

Jornada de Iniciação Científica e Tecnológica do LNCC

Petrópolis, 01 e 02 de setembro de 2021.

Laboratório Nacional de Computação Científica – LNCC

Diretor

Fabio Borges de Oliveira

Coordenação de Gestão e Administração - COGEA

Sergio Ferreira de Figueiredo

Coordenação de Métodos Matemáticos e Computacionais - COMAC

Frédéric Gerard Christian Valentin

Coordenação de Modelagem Computacional - COMOD

Laurent Emmanuel Dardenne

Coordenação de Pós-Graduação e Aperfeiçoamento - COPGA

Sandra Mara Cardoso Malta

Coordenação de Tecnologia da Informação e Comunicação - COTIC

Wagner Vieira Léo

Programa Institucional de Bolsas de Iniciação Científica &

Programa Institucional de Bolsas de Iniciação em Desenvolvimento Tecnológico e Inovação

Marcos Garcia Todorov

Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq

Presidente

Evaldo Ferreira Vilela

Coordenadora Geral do PIBIC/PIBITI

Lucimar Batista de Almeida

Jornada de Iniciação Científica e Tecnológica do LNCC

Comissão Interna do PIBIC/PIBITI-LNCC

Marcos Garcia Todorov
Antonio Tadeu Azevedo Gomes
Eduardo Lucio Mendes Garcia
Jack Baczynski
Fábio Lima Custódio

Avaliadores Externos

Leonardo Goliatt da Fonseca - UFJF
Carlos Cristiano Hasenclever Borges - UFJF

Apresentação

O LNCC realiza este ano a XVIII Edição da Jornada de Iniciação Científica e Tecnológica, que é um fórum de divulgação das pesquisas desenvolvidas no contexto dos Programas Institucionais de Bolsas de Iniciação Científica (PIBIC) e de Bolsas de Iniciação Tecnológica (PIBITI) fomentados pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq). No período de setembro de 2020 a agosto de 2021, o PIBIC e PIBITI congregaram alunos de várias instituições de ensino e de diversas áreas do conhecimento. Este volume apresenta os resumos dos trabalhos desenvolvidos pelos bolsistas no período. Durante a Jornada, os trabalhos são apresentados pelos bolsistas oralmente e avaliados por um comitê científico externo.

Nesta XVIII Edição da Jornada, o Comitê Externo de Avaliação do PIBIC/PIBITI tem a seguinte composição:

Prof. Leonardo Goliatt da Fonseca- UFJF

Prof. Carlos Cristiano Hasenclever Borges - UFJF

Destacamos o papel relevante do PIBIC/PIBITI do LNCC no desenvolvimento das pesquisas no LNCC e, principalmente, na formação complementar dos bolsistas, promovendo o aprimoramento do conhecimento, espírito criativo, reflexão crítica e ética. Estas características têm contribuído para suas inserções no mercado de trabalho e em programas de pós-graduação, como o PPG em Modelagem Computacional do LNCC. Este é o resultado do esforço e dedicação de todos os participantes.

Agradecimentos

Agradecemos ao CNPq pelas bolsas concedidas, à Direção do LNCC pelo apoio e à Comissão Interna do PIBIC e PIBITI no LNCC.

Agradecemos a disponibilidade e contribuição dos membros do Comitê Externo de Avaliação. O sucesso desta Jornada, e do Programa como um todo, é o resultado da dedicação e do esforço de toda a comunidade do LNCC. Expressamos em particular nosso reconhecimento ao apoio concedido pela secretaria do PPG-LNCC e, em particular, à Sra. Roberta Machado.

Marcos Garcia Todorov
Coordenador do PIBIC/PIBITI - LNCC

Índice

Bolsistas PIBIC ativos

Avaliação de Métodos de Aprendizado de Máquina no Desenvolvimento de Portais Científicos Inteligentes.

Bolsista: Alexandre Clem Belchior dos Santos

Orientadores: Kary Ann del Carmen Ocaña Gautherot, Douglas Cardoso e Carla Osthoff Ferreira de Barros

Sistema Embarcado para robô Humanoide

Bolsista: Ana Luiza Buse da Silva

Orientadores: Jauvane Cavalcante de Oliveira e Erick Menezes Moreira

Desenvolvimento de algoritmo de para estabelecimento de rede AD HOC usando aeronaves remotamente pilotadas dotadas de antenas repetidoras

Bolsista: Arthur Santiago de Oliveira Meneses

Orientador: Jauvane Cavalcante de Oliveira e Erick Menezes

Exploração de Módulos Paralelo-Híbrido de Bioinformática para Ambientes GP de Supercomputação

Bolsista: Guilherme Freire da Silva Dornelas

Orientadores: Carla Osthoff Ferreira de Barros, Kary Ann del Carmen Ocaña Gautherot

Projeto e Implementação de Workflows Científicos Reprodutíveis de Alto Desempenho

Bolsista: Lucas da Cruz Silva

Orientadores: Luiz Gadelha, Carla Osthoff e Kary Ocaña

Modelagem e Integração de Bancos de Dados Relacionais na arquitetura do Bioinfo-Portal

Bolsista: Marco Antonio Silva Cabral

Orientador: Kary Ann del Carmen Ocaña Gautherot e Marcelo Monteiro Galheigo

Aplicação de Conceitos Topológicos para Análise de Imagens

Bolsista: Mariana Dória Prata Lima

Orientadores: Gilson Antônio Giraldo e Gastão Florêncio Miranda Junior

Algoritmos Quânticos

Bolsista: Matheus de Araújo Ferreira

Orientador: Renato Portugal

Criação Automática de Árvores Arteriais em Territórios Vasculares com Ênfase na Vasculatura Renal

Bolsista: Rodrigo Schmidt

Orientador: Pablo Javier Blanco

Introdução à Computação Quântica e Criptografia

Bolsista: Thaina Lucciola Hipolito de Lima

Orientador: Renato Portugal

Fatoração rápida via Método das Curvas Elípticas

Bolsista: Thaís Henter de Melo Dias

Orientador: Fábio Borges de Oliveira

Inversão de Dados Sísmicos.
Bolsista: Vinícius Theobaldo Jorge
Orientador: Marcio Rentes Borges

Uso de Aprendizado de Máquina para Melhorias no Desempenho e no Consumo de Energia em Sistemas HPC

