

Paulo Rogério da Motta Junior

**Uma Abstração para Programação
Paralela**

Suporte para o Desenvolvimento de
Aplicações

TESE DE DOUTORADO

DEPARTAMENTO DE INFORMÁTICA
Programa de Pós-graduação em Informática

Rio de Janeiro
Agosto de 2011



Paulo Rogério da Motta Junior

Uma Abstração para Programação Paralela

Suporte para o Desenvolvimento de Aplicações

Tese de Doutorado

Tese apresentada ao Programa de Pós-graduação em Informática do Departamento de Informática do Centro Técnico Científico da PUC-Rio como requisito parcial para obtenção do grau de Doutor em Informática

Orientadora: Prof. Noemi Rodriguez

Rio de Janeiro
Agosto de 2011



Paulo Rogério da Motta Junior

Uma Abstração para Programação Paralela

Suporte para o Desenvolvimento de Aplicações

Tese apresentada como requisito parcial para a obtenção do grau de Doutor pelo Programa de Pós-Graduação em Informática do Centro Técnico Científico da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Prof. Noemi Rodriguez

Orientadora

Departamento de Informática — PUC-Rio

Prof. Carlos José Pereira de Lucena

Departamento de Informática – PUC-Rio

Prof. Renato Fontoura de Gusmão Cerqueira

Departamento de Informática – PUC-Rio

Prof. Alexandre Sztajnberg

Departamento de Informática e Ciência da Computação – UERJ

Prof. Simone de Lima Martins

Instituto de Computação - UFF

Prof. José Eugenio Leal

Coordenador Setorial do Centro Técnico Científico — PUC-Rio

Rio de Janeiro, 11 de Agosto de 2011

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

Paulo Rogério da Motta Junior

Graduou-se em Bacharel em Informática e Tecnologia da Informação na UERJ e fez o Mestrado em Ciência da Computação na UFF. Atuou em projetos de desenvolvimento de sistemas de médio e grande porte na EDS, Sakonnet e atualmente presta consultoria para o Exército Brasileiro. Seus interesses são paralelismo, distribuição, engenharia de software e linguagens de programação.

Ficha Catalográfica

Motta Junior, Paulo Rogério da

Uma Abstração para Programação Paralela : Suporte para o Desenvolvimento de Aplicações / Paulo Rogério da Motta Junior; orientadora: Noemi Rodriguez. — Rio de Janeiro : PUC-Rio, Departamento de Informática, 2011.

v., 107 f: il. ; 29,7 cm

1. Tese (doutorado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática.

Inclui referências bibliográficas.

1. Informática – Teses. 2. Paralelismo. 3. Linguagens Interpretadas. 4. Desempenho. 5. Tempo de Execução. I. Rodriguez, Noemi. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Dedico esse trabalho a minha esposa Kátia, amada, amiga e companheira,
sem você eu não teria conseguido chegar ao fim dessa jornada.

Agradecimentos

À minha orientadora por toda a paciência, conselhos, força, puxões de orelha e pelas palavras sábias nos momentos mais difíceis da pesquisa.

Aos meus pais, irmãs, sobrinha e avó que me deram muito apoio e força.

Ao Hamilton e Renata Lima que sempre foram pacientes com meus atrasos e ausências.

Ao Philipe e a Nívea que estão esperando um jogo de tênis até hoje.

À PUC-Rio, pelo auxílio concedido, sem o qual este trabalho não poderia ter sido realizado.

Ao Adriano Branco por todas as consultorias de Latex.

Aos amigos que fiz aqui e me deram o apoio para continuar a jornada.

Aos professores que me ofereceram a oportunidade de aprender tanto.

Resumo

Motta Junior, Paulo Rogério da; Rodriguez, Noemi. **Uma Abstração para Programação Paralela Suporte para o Desenvolvimento de Aplicações**. Rio de Janeiro, 2011. 107p. Tese de Doutorado — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

A evolução do campo de programação tradicionalmente troca desempenho por abstrações mais poderosas capazes de simplificar o trabalho do programador. É possível observar os efeitos dessa evolução na área de programação paralela. Tipicamente, a programação paralela se concentra em alto desempenho baseado no paradigma procedural para atingir o mais alto rendimento possível, porém determinar o ponto em que deve-se trocar desempenho por abstrações mais poderosas continua um problema em aberto. Com o advento de novas ferramentas e bibliotecas de sistema que fornecem melhor desempenho sem a intervenção do programador, a crença de que o programador da aplicação deve otimizar o código de comunicação começa a ser questionada. De acordo com a crescente demanda por soluções paralelas de larga escala tornando-se evidentes, problemas como complexidade de código, poder de modelagem e projeto, manutenibilidade, desenvolvimento rápido, maior segurança e reuso, deverão ser considerados quando for necessário decidir que abordagem usar. Nesse trabalho, investigamos o custo do uso de camadas de abstração de mais alto-nível e que podem prover muitos benefícios para desenvolvedores de aplicações paralelas. Além disso argumentamos que o uso de linguagens interpretadas pode ajudar na abstração da arquitetura de processador surgindo a oportunidade para otimizar as máquinas virtuais sem que isso afete o código da aplicação do usuário.

Palavras-chave

Paralelismo; Linguagens Interpretadas; Desempenho; Tempo de Execução.

