



Alexandre Rupert Arpini Skyrme

**Um Modelo Alternativo para Programação
Concorrente em Lua**

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Redes e Sistemas Distribuídos do Departamento de Informática da PUC-Rio

Orientador: Prof. Noemi de La Rocque Rodriguez

Rio de Janeiro
Junho de 2008



Alexandre Rupert Arpini Skyrme

**Um Modelo Alternativo para Programação
Concorrente em Lua**

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Redes e Sistemas Distribuídos do Departamento de Informática do Centro Técnico-Científico da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Prof. Noemi de La Rocque Rodriguez

Orientador

Departamento de Informática — PUC-Rio

Prof. Roberto Ierusalimschy

PUC-Rio

Prof. Renato Fontoura de Gusmão Cerqueira

PUC-Rio

Prof. Ana Lúcia de Moura

RNP

Prof. José Eugênio Leal

Coordenador Setorial do Centro Técnico-Científico — PUC-Rio

Rio de Janeiro, 06 de Junho de 2008

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

Alexandre Rupert Arpini Skyrme

Graduou-se em Engenharia de Computação na Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio). Durante a graduação trabalhou no Núcleo de Arte Eletrônica (NAE) do Departamento de Artes e Design da PUC-Rio, no Laboratório de Processamento de Sinais (LPS) do Departamento de Engenharia Elétrica da PUC-Rio e no Grupo de Tecnologia em Computação Gráfica (TeCGraf) da PUC-Rio. Após a graduação, trabalhou como consultor de Segurança da Informação e gerente na Cipher, analista de sistemas no Serviço Federal de Processamento de Dados (SERPRO) e analista de sistemas na TMP Consultoria. Atualmente é analista da Gerência de Segurança da Informação do Banco Nacional de Desenvolvimento Econômico e Social (BNDES).

Ficha Catalográfica

Skyrme, Alexandre

Um Modelo Alternativo para Programação Concorrente em Lua / Alexandre Rupert Arpini Skyrme; orientador: Noemi de La Rocque Rodriguez. — Rio de Janeiro : PUC-Rio, Departamento de Informática, 2008.

v., 111 f: il. ; 29,7 cm

1. Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática.

Inclui referências bibliográficas.

1. Informática – Tese. 2. Concorrência. 3. Paralelismo. 4. Lua. 5. Threads. 6. Multithreading preemptivo. 7. Multithreading não-preemptivo. 8. luaproc. I. Rodriguez, Noemi. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 510

Agradecimentos

À professora Noemi Rodriguez, pela simpatia, delicadeza e receptividade constantes, pela compreensão com as restrições de horário, pelo apoio na resolução de percalços da vida acadêmica, pela orientação irrepreensível e pelo interesse instigante no trabalho.

A Bruno Oliveira Silvestre, pela leitura criteriosa do código-fonte, pelas sugestões extremamente pertinentes e pelo apoio na resolução de inúmeros bugs na implementação da biblioteca.

À PUC-Rio, pela bolsa concedida e pelo ambiente sempre acolhedor.

À professora Rejane Spitz, a quem nunca poderei agradecer o suficiente, por toda a ajuda durante a graduação, pelo aprendizado e apoio no ingresso na pós-graduação.

Ao professor Paulo Osório (*in memoriam*), pelo apoio durante a graduação e no ingresso na pós-graduação.

Ao Banco Nacional de Desenvolvimento Econômico e Social (BNDES), pelo horário de trabalho flexível e, em particular, a Herilmar Pompermayer Freire, pelo apoio e pela compreensão com os ocasionais compromissos acadêmicos durante o horário do expediente.

À Tia Patricia (*in memoriam*), a quem dedico este trabalho, pela generosidade decisiva ao meu ingresso na pós-graduação.

À Clarisse Siqueira, pelo amor, carinho, compreensão e paciência, por me acalmar e me ensinar a ser uma pessoa melhor.