Bolsista: Vitor Carlos Soares Sá
Orientadores: Bruno Schulze e Mariza Ferro

Análise de Desempenho e Características de Paralelização dos Algoritmos de Aprendizado de Máquina

Bolsista: Vitor Goncalves Vieira
Bolsista: Bruno Schulze e Mariza Ferro

Método dos Volumes Finitos aplicados para a solução de problemas de mudança de fase

Bolsista: Ophir Valiante Duarte Neto
Orientadores: Elson Magalhães Toledo e Maicon William Niebus Odone

Bolsistas PIBITI ativos

Gerenciamento de grandes volumes de dados nos sistemas Web do SINAPAD

Bolsista: Lucas Fernandes Calheiros
Orientador: Antônio Tadeu Azevedo Gomes

Auto localização robótica e mapeamento com aprendizado de máquina

Bolsista: Gustavo Alves Casqueiro
Orientador: Jauvane Cavalcante de Oliveira

Relatório de Atividades

Título do Projeto

Avaliação de Métodos de Aprendizado de Máquina no Desenvolvimento de Portais Científicos Inteligentes.

Instituição

Laboratório Nacional de Computação Científica.

Nome do Bolsista

Alexandre Clem Belchior dos Santos

Orientadores

- D.Sc. Kary Ann del Carmen Ocaña Gautherot (Tecnologista Pleno – LABINFO/LNCC, Orientador)
- D.Sc. Douglas Cardoso (Professor Associado – CEFET/RJ, Colaborador – LNCC, Coorientador)
- D.Sc. Carla Osthoff Ferreira de Barros (Tecnologista Sênior – SEPAD/COTIC//LNCC, Coorientador)

Tipo de Bolsa

PIBIC

Período do Relatório

01/04/2021 – 31/07/2021

Objetivos

O projeto almeja, em linhas gerais, otimizar através de métodos de aprendizado de máquina a utilização dos recursos computacionais do LNCC. O modelo preditivo foi aplicado em execuções de processos de bioinformática focados na obtenção de redes filogenéticas.

Introdução

A crescente na requisição de processos executados no LNCC criou a necessidade de alocar e/ou ajustar os programas de modo que se obtenha a melhor eficiência computacional, ou seja, minimizar o tempo e memória necessários. Além da minoração destas variáveis, a predição delas é útil tanto na prevenção de possíveis processos que extrapolem o tempo limite das execuções no Santos Dumont quanto para um maior controle que os autores dos processos terão sobre eles.

O presente trabalho intenta criar um modelo preditivo que mensure tempo e memória gastos durante processos para a obtenção de redes filogenéticas. Tal modelo atua de forma integrada com o projeto principal que visa, de fato, a obtenção das redes.

No que concerne o projeto principal, resumidamente, a cada processo, um em cinco fluxos possíveis será executado gerando no fim uma rede filogenética. O atual trabalho, portanto, atuará em cada um desses fluxos obtendo para eles um modelo inteligente que seja capaz

de prever tempo e memória utilizados baseando-se nas variáveis dos dados de entrada e os parâmetros e hiper parâmetros dos programas.

Descrição das etapas do modelo preditivo:

1. Geração dos Dados

Esta etapa consiste em obter as tabelas de cada fluxo, de modo que possuam as variáveis *versus* tempo e memória.

- Gerador Estático

Gerador utilizado de forma local, ou seja, carregam-se os dados de entrada em um diretório local, executam-se os fluxos para a obtenção das redes e inserem-se as variáveis e o tempo e memória requeridos nas tabelas.

- Gerador Dinâmico

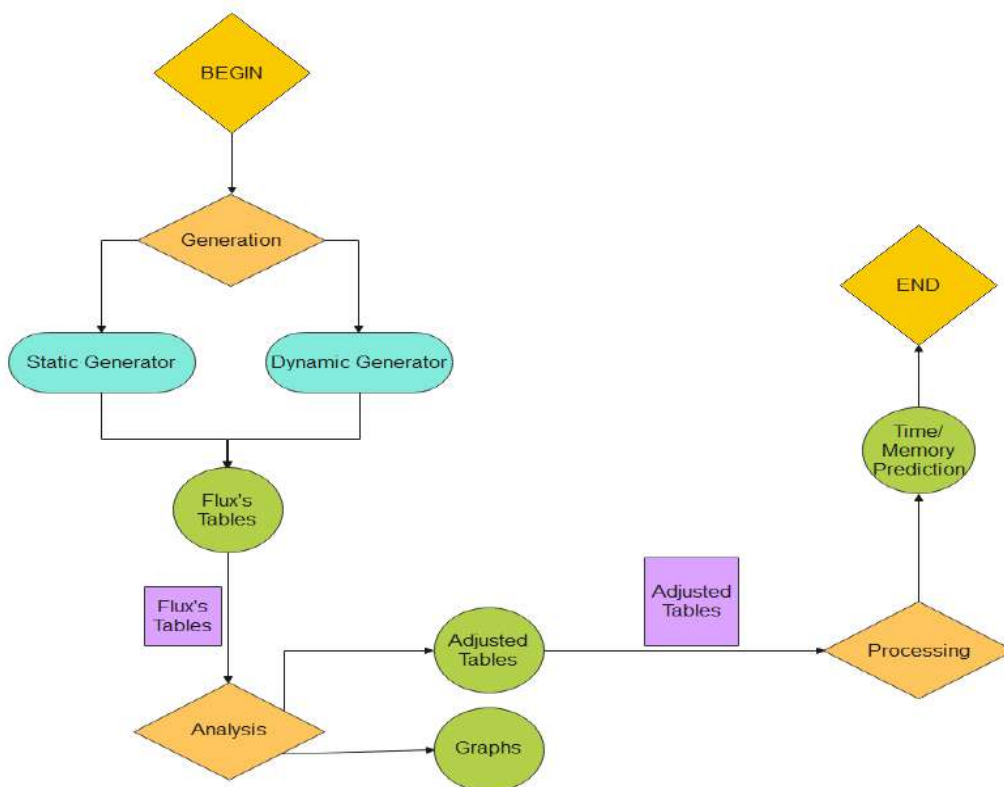
Gerador utilizado de forma integrada nas execuções no Santos Dumont. Ao fim da execução de um fluxo, as variáveis e o tempo e memória requeridos são inseridos na tabela.

2. Análise dos Dados

Neste momento as tabelas de dados criadas na primeira etapa serão analisadas, obtendo-se tabelas ajustadas e gráficos referentes às tabelas.

