UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Ciência de dados aplicada em programas paralelos MPI para modelagem e predição de desempenho

Benício Ramos Magalhães

Trabalho de Conclusão de Curso - MBA em Ciência de Dados (CEMEAI)

ICMC USP SÃO CARLOS

UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Ciência de dados aplicada em programas paralelos MPI para modelagem e predição de desempenho

Benicio Ramos Magalhaes

USP - São Carlos
2020

BENICIO RAMOS MAGALHAES

Ciência de dados aplicada em programas paralelos MPI para modelagem e predição de desempenho

Trabalho de conclusão de curso apresentado ao Centro de Ciências Matemáticas Aplicadas à Indústria do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, como parte dos requisitos para conclusão do MBA em Ciência de Dados.

Área de concentração: Ciências de Dados

Orientador: Prof. Dr. Antonio Castelo Filho

USP - São Carlos

2020

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

M188c

Magalhaes, Benicio Ramos
Ciência de dados aplicada em programas paralelos MPI para modelagem e predição de desempenho / Benicio Ramos Magalhaes; orientador Antonio Castelo Filho. -- São Carlos, 2020.
p.

Trabalho de conclusão de curso (MBA em Ciência de Dados) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2020.

1. Trabalho de conclusão de curso. 2. MBA. 3. Ciência de dados. 4. Computação paralela. 5. Predição de desempenho. I. Filho, Antonio Castelo, orient. II. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

ERRATA

FOLHA DE AVALIAÇÃO OU APROVAÇÃO

DEDICATÓRIA

A minha esposa pela compreensão, carinho e apoio incansável.

AGRADECIMENTOS

“Jamais considere seus estudos como uma obrigação, mas como uma oportunidade invejável para aprender a conhecer a influência libertadora da beleza do reino do espírito, para seu próprio prazer pessoal e para proveito da comunidade à qual seu futuro trabalho pertencer.”

- Albert Einsten

RESUMO

MAGALHÃES, B. R. **Ciência de dados aplicada em programas paralelos MPI para modelagem e predição de desempenho**. 2020. XX f. Trabalho de conclusão de curso (MBA em Ciência de Dados) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2020.

Este trabalho é um estudo de técnicas de predição de desempenho de softwares de computação paralela MPI utilizando ciência de dados. O objetivo é modelar o comportamento desses programas e estudar diversas técnicas existentes na área de ciência de dados e com isso predizer seu comportamento com relação à métricas não conhecidas. Inicialmente, realizamos uma pesquisa bibliográfica acerca dos principais trabalhos relacionados à previsão de desempenho de programas paralelos e, em seguida, obtivemos uma base de dados de métricas de desempenho. A partir dessa base, elaboramos um modelo representativo e, por fim, realizamos predições de desempenho com relação às métricas coletadas.

Palavras-chave: Modelagem. Avaliação de desempenho. Computação paralela MPI. Ciência de dados. Análise de predição de dados.

ABSTRACT

MAGALHAES, B. R. **Data science applied in parallel MPI programs for modeling and performance prediction**. 2020. XX f. Trabalho de conclusão de curso (MBA em Ciência de Dados) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2020.

This work is a study of the performance prediction technique of MPI parallel computing software using data science. The objective is to model the behavior of these programs and to study several existing techniques in the area of data science and thereby predict their behavior in relation to unidentified measures. Initially, we performed a bibliographic search on the main works related to the performance prediction of parallel programs, and then we obtained a database of performance metrics. From this base, we elaborate a representative model and, finally, we make performance predictions regarding the collected metrics.

Keywords: Modeling. Performance evaluation. MPI parallel computing. Data science. Data prediction analysis.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AET	–	Application Execution Time
CRISP-DM	–	Cross Industry Standard Process for Data Mining
DP*Graph	–	Modelo de representação gráfica de programas paralelos
DP*Graph+	–	Modelo de representação gráfica de programas paralelos atualizado
MAE	–	Mean Absolute Error
MPI	–	Message Passing Interface
PAS2P	–	Parallel Application Signature for Performance Prediction
PET	–	Predicted Execution Time
PETE	–	Prediction Execution Time Error
RAE	–	Relative Absolute Error
RMSE	–	Root Mean Squared Error
RRSE	–	Root Relative Squared Error
SET	–	Signature Execution Time
T*Graph	–	Modelo de representação gráfica de programas paralelos de alto nível

LISTA DE SÍMBOLOS

SUMÁRIO

1 INTRODUÇÃO

Em um mundo tecnológico e moderno, onde os sistemas computacionais oferecem diversos benefícios à sociedade, existe uma necessidade cada vez maior de trabalhar com aplicações de alto desempenho e isso tornou-se viável por meio da utilização de sistemas distribuídos. “Um sistema distribuído é um conjunto de computadores independentes que se apresenta a seus usuários como um sistema único e coerente.” (TANENBAUM, 2007, p.1). Com isso os desenvolvedores começaram a procurar meios para escrever aplicações distribuídas com alta eficiência e isso resultou no MPI (*Message Passing Interface*) que é um padrão de comunicação de dados através de troca de mensagens para programas paralelos.

“Computação paralela é mais que uma estratégia para atingir um alto desempenho, ela é uma visão de como a computação pode ser escalonada para ter um poder computacional praticamente ilimitado.” (DONGARRA et al, 2003, p.24).

Esse aumento de poder computacional com baixo custo tem viabilizado a resolução de problemas complexos aplicada em diversas áreas de conhecimento.

Para aferir e garantir que uma aplicação atenda aos requisitos não-funcionais de alto desempenho, exigidos principalmente por sistemas de missão crítica, existem diversas técnicas de avaliação de desempenho. Jain (1991) define uma avaliação de desempenho como uma arte, portanto, assim como uma obra de arte, toda a avaliação requer um conhecimento íntimo do que está sendo modelado e uma seleção cuidadosa da metodologia, carga de trabalho e ferramentas.