Resumo

Skyrme, Alexandre; Rodriguez, Noemi. **Um Modelo Alternativo para Programação Concorrente em Lua**. Rio de Janeiro, 2008. 111p. Dissertação de Mestrado — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

A popularização dos processadores multinúcleo e de tecnologias como o *hyper-threading* evidencia uma mudança de foco na evolução dos processadores. Essa mudança fomenta o interesse por programação concorrente e a exploração de paralelismo para obtenção de melhor desempenho. Entretanto, os modelos atuais para programação concorrente são alvo de críticas recorrentes, o que estimula a elaboração de propostas alternativas. Este trabalho apresenta uma análise crítica do *multithreading* preemptivo com compartilhamento de memória, um modelo amplamente utilizado para programação concorrente, e faz um breve apanhado de trabalhos que abordam alternativas para programação concorrente. Em seguida, propõe um modelo para programação concorrente estruturado com a linguagem de programação Lua e descreve as suas principais características e vantagens. Finalmente, apresenta os resultados da avaliação de diversos aspectos de uma biblioteca desenvolvida para implementar o modelo proposto.

Palavras-chave

Concorrência. Paralelismo. Lua. Threads. Multithreading preemptivo. Multithreading não-preemptivo. luaproc.

Abstract

Skyrme, Alexandre; Rodriguez, Noemi. **An Alternative Model for Concurrent Programming in Lua**. Rio de Janeiro, 2008. 111p. MsC Thesis — Department of Informatics, Pontifícia Universidade Católica do Rio de Janeiro.

The popularization of multi-core processors and of technologies such as hyper-threading indicates a different approach to the evolution of processors. This new approach brings about an increased interest in concurrent programming and the exploration of parallelism in order to achieve better performance. However, concurrent programming models now in use are subject to recurring criticism, which stimulates the development of alternative proposals. This work presents a critical analysis of preemptive multithreading with shared memory, which is a widely used model for concurrent programming, and briefly summarizes some studies that deal with alternatives for concurrent programming. It then, proposes a model for concurrent programming structured with the Lua programming language and describes its main characteristics and advantages. Finally, it presents the results of an evaluation of several aspects of a library developed to implement the proposed model.

Keywords

Concurrency. Parallelism. Lua. Threads. Preemptive multithreading. Non-preemptive multithreading. luaproc.

Sumário

1	Introdução	11
2	Multithreading	13
3	Trabalhos Relacionados	17
3.1	Sequential Object Monitors	17
3.2	Polyphonic C#	18
3.3	Concurrency and Coordination Runtime	19
3.4	Erlang	21
3.5	Programação Dirigida a Eventos sem Inversão de Controle	24
4	Modelo Proposto	28
5	Implementação do Modelo	31
5.1	Estrutura dos Processos Lua	32
5.2	Escalonador	33
5.3	Comunicação entre Processos	34
5.4	Estratégia de Bloqueio	34
5.5	Ambientes Distribuídos	35
6	Avaliação	38
6.1	Medidas de Desempenho	40
6.2	Comparativo com POSIX Threads	52
6.3	Comparativo com Erlang	55
6.4	Considerações	60
7	Conclusão	65
	Referências Bibliográficas	67
A	API da Biblioteca para Programação Concorrente em Lua	71
B	Exemplo Simples de Serviço de Escalonamento Remoto	74
B.1	Servidor	74
B.2	Cliente	75
C	Disparador de Processos Lua com Sincronização	77
D	Envio e Recebimento Simples de Mensagens	79
E	Anel de Mensagens	81
F	Disparador de Processos Lua sem Sincronização	83
G	Disparador de Processos Lua com Retardos	84

H	Busca de Cadeias de Caracteres	86
H.1	Versão Processos Lua	86
H.2	Versão Lua Simples	93
I	Disparador de Pthreads sem Sincronização	96
J	Disparador de Pthreads com Sincronização	98
K	Disparador de Processos Erlang	101
K.1	Versão Interpretada	101
K.2	Versão Compilada	102
L	Envio e Recebimento Simples de Mensagens em Erlang	103
L.1	Versão Interpretada	103
L.2	Versão Compilada	105
M	Anel de Mensagens em Erlang	108
M.1	Versão Interpretada	108
M.2	Versão Compilada	110

Lista de figuras

6.1	Criação sucessiva de processos Lua com estados criados com lua_newthread e luaL_newstate.	41
6.2	Criação sucessiva de processos Lua com carga de todas as bibliotecas padrão nos estados.	42
6.3	Criação sucessiva de processos Lua sem carga de bibliotecas padrão nos estados.	43
6.4	Tempos de execução para a busca de cadeias de caracteres.	51
6.5	Criação sucessiva de Pthreads.	54
6.6	Criação sucessiva de processos Lua sem sincronização entre processos.	54
6.7	Criação sucessiva de processos Erlang.	57

*Software gets slower faster than hardware gets
faster.*

Niklaus Wirth, *A Plea for Lean Software.*