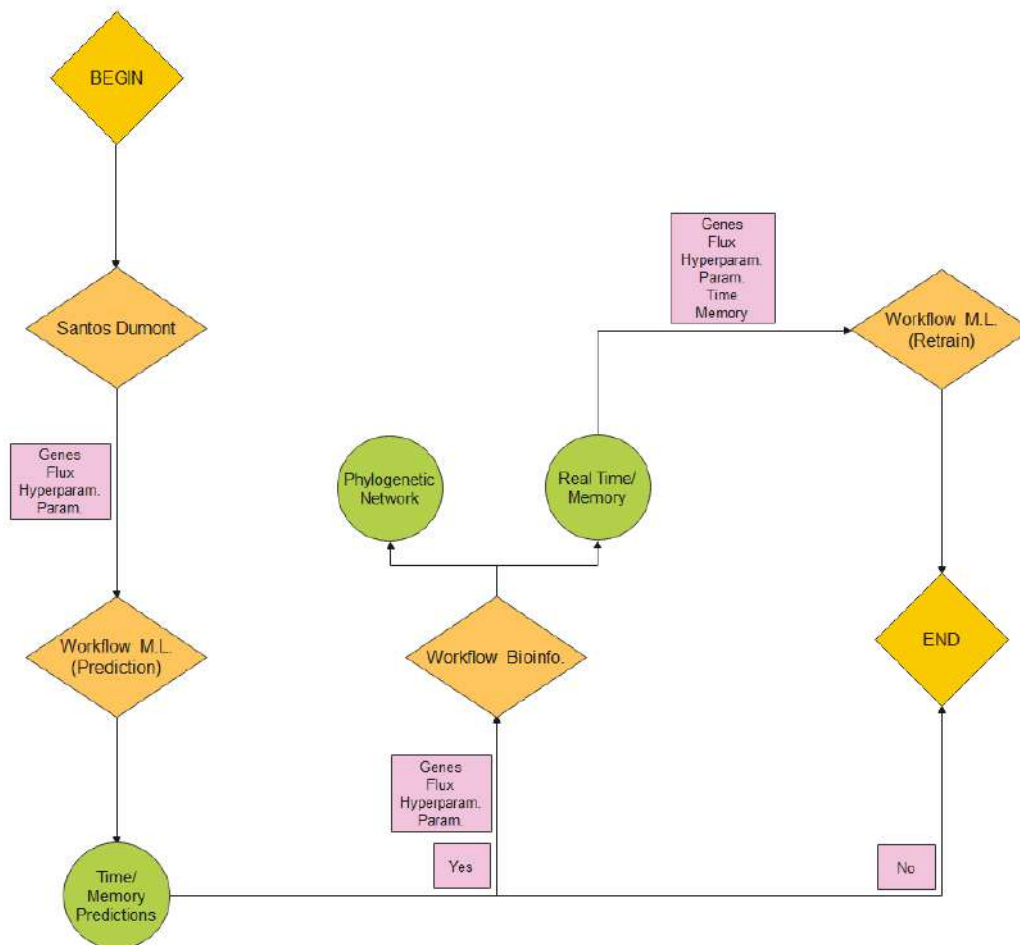
3. Processamento dos Dados

A última etapa recebe as tabelas ajustadas, faz o pré-processamento, treino, predição e avaliação das técnicas de aprendizado de máquina definidas.



Fluxo de Trabalho do Sistema de Aprendizado de Máquina

Abaixo segue o protótipo de uma possível integração entre o modelo preditivo e o projeto principal, executados no Santos Dumont.



Fluxo de Trabalho Integrado

Metodologia

Todo o procedimento foi feito de modo local, sendo assim, o gerador estático foi utilizado. O fluxo analisado foi o fluxo de número um. Seguem-se as etapas:

1. Definição das variáveis do fluxo número um:
 - Variáveis dos Dados (genes de entrada):
 - i) Número de genes (n° de arquivos *seqgen.nex*).
 - ii) Número de táxons (n° de sequências).
 - iii) Número de nucleotídeos (n° de caracteres de cada sequência).
 - Variáveis dos Programas:
 - i) Hiper parâmetros:
 - a) Número de *bootstraps* do *Astral* e *Raxml* (softwares do fluxo 1).
 - b) Número de *Threads*.
 - ii) Parâmetros:
 - a) Número de hibridizações.

b) Número de execuções.

2. Geração das tabelas com as variáveis *versus* tempo e memória, utilizando o gerador estático:
stat_gen.py → *table_flux_1.csv*
3. Análise:
analysis.py → *graph_i_flux_1.png* + *table_adj_flux_1.csv*
4. Processamento:
processing.py → *prediction* (terminal)

Resultados e Discussão

Aplicação da metodologia já descrita em termos práticos:

1° Diretórios do modelo preditivo.

✓ 1 - generation	09/08/2021 20:10	Pasta de arquivos
✓ 2 - analysis	11/08/2021 08:50	Pasta de arquivos
✓ 3 - processing	11/08/2021 09:57	Pasta de arquivos

2° Geração dos Dados

- Diretório “1 – Generation” com os dois geradores (estático e dinâmico) e as tabelas de cada fluxo.

dynamic generator	08/08/2021 10:19	Pasta de arquivos	
static generator	09/08/2021 20:10	Pasta de arquivos	
table_flux_1.csv	09/08/2021 19:51	Arquivo de Valore...	7 KB
table_flux_2.csv	09/08/2021 20:09	Arquivo de Valore...	7 KB
table_flux_3.csv	09/08/2021 20:09	Arquivo de Valore...	7 KB
table_flux_4.csv	09/08/2021 20:09	Arquivo de Valore...	7 KB
table_flux_5.csv	09/08/2021 20:10	Arquivo de Valore...	0 KB