Outra área de interesse neste trabalho, envolve o estudo de modelos preditivos para estimar o comportamento dos sistemas com relação ao seu desempenho. De acordo com SAS (2020), os modelos preditivos utilizam resultados conhecidos para desenvolver (ou treinar) um modelo que pode ser usado para prever valores para dados diferentes ou novos.

Neste trabalho, vamos nos basear em uma metodologia de análise de desempenho de programas paralelos proposta em Laine (2003), onde iremos realizar um estudo adicional de predição de desempenho com ênfase nas diversas técnicas existentes em ciência de dados, obtendo modelos variados, realizando simulações e comparando os resultados previstos com os dados reais medidos.

1.1 Objetivos

Este trabalho tem como objetivo principal estudar técnicas e metodologias de predição existentes em ciência de dados aplicados em avaliações de desempenho de sistemas computacionais. Para isso, utilizamos alguns modelos de programas paralelos elaborados por Laine (2003) e propomos algumas técnicas de predição de desempenho usando algoritmos de aprendizado de máquina e estatística.

Como objetivos específicos, montamos um modelo representativo para explicar as características principais de um sistema computacional com relação as suas funções e seus parâmetros. Também aplicamos simulações no modelo para predizer seu comportamento com relação à métricas não conhecidas.

1.2 Motivação

A ciência de dados é uma área que se tem mostrado promissora para resolver problemas reais de negócios, com o uso de métodos científicos e técnicas avançadas para captura, tratamento de dados, aprendizado de máquina, redes neurais e inteligência artificial.

De maneira geral, a principal motivação para avaliar o desempenho de programas computacionais é melhorar a eficiência dos algoritmos para que seja viável resolvermos problemas de grande complexidade de forma mais rápida e barata.

1.3 Justificativa

O intuito é aplicar os conhecimentos dessa área numa base de dados obtida através de testes de desempenho de um programa paralelo MPI para que seja possível modelar e prever o comportamento de programas sobre condições que ainda não foram testadas.

1.4 Metodologia

Primeiramente foi feito um levantamento bibliográfico acerca das principais técnicas de predição de dados aplicados a sistemas de programas paralelos. Em seguida, mediante uso de programação paralela MPI e conceitos de execução de testes de desempenho, foi obtida uma

base de dados de métricas de desempenho do programa, como por exemplo, tempo de processamento dada uma variação nos parâmetros de entrada. Na sequência, criamos alguns modelos representativos com as principais características do sistema e a partir desses modelos elaborados, realizamos predições de desempenho utilizando simulações e técnicas de ciência de dados.

1.5 Organização do trabalho

O próximo capítulo apresentará alguns trabalhos relacionados à predição de desempenho de programas paralelos MPI, seguida das técnicas mais utilizadas em ciência de dados. No capítulo 3, detalharemos o que é computação paralela focado na programação MPI, como realizar medições e extrair métricas para análise de dados. O capítulo 4 aborda com mais detalhes o que é uma avaliação de desempenho computacional e quais as técnicas utilizadas para avaliação deste trabalho acadêmico. O capítulo 5 apresenta quais são e como são aplicados os modelos de predição com aprendizado de máquina. Já o capítulo 6 é a implementação dos algoritmos, geração e análise dos resultados, verificando a diferença entre os modelos, discutindo sua eficácia e comparando as predições com os dados reais medidos em testes. Finalmente, algumas conclusões decorrentes deste estudo são apresentadas no capítulo 7.

2 REVISÃO BIBLIOGRÁFICA

Fizemos um levantamento bibliográfico onde destacamos alguns trabalhos relevantes para o tema de pesquisa deste trabalho. As teses mais relevantes, onde baseamos o estudo com relação à metodologia, base de dados e resultados foram realizados por Li (2001) e Laine (2003), porém, avaliamos também outros trabalhos mais recentes de predição que são interessantes de serem citados como a PAS2P proposta por Wong, Rexachs e Luque (2015) e CRISP-DM proposta em Chapman et al [1]* (2000 *apud* QAZDAR et al, 2019, p.3580).

A referência Li (2001) apresenta uma metodologia de análise e predição de desempenho utilizando MPI em redes de estação de trabalho. Foi realizada uma representação dos programas paralelos MPI utilizando grafos, denominados *T-graph**. Com isso é possível conhecer todo o fluxo de execução do algoritmo e, para representar os programas paralelos, foi utilizada uma outra classe conhecida como *DP*Graph*, na qual possibilita identificar de forma clara suas concorrências e possíveis gargalos. Além da criação do modelo representativo, foram utilizadas algumas técnicas de avaliação de desempenho e assim obter predições baseado em dados experimentais.

O trabalho desenvolvido por Laine (2003) tem como base a proposta feita por Li (2001), porém, ele inclui representações novas considerando estruturas de repetição em seus algoritmos e também programas do tipo *master/slave.* Além disso, ele propõe um novo modelo representativo conhecido como *DP*Graph+*, que permite obter uma maior precisão na implementação de um programa MPI.

Em Wong, Rexachs e Luque (2015) é proposta uma metodologia chamada de PAS2P (*Parallel Application Signature for Performance Prediction*) onde dados de desempenho do sistema são coletados e caracterizados por suas fases de execução (comportamento de repetição do algoritmo) e pesos (valor associado às métricas coletadas). Com essa informação é realizada uma predição do tempo de execução da aplicação paralela e depois validada através de resultados experimentais.

Por fim, a metodologia proposta por Chapman et al* (2000 *apud* QAZDAR et al, 2019, p.3580) é chamada de CRISP-DM (*Cross Industry Standard Process for Data Mining*). A metodologia consiste em alguns passos que vai desde o entendimento do negócio, passando

[1] *Chapman, P., Clinton, J., Kerber, R., Khabza, T., Reinartz, T., Shearer, C., & Wilrth, R. (2000). CRISP-DM 1.0 step-by-step data mining guide. The CRISP-DM consortium.

pelo entendimento e preparação dos dados, a modelagem em si, que envolve implementações de aprendizado de máquina e a validação do modelo. Uma vez que o modelo está criado, testado e validado, a última etapa consiste na implantação, que pode ser um relatório com os resultados obtidos.

