

**Elbio Renato Torres Abib**

**Escalonamento de Tarefas  
Divisíveis em Redes Estrela**

**DISSERTAÇÃO DE  
MESTRADO**

**DEPARTAMENTO DE INFORMÁTICA**

Programa de Pós-graduação em  
Informática

Rio de Janeiro  
Junho de 2004





**Elbio Renato Torres Abib**

**Escalonamento de Tarefas Divisíveis em  
Redes Estrela**

**Dissertação de Mestrado**

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

Orientador: Prof. Celso Carneiro Ribeiro

Rio de Janeiro  
Junho de 2004





**Elbio Renato Torres Abib**

**Escalonamento de Tarefas Divisíveis em  
Redes Estrela**

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. Celso Carneiro Ribeiro**

Orientador

Departamento de Informática — PUC-Rio

**Prof. Alexandre Plastino de Carvalho**

UFF

**Prof. Maria Cristina Boeres**

UFF

**Prof. Noemi Rodriguez**

PUC-Rio

**Prof. José Eugênio Leal**

Coordenador Setorial do Centro Técnico Científico —

PUC-Rio

Rio de Janeiro, 30 de Junho de 2004



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

**Elbio Renato Torres Abib**

Graduou-se em Ciência da Computação na Universidade Federal do Rio Grande do Sul (Porto Alegre-RS, Brasil).

Ficha Catalográfica

Abib, Elbio R. T.

Escalonamento de Tarefas Divisíveis em Redes Estrela/ Elbio Renato Torres Abib; orientador: Celso Carneiro Ribeiro. — Rio de Janeiro : PUC-Rio, Departamento de Informática, 2004.

v., 106 f: il. ; 29,7 cm

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

Inclui referências bibliográficas.

Tarefas Divisíveis, Escalonamento, Computação em Grade, Busca Local, Programação Inteira. I. Ribeiro, C. C.. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 510





A minha esposa, aos meus pais e a minha irmã.



## Agradecimentos

A minha esposa Vânia Berwanger Boschi Abib, aos meus pais Elbio Renato da Silva Abib e Maria Elza Macedo Torres Abib e a minha irmã Renata Torres Abib, que tanto sofreram pela saudade e, mesmo assim, tanto me incentivaram durante o desenrolar deste trabalho

Ao meu orientador Celso C. Ribeiro que sempre me instigou e me permitiu desenvolver uma boa pesquisa.

Ao CNPq e à PUC-Rio, pelos auxílios concedidos, sem os quais este trabalho não poderia ter sido realizado.

Aos meus colegas da PUC-Rio, que tanto me auxiliaram quando estive distante.

Ao pessoal do departamento de Informática pela a ajuda de todos os dias, em particular à Débora e à Carmem.



## Resumo

Abib, Elbio R. T. ; Ribeiro, C. C.. **Escalonamento de Tarefas Divisíveis em Redes Estrela**. Rio de Janeiro, 2004. 106p. Dissertação de Mestrado — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

O problema de escalonamento de tarefas divisíveis consiste em determinar como uma carga a ser processada deve ser dividida entre processadores e em que ordem cada fração de carga será enviada a cada processador. Considera-se o escalonamento em redes estrela com computadores e enlaces heterogêneos. Nesta dissertação são propostas formulações originais deste problema como modelos de programação linear inteira mista, assim como um novo algoritmo de complexidade  $O(n)$  para a solução ótima de um caso especial. Além disso, também são propostas duas novas heurísticas para o problema, que permitem a elaboração de bons escalonamentos para instâncias de grande porte em um reduzido tempo de processamento.

## Palavras-chave

Tarefas Divisíveis, Computação em Grade, Escalonamento, Processamento Distribuído, Processamento Paralelo.



## Abstract

Abib, Elbio R. T. ; Ribeiro, C. C.. **Divisible Job Scheduling in Star Networks**. Rio de Janeiro, 2004. 106p. MSc. Dissertation — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

The problem of divisible job scheduling consists of determining how to divide the data to be processed among processors and in which order each fraction should be sent to them. In this dissertation, we consider the divisible load scheduling problem in star networks with heterogeneous computers and links. Original mixed integer linear programming formulations of this problem are proposed, as well as a new algorithm with complexity  $O(n)$  to find the optimal solution for a special case. We also propose two fast heuristics that achieve good results for instances representing large scale computing systems.

## Keywords

Divisible Jobs; Divisible Load; Grid Computing; Scheduling; Parallel Computing; Distributed Computing.





# Sumário

1	Introdução	<b>23</b>
1.1	Modelo de sistema	24
1.2	Escalonamento de tarefas divisíveis	27
1.3	Trabalhos correlatos	28
1.3.1	Período único	30
1.3.2	Múltiplos períodos	30
1.4	Organização da dissertação	31
2	Escalonamento usando apenas um período	<b>33</b>
2.1	Resultados anteriores para sistemas sem latências	33
2.2	Resultados anteriores para sistemas com latências	35
2.3	Novo algoritmo rápido para resolução com ordem pré-determinada	40
2.4	Novos modelos de programação linear inteira mista	43
2.4.1	Construção do limite inferior para o problema geral	45
2.4.2	Modelo simplificado	45
2.4.3	Modelo com limites melhorados	47
2.4.4	Modelo aprimorado com desigualdades válidas	48
2.5	Nova heurística construtiva com retro-alimentação <i>HeuRet</i>	52
2.6	Busca local	54
3	Escalonamento usando múltiplos períodos	<b>59</b>
3.1	Resultados anteriores para sistemas sem latências	61
3.2	Resultados anteriores para sistemas com latências	61
3.2.1	Heurística com período fixo <i>HFix</i>	62
3.2.2	Heurística adaptativa <i>HAdapt</i>	62
3.3	Nova heurística <i>HeuMul</i> para escalonamento em múltiplos períodos	63
3.4	Novos modelos de programação linear inteira mista	65
3.4.1	Modelo com número fixo de processadores e períodos	66
3.4.2	Modelo com número máximo de períodos	68
4	Resultados computacionais	<b>73</b>
4.1	Sistema usado para execução e simulação das técnicas	73
4.2	Implementações das técnicas de escalonamento	76
4.3	Base de testes	77
4.4	Técnicas de escalonamento em período único	77

