

**Eduardo Fonseca de Andréa**

**Monitorando o Ambiente de Execução de  
Componentes de Software Distribuídos**

**Dissertação de Mestrado**

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

Orientador: Prof. Renato Fontoura de Gusmão Cerqueira

Rio de Janeiro  
Abril de 2009

**Eduardo Fonseca de Andréa**

## **Monitorando o Ambiente de Execução de Componentes de Software Distribuídos**

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

**Prof. Renato Fontoura de Gusmão Cerqueira**

Orientador

Departamento de Informática — PUC-Rio

**Prof. Noemi Rodriguez**

Departamento de Informática — PUC-Rio

**Prof. Antônio Tadeu Azevedo Gomes**

Laboratório Nacional de Computação Científica — LNCC

**Prof. José Eugênio Leal**

Coordenador Setorial do Centro Técnico Científico —  
PUC-Rio

Rio de Janeiro, 13 de Abril de 2009

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

### **Eduardo Fonseca de Andréa**

Graduou-se na Pontifícia Universidade Católica do Rio de Janeiro, cursando Informática. Atua a seis anos na área de informática em empresas dos ramos financeiro, telecomunicação e petróleo.

#### Ficha Catalográfica

Andréa, Eduardo Fonseca de

Monitorando o Ambiente de Execução de Componentes de Software Distribuídos/ Eduardo Fonseca de Andréa; orientador: Renato Fontoura de Gusmão Cerqueira. — 2009.

v., 92 f: il. ; 30 cm

Dissertação (Mestrado em Informática) - Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2009.

Inclui bibliografia.

1. Informática - Teses. 2. Monitoração de recursos. 3. Publicação de dados. 4. Componentes de software. 5. Middleware. 6. Sistemas distribuídos. I. Cerqueira, Renato Fontoura de Gusmão. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

## Agradecimentos

Gostaria de agradecer à minha família pelo o amor que me deram durante toda minha vida.

À CAPES, à PUC-Rio e ao Tecgraf, pelos auxílios concedidos, sem os quais este trabalho não poderia ter sido realizado.

À Carolina, pelo carinho e companheirismo incondicionais dados durante esses difíceis dois anos.

Ao professor Renato Cerqueira, pelas valiosas orientações que nortearam todo o trabalho desenvolvido.

Aos amigos Amadeu Junior e Sand Correa, pessoas extremamente inteligentes e acessíveis que contribuíram intensamente com ótimas idéias.

E finalmente, mas não menos importante, aos amigos Luiz Marques, Marcio Aguiar e Ricardo Leal, pelos trabalhos feitos em conjunto, pelos jogos assistidos, pelas viagens realizadas e principalmente pela grande amizade criada.

## Resumo

Andréa, Eduardo Fonseca de; Cerqueira, Renato Fontoura de Gusmão. **Monitorando o Ambiente de Execução de Componentes de Software Distribuídos**. Rio de Janeiro, 2009. 92p. Dissertação de Mestrado — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Sistemas de componentes têm como característica possibilitar a construção de aplicações através da composição de artefatos de software disponíveis. Interações podem ocorrer entre diversos componentes que podem estar distribuídos em diversas máquinas. À medida que aplicações distribuídas aumentam de tamanho, as interações existentes entre os diversos nós que a compõem vão se tornando mais complexas. Assim, torna-se importante para essas aplicações a existência de uma forma de monitorar as interações entre os componentes, com o intuito de identificar falhas e gargalos de processamento e comunicação no sistema. Este trabalho apresenta uma arquitetura capaz de oferecer mecanismos extensíveis para coleta de informações do ambiente de execução desses sistemas, e das interações realizadas entre os seus componentes. São implementadas formas de publicação dessas informações obtidas e testes comparativos para quantificar como a arquitetura desenvolvida onera o desempenho da aplicação.

## Palavras-chave

Monitoração de Recursos, Publicação de Dados, Componentes de Software, Middleware, Sistemas Distribuídos

## Abstract

Andréa, Eduardo Fonseca de; Cerqueira, Renato Fontoura de Gusmão(Advisor). **Monitoring the Execution Environment of Distributed Software Components**. Rio de Janeiro, 2009. 92p. MSc. Dissertation — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Component-based systems are characterized by the construction of applications through the composition of available software artifacts. Interactions may occur between different components that can be distributed through several machines. As distributed applications increase in size, the interactions between the various nodes that comprise them become more complex. Therefore it is important for distributed component systems to monitor the interactions between components in order to identify failures and bottlenecks in processing and communication. This dissertation presents an architecture capable of offering extensible mechanisms for monitoring the execution environment of distributed components, and the interactions between their components. It also presents a flexible mechanism for publication of the collected information, and some comparative tests to measure the performance penalty imposed by the infrastructure to the application.