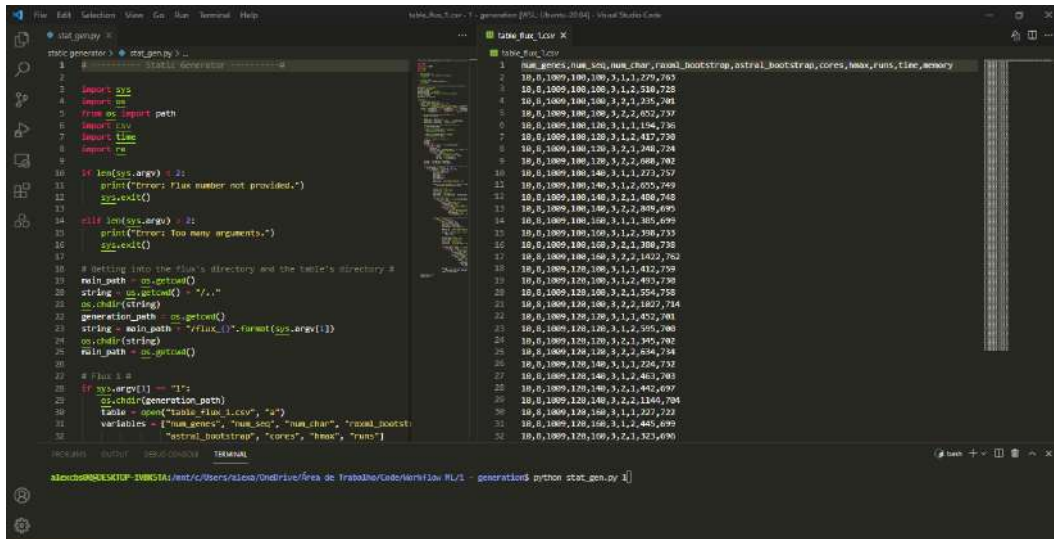
- Diretório “static generator” com os cinco fluxos possíveis e o programa do gerador.

flux_1	09/08/2021 14:16	Pasta de arquivos	
flux_2	04/07/2021 18:55	Pasta de arquivos	
flux_3	04/07/2021 18:55	Pasta de arquivos	
flux_4	05/08/2021 14:19	Pasta de arquivos	
flux_5	05/08/2021 14:19	Pasta de arquivos	
stat_gen.py	09/08/2021 19:50	Arquivo Fonte Pyt...	7 KB

- Diretório “flux_1” com os quatro conjuntos de dados utilizados durante o procedimento de geração.

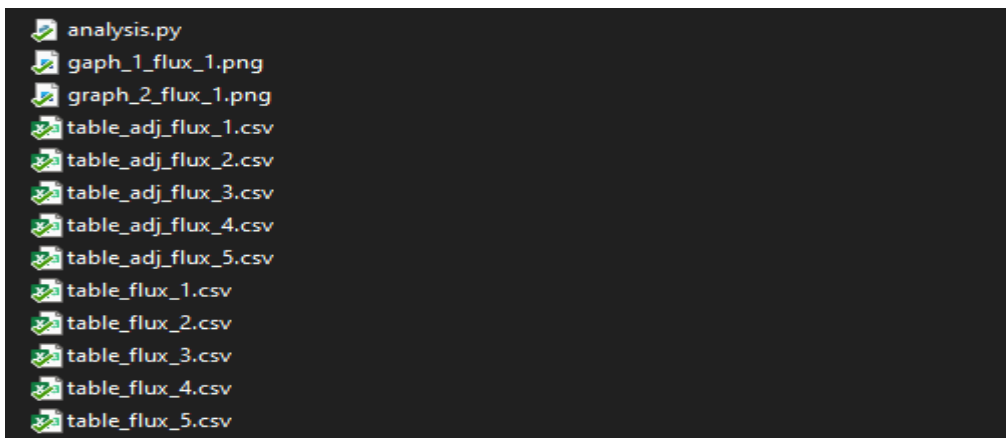
✓ denv_1	05/08/2021 15:18	Pasta de arquivos
✓ denv_2	09/08/2021 19:41	Pasta de arquivos
✓ denv_3	09/08/2021 19:41	Pasta de arquivos
✓ denv_4	09/08/2021 19:41	Pasta de arquivos

- À esquerda, o programa do gerador estático, à direita a tabela criada para o fluxo 1. No terminal encontra-se o formato da declaração para a execução do programa, passando como argumento o valor 1 (que representa o fluxo).

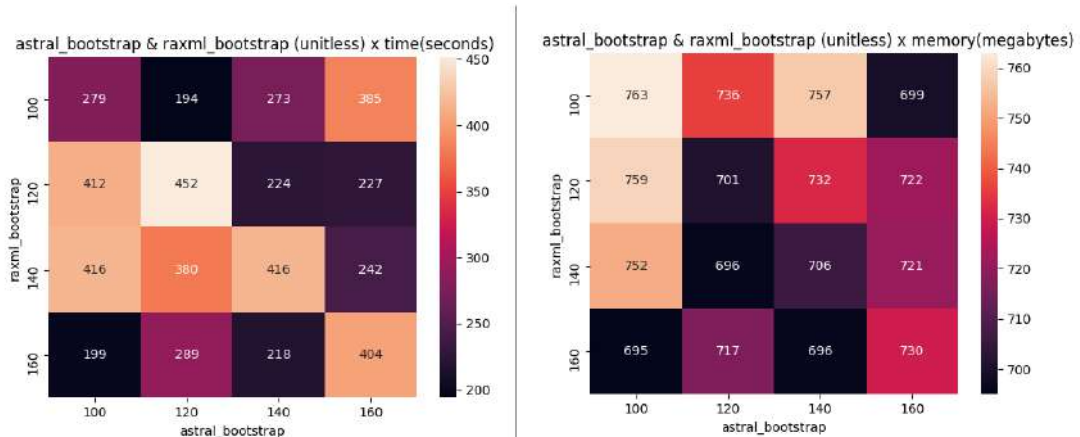


3º Análise dos dados

- Diretório “2 – analysis” com as cinco tabelas criadas no processo anterior, as cinco tabelas ajustadas, dois gráficos referentes ao fluxo 1 e o programa de análise.



- Gráficos para o fluxo 1.