Abstract

Motta Junior, Paulo Rogério da; Rodriguez, Noemi(Advisor). **An Abstraction for Parallel Programming: Support for Developing Applications**. Rio de Janeiro, 2011. 107p. DSc Thesis — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

The evolution of the field of programming traditionally trades performance for more powerful abstractions that are able to simplify the programmer's work. It is possible to observe the effects of this evolution on the parallel programming area. Typically parallel programming focuses on high performance based on the procedural paradigm to achieve the highest possible throughput, but determining the point in which one should trade performance for more powerful abstractions remains an open problem. With the advent of new system level tools and libraries that deliver greater performance without programmer's intervention, the belief that the application programmer should optimize communication code starts to be challenged. As the growing demand for large scale parallel solutions becomes noticeable, problems like code complexity, design and modeling power, maintainability, faster development, greater reliability and reuse, are expected to take part on the decision of which approach to use. In the present work, we investigate the use of higher-level abstraction layers that could provide many benefits for the parallel application developer. We argue that the use of interpreted languages may aid the abstraction of the processor architecture providing an opportunity to optimize the virtual machines without affecting the user's application code.

Keywords

Parallelism; Interpreted Languages; Performance; Execution Time.

Sumário

1	Introdução	13
1.1	Objetivo	15
1.2	Estrutura do Texto	17
2	Conceitos do Desenvolvimento de Aplicações Paralelas	18
2.1	Arquiteturas Paralelas	18
2.2	Desenvolvimento baseado em Paralelismo e Distribuição	19
2.2.1	Programação Procedural Tradicional	19
2.2.2	Programação Baseada em Abstrações	20
2.3	Linguagens Interpretadas no Paralelismo	21
2.4	Padrões de Paralelismo	23
2.5	Como Quantificar o Desempenho	23
2.5.1	Desempenho	24
2.5.2	Domínio de Aplicação	26
2.5.3	Complexidade	27
2.5.4	As Opções Disponíveis	28
2.6	Conclusão	30
3	O Modelo Numina	31
3.1	Camadas de Abstração de Numina	31
3.2	A API Numina - Um Exemplo	33
3.3	Arquitetura Numina	35
3.4	Ambiente de Execução Numina - Modelo de Cópia de Memória	37
3.5	Camadas Numina de Abstração	38
3.6	Cell Broadband Engine - Cell/BE	39
3.7	Intel/AMD Multi-núcleo	43
3.8	Cluster Híbrido	44
3.9	Reverendo as Políticas de Distribuição de Dados	46
3.10	Conclusão	48
4	Resultados e Avaliação	50
4.1	Estratégia de Testes	50
4.2	Análise Quantitativa	52
4.2.1	Teste de Primalidade de um Conjunto de Números	52
4.2.2	Multiplicação de Matrizes	53
4.2.3	Análise de Resultados Quantitativos	55
4.3	Análise Qualitativa	55
4.3.1	Aproximação de Pi Utilizando Método de Monte-Carlo	56
4.3.2	Cálculo de Integral da Função e^x Através do Método de Trapézios	58
4.3.3	Análise de Resultados Qualitativos	58
4.4	Simulação N-Body	59
4.5	Melhoria de Desempenho Através de JIT	61
4.6	Impacto do Uso de Camadas de Abstração	62
4.7	Conclusão	63

5	Trabalhos Relacionados	65
5.1	Desempenho como Responsabilidade do Sistema	66
5.2	Modelos de Abstração de Paralelismo	68
5.3	Bibliotecas para Suporte a Paralelismo no Cell/BE	74
5.4	Conclusão	77
6	Conclusão e Trabalhos Futuros	78
6.1	Trabalhos Futuros	79
	Referências Bibliográficas	81
A	API Numina em Detalhes	90
A.1	API de Sistema	90
A.2	API Core	91
A.3	API CoreGroup	92
A.4	API Cluster	93
B	Código Fonte das Aplicações para Análise Qualitativa	94
C	Aplicações para Comparação de Tempos entre Camadas	102

Lista de figuras

3.1	Arquitetura do modelo Numina	35
3.2	A execução de uma aplicação em Numina	39
3.3	Visão de um Cluster Numina	45
3.4	Políticas de distribuição de dados com CoreGroup	47
3.5	Políticas de distribuição de dados com Cluster	48

Lista de tabelas

4.1	Média dos tempos de execução - teste de primalidade sequencial	53
4.2	Média dos tempos de execução - teste de primalidade paralelo	53
4.3	Fator de aceleração pelo uso de paralelismo - teste de primalidade	53
4.4	Média dos tempos de execução - multiplicação de matrizes sequencial	54
4.5	Média dos tempos de execução - multiplicação de matrizes paralela	54
4.6	Fator de aceleração pelo uso de paralelismo - multiplicação de matrizes	54
4.7	Média dos tempos de execução em ambiente Cell - multiplicação de matrizes paralela	55
4.8	Quantidade de Linhas de Código e Mecanismos de Sincronização	57
4.9	Quantidade de Linhas de Código por versão	58
4.10	Média dos tempos de execução - simulação N-Body	60
4.11	Quantidade de Linhas de Código e Mecanismos de Comunicação	60
4.12	Comparação dos tempos de execução com o uso de JIT	61
4.13	Fatores de impacto no desempenho pelo uso de Camadas	62

*Toda tecnologia suficientemente avançada não
se distingue da magia.*

Arthur C. Clarke.