2.1 Trabalho desenvolvido por Li, K. C.

A metodologia proposta por Li é aplicada em redes de estações de trabalho, que possibilita a utilização de PCs/estações de trabalho conectados via rede de alta velocidade atuando como um sistema distribuído de grande escala.

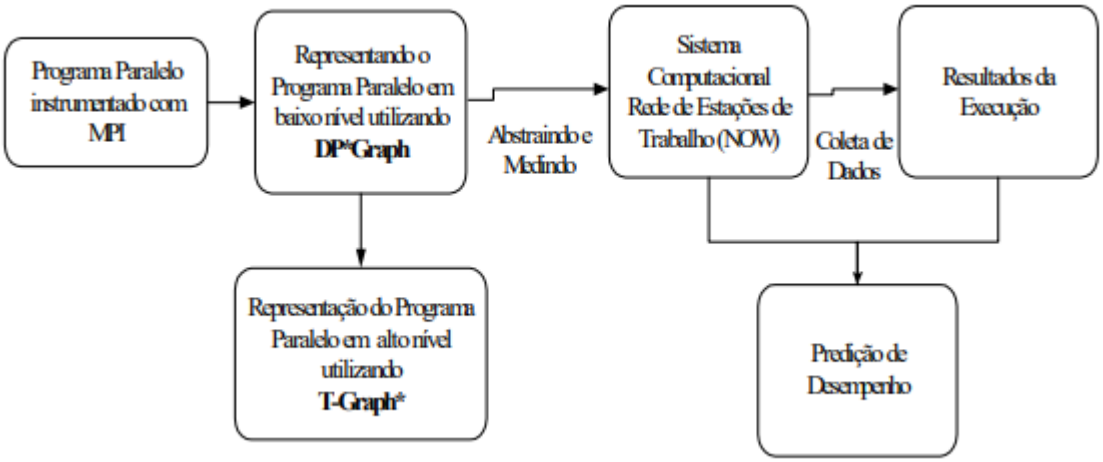

Figura 2.1 - Metodologia de Análise e Predição proposta por Li

A figura 1 mostra a estrutura completa da metodologia utilizada (LI, 2001, p.44). Ela é uma abordagem híbrida que utiliza técnicas de modelagem e execuções de testes experimentais não funcionais com variação nos seus parâmetros de entrada, como número de processadores e tamanho dos dados a serem processados. Para aplicar a metodologia são definidos os seguintes passos:

1. Analisar o código fonte do programa paralelo e criar uma representação de baixo nível utilizando *DP*Graph.*
2. Realizar a instrumentação do programa paralelo incluindo medições de tempo no início e fim de cada trecho de código. Uma vez instrumentado, o programa passa por uma bateria de testes com cenários que variam em número de nós de processamento e

tamanho dos dados, obtendo assim os tempos de execução de cada trecho para cada cenário proposto.

3. Criar um modelo matemático com base nos dados coletados. Neste passo são formuladas equações que calculam o tempo gasto para cada trecho de código, onde seus coeficientes são obtidos através da solução de um sistema de equações lineares obtidos dos dados experimentais.
4. Com os modelos definidos para cada trecho de código, é possível realizar predições de desempenho utilizando a equação que calcula o tempo gasto em cada trecho de código fixando um dos parâmetros e variando o outro. O tempo total de execução é dado pela somatória do tempo obtido nos modelos parciais.
5. A partir da representação de baixo nível, é possível ainda construir uma representação do programa em alto nível com *T-graph** e com esta simplificação, é possível demonstrar o fluxo de execução do programa paralelo, sem explicar detalhes como a comunicação entre os nós.

2.2 Trabalho desenvolvido por Laine, J. M.

Laine utiliza a metodologia aplicada em Li (2001) aprimorando algumas etapas do processo. Na metodologia original existe um modelo gráfico chamado *DP*Graph*, porém, ele aplica uma nova simbologia para melhor representar o código do programa modelado chamado *DP*Graph+*.

Para ilustrar, a figura 2.2 apresenta os símbolos utilizados no *DP*Graph* (LI, 2001, p.45)*:*