4º Processamento dos Dados

- Diretório “3 – processing” com as tabelas ajustadas criadas na etapa de análise e o programa que executa o processamento.

processing.py	11/08/2021 11:44	Arquivo Fonte Pyt...	4 KB
table_adj_flux_1.csv	11/08/2021 11:44	Arquivo de Valore...	7 KB
table_adj_flux_2.csv	09/08/2021 20:09	Arquivo de Valore...	7 KB
table_adj_flux_3.csv	09/08/2021 20:09	Arquivo de Valore...	7 KB
table_adj_flux_4.csv	09/08/2021 20:09	Arquivo de Valore...	7 KB
table_adj_flux_5.csv	09/08/2021 20:10	Arquivo de Valore...	0 KB

- Programa sendo executado utilizando regressão linear nos dados referentes ao fluxo 1. No terminal encontra-se o formato da declaração para a execução do programa, passando como argumentos: o valor do fluxo (terceira palavra da esquerda para a direita), seguido por oito valores que fazem referência às variáveis já mencionadas. Como saída, tem-se a predição do tempo e a memória para este conjunto de variáveis.

```
processing.py
25 from sklearn.base import BaseEstimator, TransformerMixin
26 from sklearn.compose import ColumnTransformer
27
28 # Model
29 from sklearn import linear_model
30 from sklearn.linear_model import LinearRegression
31 from sklearn.linear_model import LogisticRegression
32
33 from sklearn.neighbors import KNeighborsClassifier
34 from sklearn.ensemble import RandomForestClassifier
35 from sklearn.tree import DecisionTreeClassifier
36 from sklearn.model_selection import KFold
37
38 # .....
39 # Initializing #
40 # Getting the flux number
41 if len(sys.argv) < 2:
42     print("Error: Flux number not provided.")
43     sys.exit()
44
45 table_name = "table_adj_flux_{}".format(sys.argv[1])
46
47 # Loading the current flux's tables.
48 main_path = os.getcwd()
49 os.chdir(main_path + "/" + "/" + "/2 - analysis")

```

alexsb@BROOKSHP10P-13W85TA:~/mnt/c:/Users/Alexa/OneDrive/Trabalho/Code/stackflow/ML/3 - processing\$ python processing.py 1 10 8 1000 100 100 3 1 1

Flux 1
Variables values: [10, 8, 1000, 100, 100, 3, 1, 1]
Time Prediction: 415.410ec
Memory Prediction: 726.899c

Analisando o processo como um todo foi possível verificar os seguintes pontos:

1. A tabela gerada para o fluxo 1 teve 193 linhas,
2. É necessário obter mais gráficos que analisem a dependência parcial do tempo e memória em função de uma variável e assim ter uma melhor avaliação da sua influência.
3. O gerador dinâmico ainda não é funcional.
4. A predição utilizou um modelo simples de regressão linear. A tabela com apenas 193 linhas restringiu significativamente a acurácia do método.

Conclusões

Portanto, o objetivo da criação de um modelo teórico estruturado, conciso e escalável capaz de fazer previsões de tempo e memória foi alcançado. No entanto, é evidente a necessidade de ajustes e implementações práticas, principalmente no que diz respeito a obtenção de um grande volume de dados para as tabelas dos fluxos e a aplicação de diferentes técnicas de aprendizado de máquina na etapa de processamento.

Referências Bibliográficas

Relatório de Atividades

1 Dados Gerais

Título do projeto: Sistema Embarcado para robô Humanoide

Nome do Bolsista: Ana Luiza Buse da Silva

Nome do orientador: Jauvane Cavalcante de Oliveira (LNCC) e Erick Menezes Moreira (IME)

Tipo de bolsa e período do relatório: PIBITI–CNPq/LNCC, Edital 2020/2021, 10/2020 - 08/2021

2 Objetivos

O objetivo desse projeto é o desenvolvimento prático de um sistema eletrônico embarcado para um robô humanoide, visando competir na Competição Brasileira de Robótica (CBR) [1], na modalidade *Humanoid Robot Racing* (HRR), proposta pela IEEE (Instituto de Engenheiros Eletricistas e Eletrônicos) [2]. O principal propósito da competição HRR é motivar as pesquisas para melhorar os movimentos de andar e correr de um robô bípede, desenvolvendo coordenação e a fluidez do movimento para manter o equilíbrio ao realizar uma determinada tarefa, a fim de desenvolver um melhor robô humanoide para interações humanas no futuro.

Além disso, o robô foi construído em parceria com uma Dissertação de Doutorado [3], desenvolvido no Laboratório de Robótica e Inteligência Computacional (RoboIME) do Instituto Militar de Engenharia (IME).

3 Introdução

Robôs autônomos são um tipo de robô que podem cumprir tarefas em ambientes estruturados ou não sem que necessite de interferência ou ajuda humana, a partir de uma estrutura computacional previamente programada. Para essa finalidade, o robô necessita de ferramentas lógicas e componentes eletrônicos embarcados que possam permitir um software que realize um controle autônomo e um hardware com sensores que possam interagir com o ambiente.

O software programado necessita de uma unidade de processamento e uma fonte de energia para alimentação. Para o robô captar dados externos, serão necessários sensores, cujas informações serão processadas para auxiliar o robô a executar tarefas, como andar em linha reta, desviar de obstáculos, processar imagens, dentre outros.

No intuito de incentivar o avanço científico para o desenvolvimento na área de robótica, a CBR possui uma competição na área de robô humanoide autônomo. Para desenvolver completamente esse modelo de robô, o projeto foi dividido nas áreas de mecânica, inteligência, mecatrônica e eletrônica.

Por fim, este subprojeto foca na confecção de um sistema eletrônico a ser embarcado no robô autônomo, cujo projeto pertence ao time de robótica do Instituto Militar de Engenharia (RoboIME).

4 Metodologia

Para o desenvolvimento e continuidade do projeto, foram necessários os estudos e aprofundamentos em alguns conceitos eletrônicos, que são explicitados abaixo. Nessa seção também são descritos os componentes escolhidos para o projeto, assim como as razões para essa escolha.

4.1 Microprocessador

O Microprocessador escolhido para esse projeto é a Raspberry, que é um computador barato, portátil e versátil, usado principalmente em projetos de programação, robótica e em iniciativas em geral com software e hardware livre. Além disso, é importante citar as características que a Raspberry que são fundamentais para o bom desenvolvimento no projeto:

- WiFi integrado: A comunicação deve ser feita com wifi, sem o uso de cabos, pois o robô deve estar livre para andar sem nenhum tipo de apoio físico do mundo externo.
- Interface CSI (câmera): É necessário uma interface de visão computacional, pois a correção da caminhada e desvio de obstáculos é feita usando-se esse recurso.
- Processador : Várias coisas devem processadas ao mesmo tempo, como o algoritmo de caminhada, a visão computacional e a tomada de decisões, assim necessita-se de alto poder e velocidade de processamento.