## Keywords

Resource Monitoring, Data Dissemination, Software Components, Middleware, Distributed Systems

## Sumário

1	Introdução	11
1.1	Objetivos	13
1.2	Solução Proposta	14
1.3	Estrutura do Texto	15
2	Trabalhos Relacionados	16
2.1	LuaMonitor	16
2.2	Chung e Chang	19
2.3	Debusmann et al	20
2.4	Marchetti et al	22
2.5	Considerações Finais	24
3	O Sistema de Componentes de Software	26
3.1	Modelo de Interação	27
3.2	Modelo de Configuração	28
3.3	Modelo de Introspecção	29
3.4	O Componente SCS	29
3.5	Ambiente de Execução	31
3.5.1	<i>Container</i>	31
3.5.2	Nós de Execução	32
3.6	Implementações do SCS	34
4	Arquitetura	35
4.1	Comunicação	35
4.2	Topologia	36
4.3	Recursos	37
4.4	Desenho da Arquitetura	37
4.4.1	Exemplo de Uso	40
5	Coleta e Publicação de Métricas	42
5.1	Interceptadores	43
5.1.1	Detalhes de Interceptadores Java e Lua	44
5.2	Descrição de Métricas	46
5.3	Instrumentando o SCS	48
5.3.1	Com Interceptadores	49
5.3.2	Com <i>Threads</i>	52
5.3.3	Definição de Novas Métricas	53
5.4	Publicação	54
5.4.1	Estrutura de Dados	56
5.4.2	Cliente Agregador	58
5.4.3	Monitor Gráfico	59
5.4.4	Políticas de Publicação	60
6	Avaliação Experimental	68

6.1	Produtor Consumidor	68
6.1.1	Descrição	69
6.1.2	Implementação	70
6.1.3	Resultados	72
6.2	Uso das Políticas	75
6.3	Map Reduce	78
6.3.1	Descrição	78
6.3.2	Implementação	80
6.3.3	Resultados	82
7	Conclusões	85
7.1	Avaliação do Estudo	85
7.2	Trabalhos Futuros	87
8	Referências Bibliográficas	89



## Lista de Figuras

2.1	<i>Smart Proxy</i> [1]	17
2.2	Arquitetura para Adaptação Dinâmica de Aplicações Distribuídas [1]	18
2.3	Camadas do GRIM [2]	19
2.4	Arquitetura do AppMan [3]	21
2.5	Pontos de Interceptação Utilizando ARM [3]	22
2.6	Redirecionamento Permanente [4]	23
2.7	Redirecionamento Não Permanente [4]	23
2.8	Piggybacking [4]	23
2.9	Uso de <i>Proxy</i> [4]	24
3.1	A Estrutura de um Componente SCS	29
3.2	<i>Container</i> SCS	32
3.3	<i>container</i> SCS	33
4.1	Configuração do Exemplo <i>PingPong</i>	39
4.2	Componentes PingPong Conectados	41
5.1	Medidas Retiradas com Interceptadores	44
5.2	Métricas Básicas Coletadas	47
5.3	Arquitetura SCS Instrumentada	56
5.4	Estrutura de Dados	57
5.5	Interface Gráfica da Ferramenta Monitora	60
5.6	Políticas AOC, AACS, e AACD	63
5.7	Políticas ARI e ADI	65
5.8	Política ACTC	65
6.1	Produtor Consumidor	69
6.2	Tempo Médio das Execuções do Produtor Consumidor	73
6.3	Tempo Médio para Tarefas de Tamanho 100.000	74
6.4	Número de Mensagens Publicadas em Cada Tipo de Instrumentação	75
6.5	Resultados da política AACD	76
6.6	Resultados da política ADI	77
6.7	Resultados da política ACTC	77
6.8	MapReduce [5]	79
6.9	<i>Framework</i> MapReduce Implementado com Componentes SCS	80
6.10	Diagrama de Classes para o <i>Framework</i> MapReduce	83
6.11	Tempos Coletado no <i>MapReduce</i>	84

## Lista de Códigos

2.1	Pseudo-Código do Aspecto de Tendência de Uso . . . . .	18
3.1	Facetas do Componente SCS . . . . .	30
3.2	Execution Node . . . . .	33
4.1	Implementação do Método Ping em Java . . . . .	40
5.1	IDL do InstrumentationManager . . . . .	48
5.2	Código Java que Obtém a Instância do Componente . . . . .	52
5.3	Código Parcial da Classe MetricCollector . . . . .	53
5.4	Metrica . . . . .	54
5.5	InstrumentationManager . . . . .	55
5.6	Estrutura de Dados MachineStats . . . . .	58
5.7	Interface do <i>StatsCollection</i> . . . . .	58
5.8	Estrutura da Política de Publicação . . . . .	61
5.9	Classe Policy . . . . .	66
5.10	Script Lua para Determinar a Política de Publicação . . . . .	66
6.1	IDL ProducerConsumer . . . . .	70
6.2	Funções Básicas <i>ProducerConsumer</i> . . . . .	71
6.3	Método que Inicializa a Instrumentação de um <i>Container</i> . . . . .	72
6.4	Pseudo-Código <i>MapReduce</i> [5] . . . . .	79
6.5	Interfaces IDL Usadas no <i>Framework</i> MapReduce . . . . .	81