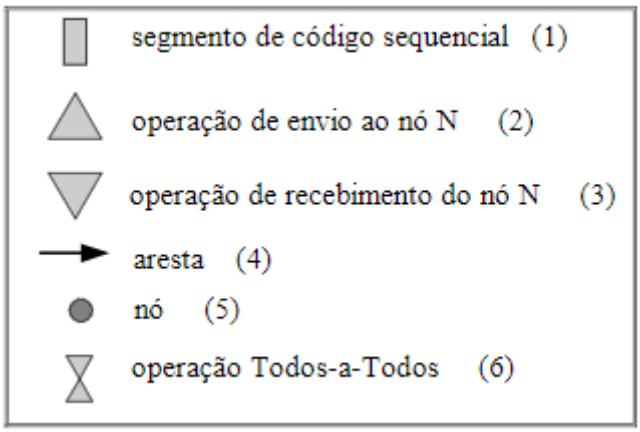

*Figura 2.2 - Símbolos de representação da classe DP*Graph*

A figura 2.3, inclui os símbolos criados por Laine para o *DP*Graph+:*

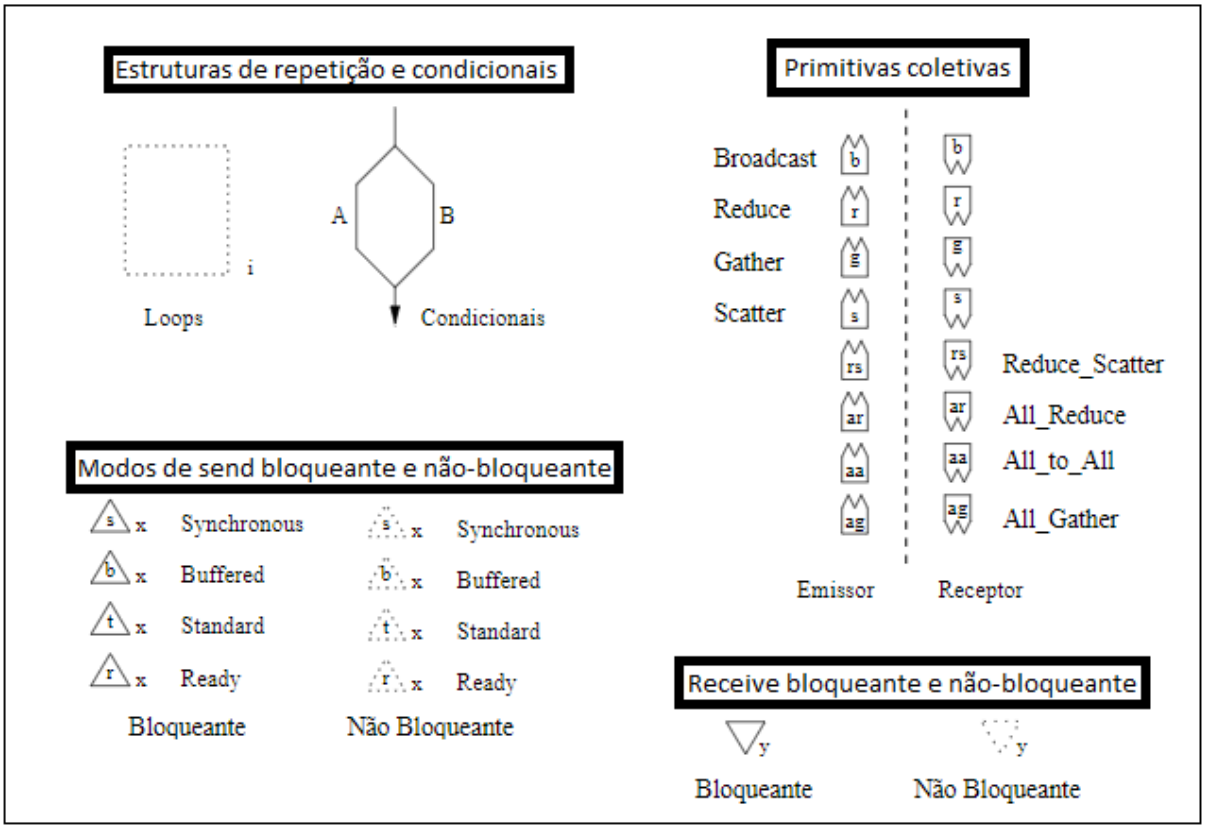

*Figura 2.3 - Símbolos de representação da classe DP*Graph+*

E a seguir a figura 2.4 mostra um exemplo da representação de um programa de multiplicação de matrizes em *DP*Graph+* extraída de (LAINE, 2003, p.54):

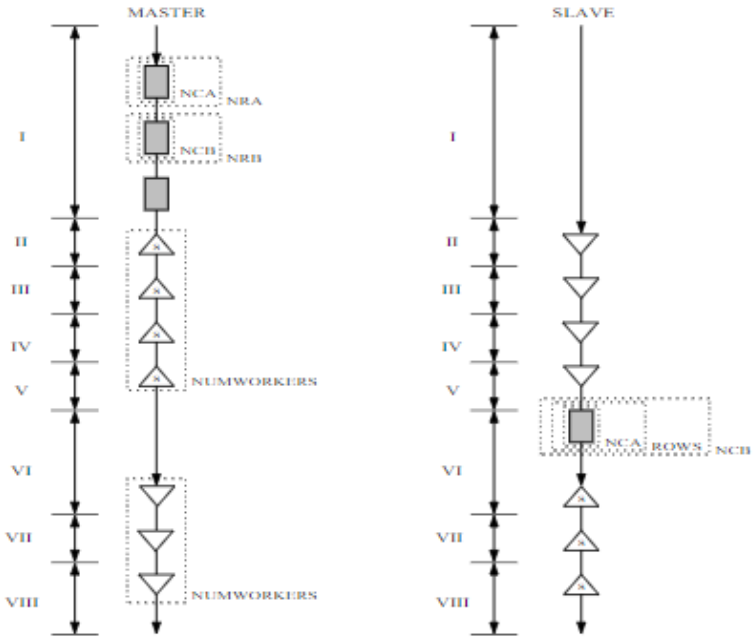

*Figura 2.4 - DP*Graph+ de exemplo para um programa paralelo MPI de multiplicação de matrizes.*

Seguindo com a metodologia apresentada, o desenvolvimento do modelo analítico é obtido através dos seguintes passos:

1. Instrumentar o código MPI;
2. Executar testes não-funcionais com o programa instrumentado em ambiente de testes variando seus parâmetros de entrada;
3. Coletar as métricas de desempenho em cada trecho onde tem-se interesse em modelar;
4. Aplicar técnicas de ajuste de curvas sobre os dados coletados.

Com os dados dos testes experimentais, foi utilizado um critério de seleção onde desconsideraram todos os resultados que ficaram 30% acima do menor valor obtido e o modelo matemático utilizado foi o método de ajuste de curvas dos mínimos quadrados.

Por fim, é feita uma análise detalhada de cada fluxo de execução do algoritmo e é obtida uma equação do tempo de processamento parcial do processo. O modelo final do tempo de execução do programa é dado pela soma dos modelos parciais e estes tempos preditos são comparados e validados com os tempos obtidos experimentalmente. A partir deste ponto ele obteve resultados estimados de tempo de execução com boa precisão, apresentando erros em torno de 5%.