4.2 Motores

As juntas do robô humanoide são feitas com os servos Dynamixel AX-12A, que são a versão básica de servos da empresa Robotis inc. Eles são usados por possuírem algumas características essenciais para o robô:

- Relata posição, velocidade, carga, tensão e temperatura;
- Alta precisão em posição angular (resolução de 0.29°);
- Sistema de proteção de desligamento automático quando em tensão carga máxima ou temperatura limite, que impede danos ao motor durante a caminhada.
- Todo o gerenciamento do sensor e controle de posição são controlados pelo microcontrolador integrado do servo (ATMega 8), não exigindo construção de um sistema de controle pelo usuário.

Foram usados 8 motores AX-12A para cada junta das pernas, enquanto para o tronco, braços e pescoço foram usados 4 Servos Motores 9g, para diminuir o peso do robô. Isso pois essas juntas não foram usadas para o projeto, já que não havia a necessidade de levantar o robô em caso de uma queda, assim não foi preciso usar motores com grande torque, optando-se por motores leves que facilitam a movimentação.

4.2.1 Microcontroladores

Microcontrolador é um pequeno computador num único circuito integrado o qual contém um núcleo de processador, memória e periféricos programáveis de entrada e saída. Eles são concebidos para aplicações embarcadas, com baixo consumo de energia.

Para este projeto, foi escolhido a MyRIO: um dispositivo embarcado da National Instruments. Isso porque ela é recomendada para aplicações que utilizam FPGA e microprocessador integrados, que é justamente o que acontece nesse projeto. Além disso, ela é programada em LabVIEW, uma linguagem gráfica muito utilizada em automação. [4]

A adição da MyRIO é justificada porque foi percebida a necessidade de se ter um microcontrolador embarcado no circuito para a conexão com os motores, isso por causa do fato de a Raspberry só ter uma porta Tx e uma Rx para controlar todos os motores. Sem o microcontrolador, os motores eram comandados diretamente pela Raspberry, de forma sequencial, e não ao mesmo tempo, tirando a fluidez necessária para o movimento. Além disso, ela torna mais possível o uso da visão do robô, que é feita pela Raspberry, pois o controle dos motores ocupava grande parte do poder de processamento da Raspberry, o que dificultava o uso da visão. Ela possui um processador Zynq real-time com FPGA integrado, programado em LabVIEW, que será usado da seguinte forma:

- Real-time: faz a comunicação com o Raspberry Pi e com o computador receptor da telemetria. Tem por objetivo receber a tabela de resultados da simulação mecatrônica e comandar de maneira sincronizada os movimentos dos motores.

- FPGA: tem a possibilidade de ser programado para comunicar-se em quase todos os protocolos existentes com até 36 periféricos. Apesar da Myrio ter apenas 2 linhas de UART, é possível, através do FPGA, configurar para as pinos IO serem operantes como linhas UART. Foi feita uma programação para que ele tenha um número de portas Uart Half-duplex igual ao número de motores AX-12A existentes no robô. Assim, os comandos de posição e velocidade podem ser enviados a todos os motores ao mesmo tempo, evitando-se atrasos causados por uma linha de comunicação compartilhada.

5 Resultados e Discussão

Para este projeto foi desenvolvido uma biblioteca em LabVIEW para MyRIO capaz de enviar comandos para os motores AX12-A de forma automática de acordo com o ângulo e a velocidade requeridos. Além disso, foi feita uma PCB a ser embarcada no robô, e a partir disso foram feitos testes e ajustes. Por fim, o robô competiu na maior competição de robótica da América Latina, a *Latin American Robotics Competition*, onde obteve o 3º lugar na modalidade Humanoid Robot Racing.

5.1 Desenvolvimento da biblioteca do AX12-A em LabVIEW para a MyRIO

Um passo do robô é definido pela inteligência, usando algoritmos de inteligência artificial, sendo os valores dos ângulos e velocidades de cada motor armazenados em um arquivo CSV. Como os motores se comunicam com comandos de forma bem específicas, foi desenvolvido uma biblioteca em LabVIEW para interpretar esses arquivos e enviar os comandos para os motores AX-12A. O funcionamento dessa biblioteca é esquematizado pelo fluxograma da figura 1, onde é possível observar todas as etapas do código.

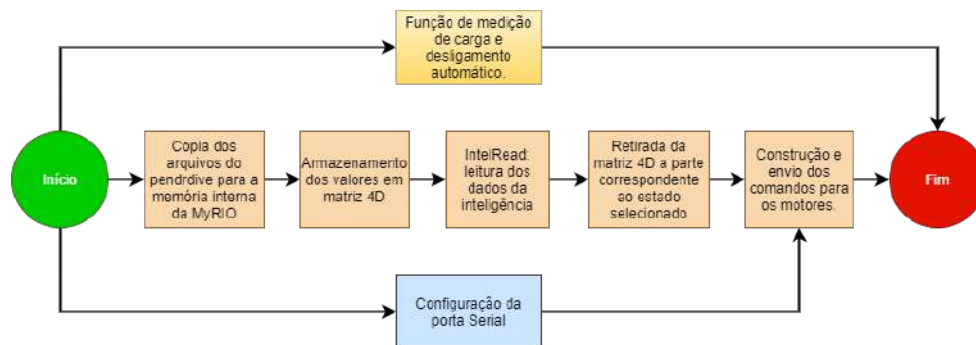


Figura 1: Fluxograma do funcionamento do código.

Como há vários arquivos arquivos CSV, um para cada tipo de movimento, todos esses arquivos são copiados do pendrive conectado na MyRIO para a memória interna dela, de forma que esse pendrive possa ser retirado para a minimização do peso. Esse passo pode ser visto no código na figura 2.

Esse arquivo na memória da MyRIO é lido e seus valores são armazenados em uma matriz. Isso é feito conforme o código do lado esquerdo da figura 3. Após isso, a Raspberry recebe da inteligência qual o estado deve ser executado. Isso é feito pela função IntelRead do lado direito da figura 3, que verifica a existência de dados nesse canal e os recebe.

A partir disso é possível então, pode-se retirar da matriz 4D a parte correspondente do estado selecionado, resultando uma matriz 2D, que é enviada para uma função de construção dos comando. Ela pega a matriz do estado e constrói um vetor com a forma certa dos ângulos e velocidade para o movimento de cada motor. Posteriormente é acrescido alguns parâmetros a esse vetor, inclusive com Cheksum.

