

**Cristina Ururahy da Fontoura
Cerqueira**

**Um modelo de programação
multilinguagem para
aplicações geograficamente
distribuídas**

TESE DE DOUTORADO

DEPARTAMENTO DE INFORMÁTICA

Programa de Pós-graduação em
Informática

Rio de Janeiro
setembro de 2003

PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO DE JANEIRO



Cristina Ururahy da Fontoura Cerqueira

**Um modelo de programação
multilinguagem para aplicações
geograficamente distribuídas**

Tese de Doutorado

Tese apresentada ao Programa de Pós-graduação em In-
formática do Departamento de Informática da PUC-Rio
como requisito parcial para obtenção do título de Doutor
em Informática

Orientador: Prof. Noemi de La Rocque Rodriguez

Rio de Janeiro
setembro de 2003



Cristina Ururahy da Fontoura Cerqueira

**Um modelo de programação
multilinguagem para aplicações
geograficamente distribuídas**

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 título de Doutor em Informática. Aprovada pela Comissão Examinadora abaixo assinada.

Prof. Noemi de La Rocque Rodriguez
Orientador
Departamento de Informática — PUC-Rio

Prof. Roberto Ierusalimsky
Departamento de Informática — PUC-Rio

Prof. Markus Endler
Departamento de Informática — PUC-Rio

Prof. Thaís Batista
Departamento de Informática e Matemática Aplicada —
UFRN

Prof. Vinod Rebello
Instituto de Computação — UFF

Prof. Michael Stanton
Instituto de Computação — UFF

Prof. Ney Augusto Dumont
Coordenador Setorial do Centro Técnico Científico —
PUC-Rio

Rio de Janeiro, 12 de setembro de 2003

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

Cristina Ururahy da Fontoura Cerqueira

Graduou-se em Engenharia de Computação na PUC-Rio em 1995. Obteve o título de Mestre em Informática pela PUC-Rio em 1998.

Ficha Catalográfica

Urrahy, Cristina

Um modelo de programação multilinguagem para aplicações geograficamente distribuídas/ Cristina Ururahy da Fontoura Cerqueira; orientador: Noemi de La Rocque Rodriguez. — Rio de Janeiro : PUC-Rio, Departamento de Informática, 2003.

121 f: il. ; 29,7 cm

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

Inclui referências bibliográficas.

1. Informática – Teses. 2. Sistemas distribuídos. 3. Aplicações paralelas. 4. Computação em grade. 5. Linguagens interpretadas. 6. Comunicação orientada a eventos. 7. Coordenação, monitoramento e adaptação de aplicações. I. Rodriguez, Noemi. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 330

Ao Renato e nossos filhos.

Agradecimentos

Ao Renato por todo o seu amor e amizade e por estar sempre presente.

À Noemi por ter acreditado neste trabalho desde o início, por sua paciência e amizade.

Ao Roberto por suas inúmeras contribuições.

Ao Felipe, José Maurício, Beatriz e Ana Lúcia pela alegria que vocês trazem a minha vida.

À minha família por compreender a importância deste trabalho.

À Geiza, Bazílio e Ana por sua companhia.

Ao Alésio e a Ana por sua colaboração.

Ao TecMF, LES, LabPar e Lab-DI pelo excelente ambiente de trabalho oferecido.

À PUC pela bolsa de isenção e pela oportunidade de desenvolver este trabalho e de conhecer pessoas maravilhosas.

Ao CNPq pelo apoio financeiro.

Resumo

Ururahy, Cristina; Rodriguez, Noemi. **Um modelo de programação multilinguagem para aplicações geograficamente distribuídas**. Rio de Janeiro, 2003. 121p. Tese de Doutorado — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Neste trabalho propomos usar o ALua, um mecanismo de comunicação orientado a eventos, baseado na linguagem interpretada Lua, para coordenação e desenvolvimento de aplicações paralelas distribuídas. O ALua adota um modelo de programação multilinguagem para aplicações paralelas distribuídas, agindo como elemento de ligação, permitindo que partes pré-compiladas do programa sejam executadas em diferentes máquinas.

Novas tecnologias em programação paralela, como computação em grade, e o interesse atual em computação distribuída para redes geográficas demandam novos níveis de flexibilidade, como o uso de estratégias de adaptação e a habilidade para um usuário interferir em uma computação sem a necessidade de interrompê-la. Além disso, devido a sua natureza assíncrona, a programação dirigida a eventos oferece um modelo apropriado para ambiente sujeitos a falhas e retardos, que são freqüentes no contexto de redes geográficas.

Neste trabalho, mostramos que o ALua pode trazer a flexibilidade desejada, através de mecanismos de adaptação e monitoramento não só de aplicações, mas do próprio ambiente de execução, e ainda tirar proveito de sua natureza interpretada para permitir a intervenção do usuário na aplicação mesmo durante a sua execução.