A figura 2.5 apresenta um gráfico com os erros percentuais obtidos do tempo predito em relação ao tempo medido para uma das aplicações avaliadas por essa metodologia (LAINE, 2003, p.90):

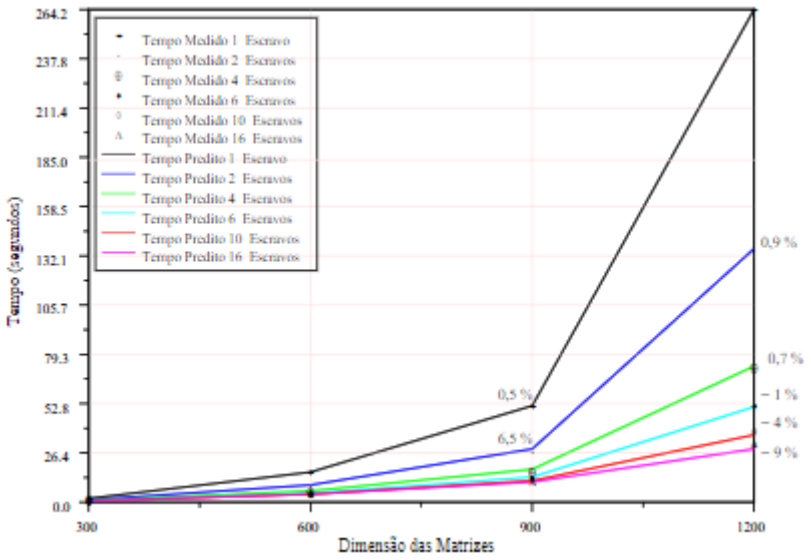

Figura 2.5 - Resultados obtidos pela metodologia utilizada por Laine, comparando tempo predito contra o tempo medido.

2.3 PAS2P

PAS2P (*Parallel Application Signature for Performance Prediction*) proposta por Wong, Rexachs e Luque (2015) é um método de predição de desempenho empenhado em descrever uma aplicação baseada no seu comportamento. O PAS2P consiste basicamente em dois estágios:

1. Análise da aplicação e geração de sua assinatura;
2. Predição de desempenho.

A figura 2.6 apresenta uma visão geral da metodologia (WONG, REXACHS, LUQUE, 2015, p.2010):

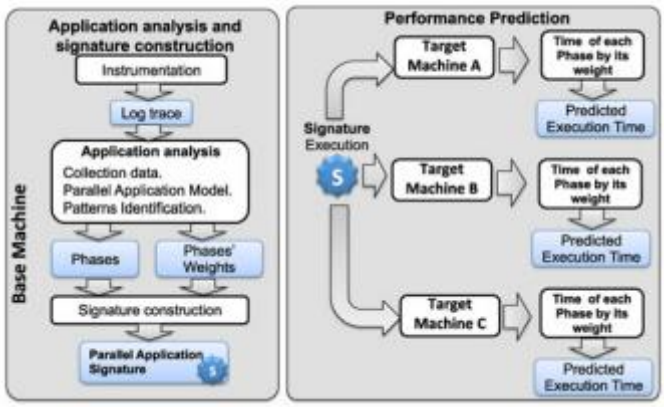

Figura 2.6 - Visão geral da metodologia PAS2P

A primeira etapa é o processo de gerar a assinatura da aplicação, que consiste em instrumentar o programa e executá-lo em uma máquina base gerando um log de rastreamento. Para modelar a aplicação, são coletadas métricas de tempo de execução dividindo as etapas de processamento em fases. Essas fases são agrupadas e é estipulado um peso para cada uma, que é definido pelo número de vezes em que elas ocorrem (repetições de um mesmo tipo de comunicação). A partir disso, as fases são marcadas (*checkpoints*) e finalmente são geradas as assinaturas, que nada mais são do que marcações do código instrumentado que sabem exatamente onde começa e termina cada fase.

A figura 2.7 exemplifica os passos para extração e determinação das fases do algoritmo (WONG, REXACHS, LUQUE, 2015, p.2013):

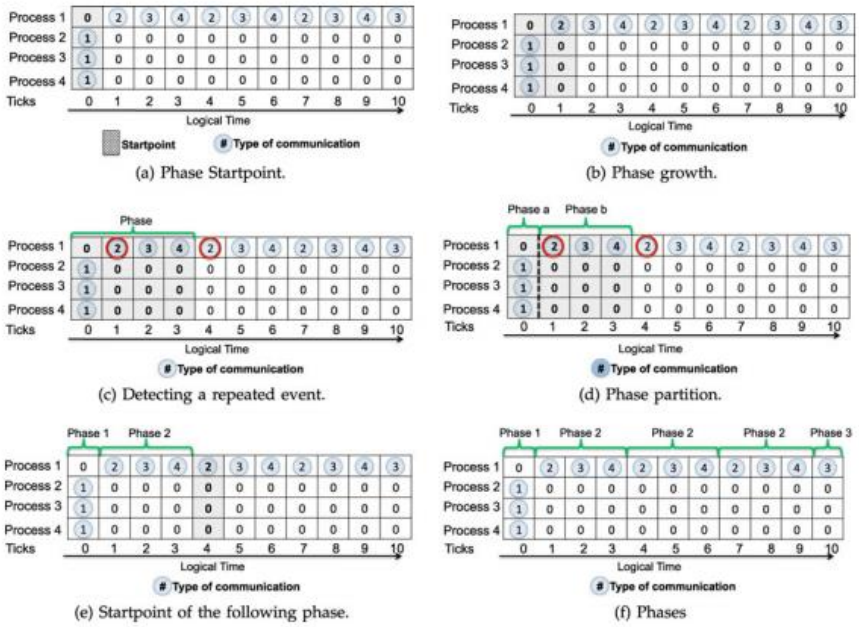

Figura 2.7 - Exemplificação do padrão do algoritmo de identificação das fases.

A figura 2.8 exemplifica uma tabela de fases que serve como base para construção da assinatura (WONG, REXACHS, LUQUE, 2015, p.2014):

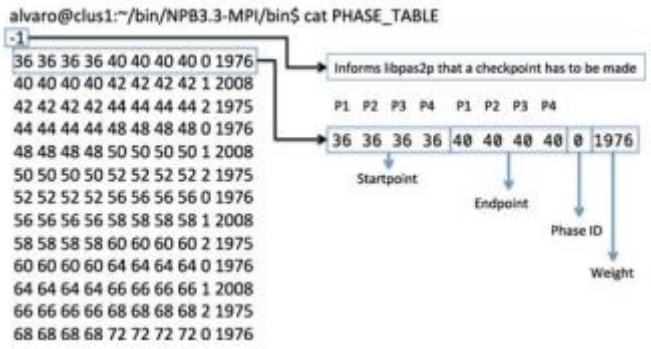

Figura 2.8 - Tabela de fases para construção da assinatura.