4.4.1	Modelos de programação linear inteira mista	78
4.4.2	Heurística <i>HeuRet</i> e buscas locais	80
4.5	Técnicas de escalonamento em múltiplos períodos	83
4.5.1	Modelos de programação linear inteira mista	84
4.5.2	Heurística <i>HeuMul</i>	85
4.5.3	Heurísticas de período fixo e adaptativo	89
5	Conclusão e trabalhos futuros	<b>93</b>
6	Referências bibliográficas	<b>95</b>
A	Visualizador de soluções	<b>99</b>
A.1	Formato do arquivo de entrada	101
B	Exemplo de utilização das classes criadas	<b>105</b>

## Lista de Figuras

1.1	Rede estrela	25
1.2	Escalonamento ótimo em apenas um período	28
1.3	Escalonamento não ótimo em apenas um período	28
1.4	Escalonamento em múltiplos períodos	29
2.1	Exemplo de escalonamento em período único (sistema com latências)	34
2.2	Escalonamento ótimo em apenas um período (sistema sem latências)	36
2.3	Escalonamento ótimo em apenas um período (sistema com latências)	37
3.1	Escalonamento em três períodos	60
3.2	Escalonamento em três períodos sempre utilizando sete processadores	60
3.3	Escalonamento usando a heurística de período fixo (aproximação 1x)	63
3.4	Escalonamento usando a heurística de período fixo (aproximação 20x)	63
3.5	Escalonamento usando a heurística adaptativa (aproximação 1x)	64
3.6	Escalonamento com número fixo de processadores e períodos	65
3.7	Escalonamento ótimo resultante do modelo linear inteiro com número fixo de processadores e períodos	69
3.8	Escalonamento ótimo resultante do modelo linear inteiro com número fixo de processadores e períodos	69
3.9	Escalonamento ótimo resultante do modelo linear inteiro com número máximo de períodos	70
4.1	Diferença percentual dos valores das soluções de MIP1ER em relação aos valores das soluções de MIP1R	79
4.2	Tempo de resolução de MIP1R e MIP1ER	80
4.3	Melhoria percentual média conseguida pelas buscas locais a partir da heurística <i>HeuRet</i> .	82
4.4	Tempo de execução da heurística <i>HeuRet</i> e das buscas locais.	83

4.5	Diferença percentual dos <i>makespans</i> conseguidos por <i>HeuRet</i> , dos <i>makespans</i> ótimos (Opt) conseguidos por MIP1 e dos valores das soluções de MIP1ER em relação aos valores das soluções de MIP1R.	84
4.6	Resultado ótimo encontrado pelo modelo MIP2	85
4.7	Resultado encontrado pela heurística <i>HeuMul</i>	85
4.8	Resultado ótimo encontrado pelo modelo MIP2	86
4.9	Resultado encontrado pela heurística <i>HeuMul</i>	86
4.10	Diferença percentual dos <i>makespans</i> obtidos por <i>HeuMul</i> em relação aos obtidos por <i>HeuRet</i> .	87
4.11	Diferença percentual dos <i>makespans</i> obtidos por <i>HeuMul</i> em relação aos obtidos por <i>HeuRet</i> .	88
A.1	Imagem da interface do visualizador	99
A.2	Escalonamento não ótimo em apenas um período	100
A.3	Escalonamento ótimo em apenas um período	101
A.4	Escalonamento em três períodos	101
A.5	Escalonamento em dois períodos	103

## Lista de Tabelas

4.1	Combinação de características de sistema testados	77
4.2	Tempo médio ( <i>segundos</i> ) de resolução de MIP1 para diferentes sistemas	81
4.3	Tempo médio ( <i>segundos</i> ) de resolução de MIP1E para diferentes sistemas	81
4.4	Diferença percentual média dos <i>makespans</i> obtidos por <i>HeuMul</i> em relação aos obtidos por <i>HeuRet</i>	88
4.5	Média dos <i>makespans</i> obtidos por <i>HFix</i> e, entre parênteses, média dos <i>makespans</i> obtidos por <i>HeuRet</i> para instâncias com 10, 20, 40, 80 e 160 processadores	91
4.6	Média dos <i>makespans</i> obtidos por <i>HAdapt</i> e, entre parênteses, média dos <i>makespans</i> obtidos por <i>HeuRet</i> para instâncias com 10, 20, 40, 80 e 160 processadores	92



## Lista de Algoritmos

1	<b>AlgRap</b> : Algoritmo rápido para resolução com ordem pré-determinada . . . . .	44
2	<b>CalcLimS</b> : Cálculo dos limites $W_{i,j}$ . . . . .	48
3	<b>CalcLimI</b> : Cálculo dos limites $W'_{i,j}$ . . . . .	52
4	<b>HeuRet</b> : Heurística construtiva com retro-alimentação . . . . .	55
5	<b>BL</b> : Busca local . . . . .	57
6	<b>SimStar</b> : Simulador . . . . .	76

