5), (b) zigzag (8, 0) e (c) zigzag (12, 0). O nível de Fermi está deslocado para o zero, indicado pela linha tracejada (Silva, 2008).	74
Figura 40 – Nanotubo (4,2). a) Vista Frontal, b) Vista Lateral.	75
Figura 41 – Estrutura de bandas para o nanotubo (4,2). O nível de Fermi está deslocado para o zero, indicado pela linha tracejada.	78
Figura 42 – Fulereo C60.	79

Lista de Tabelas

Tabela 1 – Unidades Atômicas	21
Tabela 2 – Funções da Biblioteca CUFFT adequadas para emprego no SIESTA	58
Tabela 3 – Funções da Biblioteca CUBLAS com possibilidade de emprego para o SIESTA.	58
Tabela 4 – Funções da Biblioteca MAGMA com possibilidade de emprego para o SIESTA	58
Tabela 5 – Funções da Biblioteca CULA utilizadas no meio científico	59
Tabela 6 – Operações de adição com <i>warp</i> incompleto por bloco, replicando sete vezes o padrão ilustrado na Figura 27.	66
Tabela 7 – Operações de adição com <i>warp</i> incompleto por bloco, com o padrão ilustrado na Figura 34.	67
Tabela 8 – Operações de adição com <i>warp</i> incompleto por bloco, replicando três vezes o padrão ilustrado na Figura 27.	69
Tabela 9 – Operações de adição com <i>warp</i> incompleto por bloco, com o padrão ilustrado na Figura 37.	70
Tabela 10 – Resultados de Aceleração com a primeira versão do <i>kernel</i> da Equação de Poisson, para o nanotubo (4,2).	76
Tabela 11 – Resultados de Aceleração com a segunda versão do <i>kernel</i> da Equação de Poisson, para o nanotubo (4,2).	76
Tabela 12 – Resultados de Aceleração Reduzindo-se a Transferência de Dados através do Barramento PCI Express, para o nanotubo (4,2).	77
Tabela 13 – Resultados de Aceleração com a segunda versão do <i>kernel</i> da Equação de Poisson, para o fulereno C60.	80
Tabela 14 – Resultados de Aceleração Reduzindo-se a Transferência de Dados através do Barramento PCI Express, para o fulereno C60.	80