A segunda etapa consiste na predição de desempenho, que ocorre a partir da assinatura obtida na etapa um. Para predizer o tempo de execução (PET) da aplicação é utilizada uma equação que consiste na multiplicação do tempo de execução de cada fase pelo seu peso.

Com as informações referentes à assinatura, executa-se um novo teste nas máquinas onde desejamos predizer o comportamento da aplicação. Essa execução mede o tempo de cada fase mapeada na assinatura e no final faz o cálculo da predição do tempo de execução. A principal vantagem é que a execução da assinatura da aplicação costuma ser muito mais eficiente do que a execução da aplicação em si, pois ele considera as estruturas de repetição como fases similares e os tempos preditos validados experimentalmente apresentam erros máximos de apenas 3%. Vale observar que nesta metodologia, caso a aplicação avaliada não apresente essa característica de repetição, o tempo de execução das fases será bem similar ao tempo de execução real da aplicação.

A tabela 2.1 apresenta um exemplo com os erros percentuais obtidos do tempo predito de algumas aplicações em relação ao tempo medido utilizando essa metodologia (WONG, REXACHS, LUQUE, 2015, p.2016). Vale observar que o tempo de execução da assinatura (SET) é bem inferior ao tempo de execução da aplicação (AET) e também que percentual de erro do tempo predito (PETE) é bem pequeno, chegando ao máximo de 3% de variação.

Tabela 2.1 - Exemplos de resultados de predição de desempenho utilizando a metodologia PAS2P

Appl.	Cores	SET (Sec)	SET versus AET(%)	PET (Sec)	PETE(%)	AET (Sec)
CG-64	32	8.42	0.29	2793.42	1.90	2847.42
	64	4.87	0.32	1504.66	0.48	1511.91
BT-64	32	13.47	0.80	1652.65	0.9	1667.64
	64	10.19	0.77	1302.76	0.55	1309.91
SP-64	32	2.04	0.24	808.76	1.28	819.17
	64	2.08	0.51	388.367	3.05	400.55
SMG2k	32	16.75	2.63	633.23	0.38	635.61
64	64	8.37	10.15	162.87	2.32	166.74
Sweep	16	4.32	0.17	2494.36	0.06	2492.74
3d-32	32	3.01	0.22	1328.04	0.40	1322.62
POP-64	32	22.79	1.41	1608.85	0.17	1611.59
	64	18.36	1.79	1016.01	0.61	1022.28

SET: Signature Execution Time, SET versus AET: 100(SET/AET).
PET: Predicted Execution Time, AET: Application Execution Time.
PETE: Prediction Execution Time Error.

2.4 CRISP-DM

Mesmo não estando diretamente relacionada ao tema de predição de desempenho de programas paralelos, a metodologia apresentada por Chapman et al *apud* Qazdar et al chamou a atenção por conter um processo genérico para criação de modelos e predição de resultados.

A figura 2.9 apresenta uma visão geral da metodologia CRISP-DM (QAZDAR et al, 2019, p.3581):

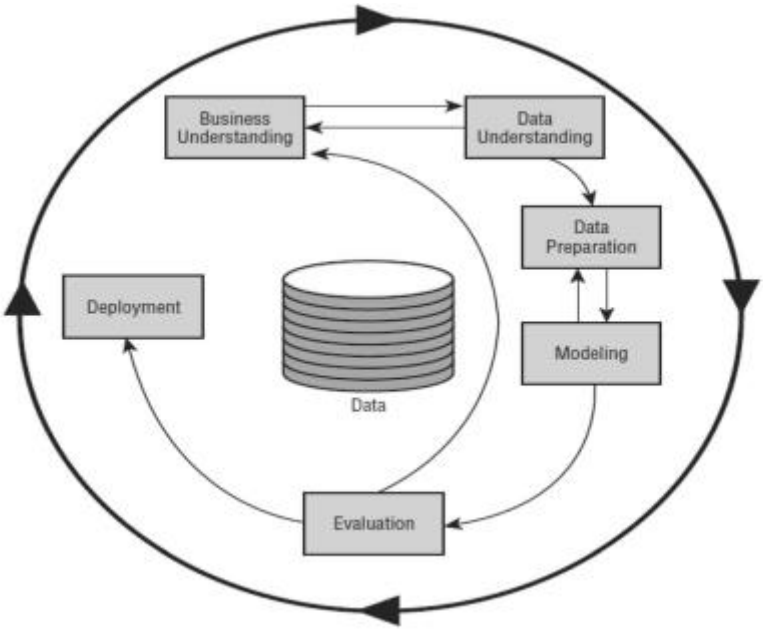

Figura 2.9 - Visão geral da metodologia CRISP-DM

Para aplicarmos a metodologia CRISP-DM (*Cross Industry Standard Process for Data Mining*) são definidos os seguintes passos (QAZDAR et al, 2019, p.3580):

1. Entendimento do negócio em todos os seus aspectos, como por exemplo, os objetivos do projeto, os requisitos, regras aplicáveis, campo de aplicação, etc.;
2. Entendimento dos dados focado nas técnicas adequadas de coleta, descrição, exploração e manipulação;
3. Preparação dos dados. É a fase que cobre todas as atividades necessárias para o tratamento dos dados, obtendo assim uma base pronta para ser manipulada na fase de modelagem;
4. Modelagem. É a implementação de diferentes técnicas de aprendizado de máquina (regressão, classificação, clusterização, recomendação). A escolha desses algoritmos depende da necessidade do projeto, da base de dados e dos resultados almejados;
5. Avaliação do modelo. Aplicam-se técnicas de medidas de erro, como por exemplo, erro absoluto médio (MAE), erro quadrático médio (RMSE), erro absoluto relativo (RAE), erro quadrático relativo (RRSE), acurácia, entre outros. A escolha da avaliação do modelo está diretamente relacionada aos requisitos do projeto, o algoritmo usado e aos resultados desejados;
6. Entrega de resultados. Uma vez com o modelo criado, testado e avaliado, a entrega pode ser um relatório ou uma implementação do processo.