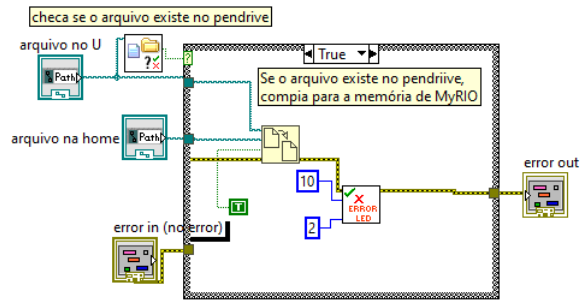


Figura 2: Verificação e cópia dos arquivos do pendrive.

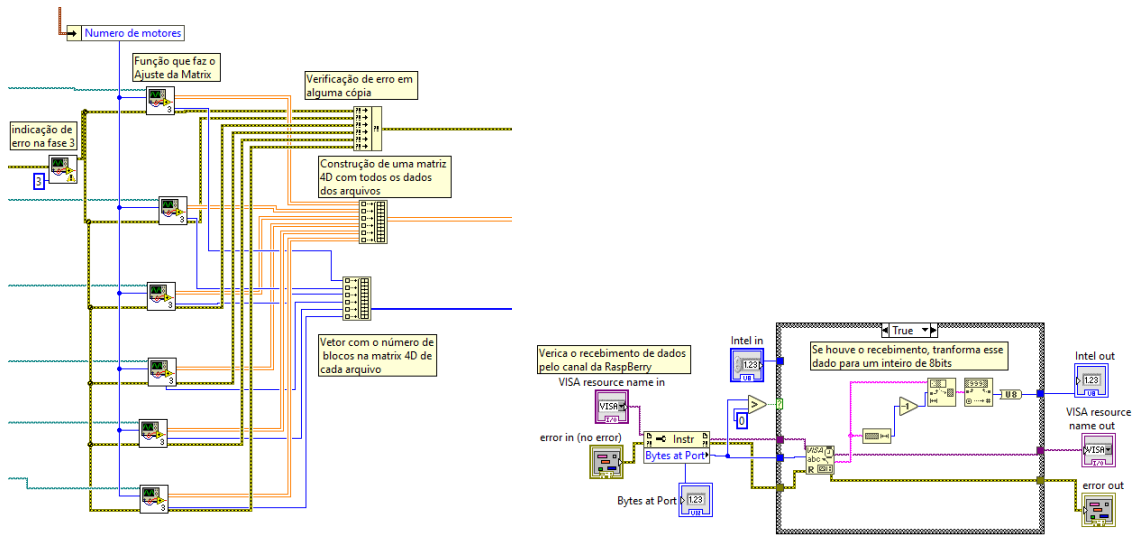


Figura 3: Armazenamento dos dados em uma matriz 4D e função IntelRead.

Entretanto, para enviar esses comandos para motor, deve-se configurar o FPGA para que cada porta Serial da MyRIO atue com o protocolo UART Half Duplex.

Além disso, foi feita uma função de verificar o nível da bateria, para caso a bateria fique com menos de 10,5V a função atuar desligando o robô, a fim de proteção dos componentes, tanto da própria bateria quanto dos motores e da PCB. Essa função foi construída de acordo com a figura 4.

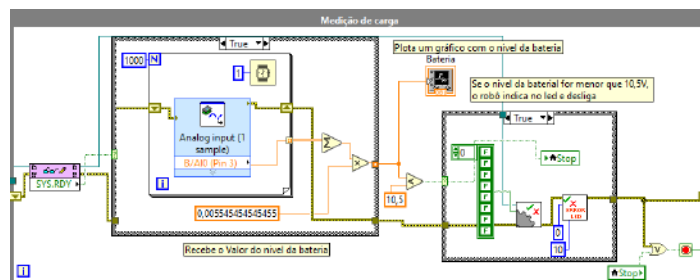


Figura 4: Função de medição de carga e desligamento automático.

5.2 Montagem da placa

Para todos os componentes serem embarcados no robô, é necessário uma placa de circuito impresso de forma a integrar esses componentes com a bateria e para a transmissão de dados. Assim, foi desenvolvido uma PCB utilizando o software Altium Designer. Na figura 5 é mostrado o esquemático dessa placa, mostrando seus principais componentes.

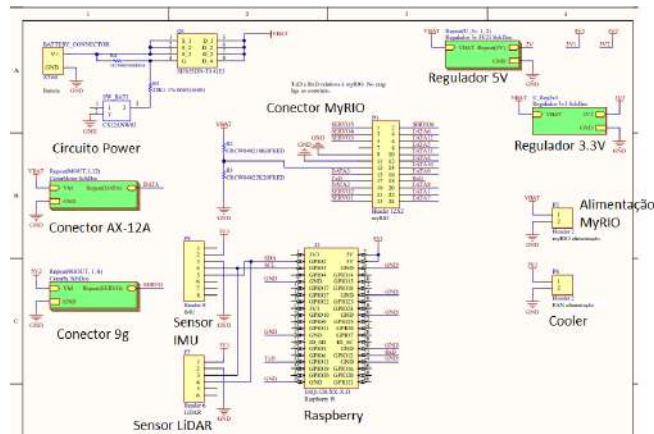


Figura 5: Esquemático da placa eletrônica.

A eletrônica foi montada no robô como na figura 6, onde a Raspberry é fixada em cima da placa e a MyRIO fica nas costas do robô. A seguir descreve-se o uso de cada componente.

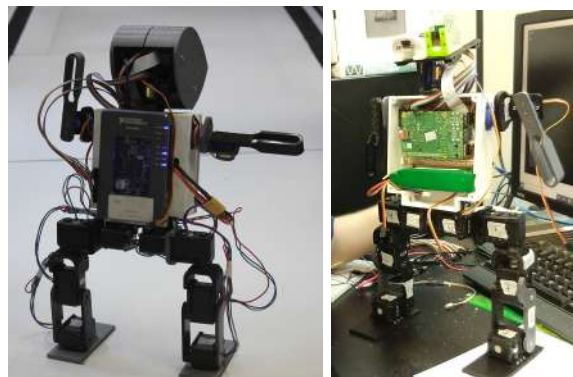


Figura 6: Visão 3D da placa.