Palavras-chave

sistemas distribuídos; aplicações paralelas; computação em grade; linguagens interpretadas; comunicação orientada a eventos; coordenação, monitoramento e adaptação de aplicações.

Abstract

Ururahy, Cristina; Rodriguez, Noemi. **A Multilanguage Programming Model for Geographically Distributed Applications**. Rio de Janeiro, 2003. 121p. PhD. Thesis — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

In this work we propose the use of ALua, an event-driven communication mechanism for coordinating and developing distributed parallel applications, based on the interpreted language Lua. ALua adopts a multilanguage programming model for distributed parallel applications, acting as a gluing element among precompiled program parts running on different machines. New developments in parallel programming, such as Grid computing, and current interest in wide-area distributed computing demand new levels of flexibility, such as the use of adaptive strategies and the ability for an user to interfere with a computation without having to stop it. Furthermore, because of its asynchronous nature, event-driven programming provides a suitable model for environments subject to failures and delays that are frequent in the context of geographically distributed computing.

In this work we show that ALua can achieve the required flexibility through mechanisms for monitoring and adapting not only applications, but also the execution environment, and also exploit its interpretive nature to allow the programmer to modify the behavior of the application during its execution.

Keywords

distributed systems; parallel applications; grid computing; interpreted languages; event-driven communication; coordination, monitoring and adaptation of applications.

Sumário

1	Introdução	11
1.1	A Tese Proposta	14
1.2	Organização do Texto	18
2	Coordenação	19
2.1	Linguagens de <i>Script</i> e de Extensão	22
2.2	Coordenação Orientada a Eventos	23
2.3	O Mecanismo de Coordenação ALua	24
3	ALua	27
3.1	O Modelo Básico do ALua	27
3.2	Suporte de Execução	30
3.3	Exemplos de Programação	31
3.4	Flexibilidade	36
4	Comunicação de Controle e Dados	45
4.1	O ALua Multicanal	45
4.2	Desempenho do ALua Multicanal	50
4.3	Aplicações Multimídia	52
5	Monitoramento e Adaptação	59
5.1	ALuaMonitor	60
5.2	Monitoramento e Adaptação de Aplicações Paralelas em Grades	64
5.3	Uso do ALua com outras Bibliotecas de Comunicação	69
5.4	Considerações Finais	75
6	Conclusão	76
6.1	Principais Contribuições	77
6.2	Trabalhos Correlatos	79
6.3	Trabalhos Futuros	83
7	Referências Bibliográficas	85
A	API do ALua	93
A.1	API Básica	93
A.2	API Estendida	99
B	API do ALuaMonitor	105
B.1	Funções do ALuaMonitor	105
B.2	Funções de um objeto <i>propriedade</i> (<i>Property</i>)	107
C	Listagem de Códigos	110
C.1	Capítulo 4	110
C.2	Capítulo 5	119

Lista de Figuras

3.1	A estrutura de um processo ALua.	29
3.2	O modelo de comunicação do ALua.	31
3.3	Mensagem quase auto-replicante.	37
3.4	Mensagem auto-replicante.	37
3.5	Paradigma de multiplicação de tuplas: esqueleto seqüencial.	39
3.6	Multiplicação de matrizes - código específico da aplicação.	40
3.7	Caminhos mais curtos - código específico da aplicação.	40
3.8	Paradigma de multiplicação de tuplas: esqueleto paralelo.	41
4.1	A estrutura de um processo ALua multicanal.	46
4.2	Um programa ALua multicanal simples.	49
4.3	Código ALua do programa de teste de desempenho.	51
4.4	Tempo de execução para clientes fazendo <i>download</i> de arquivos de 150Mb em uma rede de 100Mbits/s.	52
4.5	A arquitetura do DynaVideo.	54
5.1	Definição da propriedade <code>NUsers</code> .	61
5.2	Adição de um observador.	62
5.3	Definição de um aspecto.	63
5.4	Propriedade que indica a porcentagem de CPU livre.	66
5.5	Aspecto que indica se o uso da CPU está mais intenso.	67
5.6	Observador baseado na propriedade CPU e em seu aspecto <code>Decreasing</code> .	67
5.7	Aplicação mestre/escravo.	68
5.8	A estrutura da aplicação mestre/escravo.	69
5.9	<i>Loop</i> principal do processo mestre.	71
5.10	Aplicação típica ALua + MPI.	73
5.11	<i>Loop</i> principal do processo mestre ALua + MPI.	74

“It is not the strongest of the species that survives, nor the most intelligent, but the one most responsive to change.”

Charles Darwin.