3 COMPUTAÇÃO PARALELA

Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.

3.1 Introdução

Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.

3.2 Modelos de programação paralela

Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.

3.3 Desempenho de programas MPI

Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.

3.4 Considerações finais

Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.

4 TÉCNICAS DE MODELAGEM E AVALIAÇÃO DE DESEMPENHO

Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.

4.1 Introdução

Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.

4.2 Técnicas de avaliação de desempenho

Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.

4.3 Predição de desempenho

Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.

4.4 Considerações finais

Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.

5 MODELOS DE PREDIÇÃO EM CIÊNCIA DE DADOS

Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.

5.1 Introdução

Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.

5.2 Técnicas de predição utilizando modelos estatísticos

Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.

5.3 Algoritmos de aprendizado de máquina

Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.

5.4 Considerações finais

Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.

6 IMPLEMENTAÇÃO, RESULTADOS E DISCUSSÃO

6.1 Introdução

Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.

6.2 Ambiente de teste

Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.

6.3 Levantamento de dados

Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.

6.4 Aplicando predição de desempenho com algoritmos de aprendizado de máquina

Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.

6.5 Resultados obtidos

Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.

6.6 Considerações finais

Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.

7 CONCLUSÕES E TRABALHOS FUTUROS

7.1 Conclusões

Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.

7.2 Considerações finais

Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.

REFERÊNCIAS

AGUILAR, X., FÜRLINGER, K., LAURE E., Visual MPI Performance Analysis using Event Flow Graphs, **Procedia Computer Science**, v.51, p. 1353-1362, Doi: 10.1016/j.procs.2015.05.322

ANTONELLO, R. **Desenvolvimento De Um Modelo De Sistema Multiagente Para Previsão De Retorno Sobre Índices De Ações**. 2010. 109f. Dissertação (Mestrado) – Universidade Federal de Santa Catarina, Florianópolis, 2010.

BAO, L. et al. Performance Modeling and Workflow Scheduling of Microservice-Based Applications in Clouds. **IEEE Transactions on Parallel and Distributed Systems,** v.30, n.9, p. 2114-2129, Doi: 10.1109/TPDS.2019.2901467

BÁN, D., et al. Prediction models for performance, power, and energy efficiency of software executed on heterogeneous hardware. **J Supercomput**, v.75, p. 4001–4025, Doi: 10.1007/s11227-018-2252-6

BATISTA G. E. A. P. A., PRATI, R. C., MONARD, M. C. A study of the behavior of several methods for balancing machine learning training data. **SIGKDD Explor**. v.6, p. 20–29, Doi: 10.1145/1007730.1007735

BOURQUE P., FAIRLEY, R. E., eds., Guide to the Software Engineering Body of Knowledge, Version 3.0, **IEEE Computer Society**, 2014; Disponível em: www.swebok.org Acesso em: 30 Jun. 2020.

CAETANO, M. M. **O Uso de Técnicas de Aprendizado de Máquina na Predição de Desempenho Acadêmico de Alunos em Cursos Superiores.** 2016. 175f. Dissertação (Mestrado) - Curso de Ciência da Computação, Faculdade Campo Limpo Paulista, Campo Limpo Paulista, 2016.

CAMPOS, F. M. Teste de desempenho: Conceitos, Objetivos e Aplicação - Parte 1. **Linha de Código.** Disponível em: http://www.linhadecodigo.com.br/artigo/3256/teste-de-desempenho-conceitos-objetivos-e-aplicacao-parte-1.aspx Acesso em: 25 Jun. 2020.

DONGARRA, J., et al. **Sourcebook Of Parallel Computing**. San Francisco: Morgan Kaufmann Publishers, 2003.

DREISEITL, S., OHNO-MACHADO, L. Logistic regression and artificial neural network classification models: a methodology review, **Journal of Biomedical Informatics**, v.35, p. 352-359, February 2003.

GASPARINI, R., ÁLVARO, A. Análise entre algoritmos de aprendizado de máquina para suportar a predição do posicionamento do jogador de futebol. **Revista Brasileira de Computação Aplicada**, v.9, n.2, p. 70-83, Julho 2017.

GROVE, D. A., CODDINGTON, P. D. Coddington, Modeling message-passing programs with a Performance Evaluating Virtual Parallel Machine, **Performance Evaluation**, v.60, p. 165-187, Doi:10.1016/j.peva.2004.10.019

GROVE, D. **A. Performance Modelling of Message-Passing Parallel Programs**. 2003. 313f. Tese (Doutorado) – Department of Computer Science, The University of Adelaide, Adelaide, 2003.

GODFREY, L. **Bootstrap Tests for Regression Models**. London: Palgrave Macmillan, 2009.

GUIRADO, A. G. **Critérios robustos de seleção de modelos de regressão e identificação de pontos aberrantes**. 2019. 78f. Dissertação (Mestrado) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2019.

GUNTHER, N. J. **Guerrilla Capacity Planning:** a tactical approach to planning for highly scalable applications and services. California: Springer, 2006.

GRUZ, J. **Data Science do Zero:** primeiras regras com o python. Tradução de Wellington Nascimento. Rio de Janeiro: Alta Books, 2016.

HARING, G., LINDEMANN, C., REISER, M. **Performance Evaluation:** Origins and Directions. Berlin, Heidelberg, New York, Barcelona, Hong Kong, London, Milan, Paris, Singapore, Tokyo: Springer, 2000.

HEARST, M. A., et al. Support vector machines, in **IEEE Intelligent Systems and their Applications**, vol.13, n.4, p. 18-28, Doi: 10.1109/5254.708428.

HERAI, R. H. **Proposta de um sistema de modelagem e predição analítica de desempenho para uma plataforma de processamento paralelo.** 2005. 154f. Dissertação (Mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e De Computação, Campinas, São Paulo, 2005.

HUNOLD, S., et al. PGMPI: Automatically Verifying Self-Consistent MPI Performance Guidelines. **ArXiv**, Disponível em: https://arxiv.org/abs/1606.00215 Acesso em: 30 Jun. 2020.

JAIN, R. **The Art of Computer Systems Performance Analysis:** techiniques for experimental desing, measurement, simulation and modeling. New York: John Wiley & Sons, 1991.

KALOS, M. H., WITHLOCK, P. A. **Monte Carlo Methods**: Fundamental Algorithms. 2.ed. WeinHeim: WILEY-VCH Verlag GmbH & Co. KGaA, 2008.

KSHEMKALYANI, A. D., SINGHAL, M. **Distributed Computing:** Principles, Algorithms, and Systems. 2007. Disponível em: https://www.cs.uic.edu/~ajayk/DCS-Book. Acesso em: 25 Jun. 2020.

LAINE, J. M. **Desenvolvimento de modelos para predição de desempenho de programas paralelos MPI**. 2003. 129f. Dissertação (Mestrado) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2003. Doi:10.11606/D.3.2003.tde-28082003-184400.

LAINE, J. M. **Uma metodologia para desenvolvimento de programas paralelos eficientes em ambientes homogêneos e heterogêneos**. 2008. 136f. Tese (Doutorado) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2008. Doi:10.11606/T.3.2016.tde-29112016-085713.

LI, K. C. **Análise e Predição de Desempenho de Programas Paralelos em Redes de Estações de Trabalho**. 2001. 113f. Tese (Doutorado) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2001.

MANCILLA, J. T. N. **Modelagem E Simulação De Um Secador Intermitente De Fluxos Contracorrentes Para Frutos Do Cafeeiro**. 2015. 107f. Dissertação (Mestrado) – Centro de Ciências Agrárias, Universidade Federal do Espírito Santo, Alegre, 2015.

MASSETTO, F. I. **Hybrid MPI - Uma Implementação MPI para Ambientes Distribuídos Híbridos**. 2007. 119f. Tese (Doutorado) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2007.

MERCIER, M. **Contribution to High Performance Computing and Big Data Infrastructure Convergence**. 2019. 203f. Tese (Doutorado) - Symbolic Computation [cs.SC]. Université Grenoble Alpes, Grenoble, 2019.

MONTGOMERY, D. C., PECK, E. A., VINING, G. G. **Introduction to Linear Regression Analysis.** 5.ed. New York: Wiley, 2012.

MONTGOMERY, D. C., RUNGER, G. C. **Applied Statistics and Probability for Engineers.** New York: John Wiley & Sons, 2003.

MPI FORUM. **MPI Documents**. Disponível em: https://www.mpi-forum.org/docs/ Acesso em: 25 jun. 2020.

OLIVEIRA, H. M. **Modelagem e predição de desempenho de primitivas de comunicação MPI**. 2003. Dissertação (Mestrado) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2003. Doi:10.11606/D.3.2003.tde-26082003-113045.

PETERSEN, W. P., ARBENZ, P. **Introduction to Parallel Computing:** a practical guide with examples in c. New York: Oxford University Press Inc., 2004.

POLLARD, A., MEWHORT, D. J. K., WEAVER, D. F. **High Performance Computing Systems And Applications.** New York, Boston, Dordrecht, London, Moscow: Kluwer Academic Publishers, 2002.

QAZDAR, A., et al. A machine learning algorithm framework for predicting students performance: A case study of baccalaureate students in Marocco. **Springer Science+Business Media, LLC, part of Springer Nature**, v.24, p. 3577-3589, Doi: 10.1007/s10639-019-09946-8

RICHARD, S., et al. Performance of MPI Parallel Applications. In: **International Conference on Software Engineering Advances (ICSEA'06)**, Tahiti, 2006, p. 59-64, Doi: 10.1109/ICSEA.2006.261315

SAS. Análises Preditivas: o que são e qual sua importância? **SAS**. Disponível em: https://www.sas.com/pt_br/insights/analytics/analises-preditivas.html#:~:text=An%C3%A1lises%20preditivas%20usam%20dados%2C%20algoritmos,que%20poder%C3%A1%20acontecer%20no%20futuro Acesso em: 26 Jun. 2020.

SUN J., et al. Automated Performance Modeling of HPC Applications Using Machine Learning, **in IEEE Transactions on Computers**, v.69, n.5, p. 749-763, 1 May 2020, Doi: 10.1109/TC.2020.2964767

RAMESH, S., et al. MPI performance engineering with the MPI tool interface: The integration of MVAPICH and TAU, **Parallel Computing**, v.77, p. 19-37, Doi: 10.1016/j.parco.2018.05.003

TANENBAUM, A., STEEN M. V. **Sistemas distribuídos:** princípios e paradigmas. 2.ed. Tradução de Arlete Simille Marques. São Paulo: Pearson Prentice Hall, 2007.

WANG M., MAY, A. J., KNOWLES P. J. Parallel programming interface for distributed data, **Computer Physics Communications**, v.180, p. 2673-2679, 2009, Doi: 10.1016/j.cpc.2009.05.002

WONG A., REXACHS D., LUQUE E., Parallel Application Signature for Performance Analysis and Prediction, **in IEEE Transactions on Parallel and Distributed Systems**, v.26, n.7, p. 2009-2019, 1 July 2015, Doi: 10.1109/TPDS.2014.2329688

ZHANG, W., CHENG M. K., SUBHLOK, J. DwarfCode: A Performance Prediction Tool for Parallel Applications, **in IEEE Transactions on Computers**, v.65, n.2, p. 495-507, 1 Feb. 2016, Doi: 10.1109/TC.2015.2417526

GLOSSÁRIO

Apêndice A – Digitar o título do apêndice A

Apêndice B – Digitar o título do apêndice B

ANEXO A – Digitar o título do anexo A

ÍNDICE