- Circuito Power: Formado por um MosFET e uma chave. A chave é usada para ligar e desligar o robô.
- Conector AX-12A: São colocados 12 conectores para o AX-12A, com a função de fornecimento de energia para esses motores diretamente da bateria. Há uma trilha de transmissão de dados de cada conector para cada pino da MyRIO.
- Conector motor 9g: Usa-se 4 motores 9g na cabeça e braços. A trilha de dados é conectada na MyRIO.
- Conector MyRIO: Usado para a passagem de dados dos motores para a MyRIO.
- Conector Raspberry: Para encaixe da mesma na placa e passagem de dados dos sensores e motores 9g.
- Header sensor IMU: Header de 90° soldado na placa de modo que os fios saem do espaço entre a PCB e a Raspberry e são conectados na cabeça do robô.
- Header sensor Lidar: Semelhante ao anterior, porém é um sensor de 5V.
- Regulador de tensão 5V: Necessário para alimentação da Raspberry, dos motores 9G e do sensor Lidar.

- Regulador de tensão 3.3V: Necessário para a alimentação do sensor IMU.
- Header de alimentação do Cooler: Foi anexado esse conector caso houvesse a necessidade de um Cooler para resfriamento da Raspberry, mas não foi verificado aquecimento no sistema.

Com o esquemático e o tamanho da placa, faz-se o design 3D da placa, como mostrado na figura 7, posicionando os componentes da melhor forma possível na placa, adotando medidas como a construção do plano de terra com furos de passagem, posicionamento próximos de componentes semelhantes, atentando-se para a largura das trilhas de acordo com a corrente esperada e a distância entre as trilhas, de forma a minimizar a interferência entre trilhas. Também é importante construir trilhas com formatos arredondados, e não com ângulos retos, para evitar a reflexão do sinal. [5]

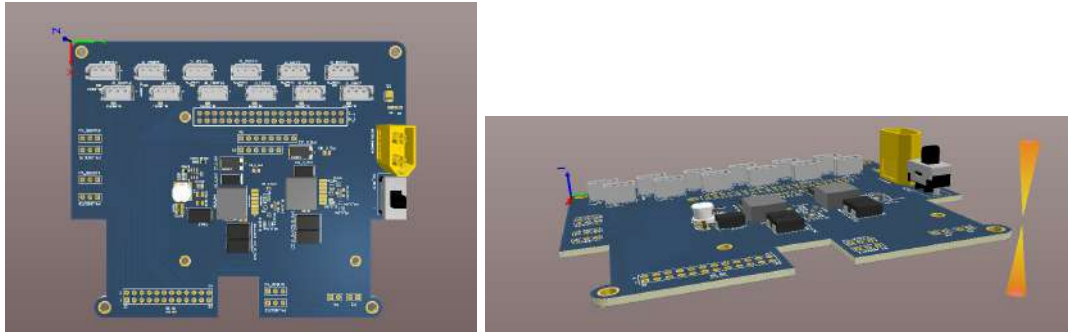


Figura 7: Visão 3D da placa.

6 Conclusões

O desenvolvimento de uma eletrônica para ser embarcada em um robô humanoide autônomo constitui um desafio, ainda mais pelo fato de ser de uma forma inovadora, sem o uso de plataformas prontas. Foram encontradas várias dificuldades, o que gerou um grande aprendizado na linha de pesquisa e desenvolvimento. Dessa forma, este projeto agregou tanto no sentido acadêmico quanto em muita experiência prática laboratorial. A figura 7 mostra o resultado do projeto eletrônico, mecânico e computacional desenvolvido em conjunto com a RoboIME no robô que participou da LARC 2020, obtendo o 3º lugar da categoria ao correr 2,80m em 3 minutos. [6]

Referências

- [1] “Cbr-competição brasileira de robótica,” <http://www.cbrobotica.org/index.php/categorias/>, accessed: 2021-08-05.
- [2] “Ieee,” <http://www.ieee.org.br/>, accessed: 2021-08-05.
- [3] F. S. CHAGAS, “Walking algorithms for robots inspired by human gait and on inverse kinematics combined with evolutionary computation,” 2015.
- [4] “myrio - dispositivo para projetos embarcados de estudantes,” <https://www.ni.com/pt-br/shop/hardware/products/myrio-student-embedded-device.html>, accessed: 2021-08-05.
- [5] “10 mandamentos da pcb,” <https://www.embarcados.com.br/10-mandamentos-da-pcb/>, accessed: 2021-08-05.
- [6] “Tdp - roboime-humanoide (iee hrr) - larc 2020.”

JAUVANE CAVALCANTE DE OLIVEIRA
Orientador

**MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO MILITAR DE ENGENHARIA
LABORATÓRIO NACIONAL DE COMPUTAÇÃO CIENTÍFICA**

ARTUR SANTIAGO DE OLIVEIRA MENESES

**DESENVOLVIMENTO DE ALGORITMO PARA ESTABELECIMENTO
DE REDE AD HOC USANDO AERONAVES REMOTAMENTE
PILOTADAS DOTADAS DE ANTENAS REPETIDORAS
(Subprojeto de pesquisa submetido ao PIBITI-CNPq/IME)**

**Orientadores: Jauvane Cavalcante de Oliveira (LNCC), Erick Menezes
Moreira (IME)**

**Projeto: Sistemas robóticos inteligentes para tarefas cooperativas em
aplicações de Defesa**

**RIO DE JANEIRO
2021**

Sumário

1	Dados Gerais	2
2	Objetivos	2
3	Introdução	2
4	Metodologia	3
5	Resultados e discussão	3
6	Conclusão	6
7	Referências Bibliográficas	6

1 Dados Gerais

Título do projeto: Desenvolvimento de algoritmo para estabelecimento de rede ad hoc usando aeronaves remotamente pilotadas dotadas de antenas repetidoras.

Nome do bolsista: Artur Santiago de Oliveira Meneses

Nome dos orientadores: Jauvane Cavalcante de Oliveira (LNCC), Erick Menezes Moreira (IME)

Tipo de bolsa: PIBITI, novembro de 2020 - agosto 2021

2 Objetivos

O presente projeto possui por objetivo último o desenvolvimento de um algoritmo, o qual possibilite uma estimativa otimizada sobre o melhor posicionamento de uma aeronave remotamente pilotada, munida de antena repetidora, em um sistema de redes de comunicação móveis ad-hoc.

O funcionamento do algoritmo deverá se basear nas posições de usuários da rede, e apresentar critério de escolha condicional para melhor posição de emissão de sinais, levando em consideração as características geográficas da região, a demanda de uso da rede e a densidade local de usuários por setor de atuação.

O projeto também possui como objetivo a simulação de protocolos ad-hoc e de outras funcionalidades como maneira de propor melhorias em códigos já existentes que modelem esse tipo de rede de comunicação e reconhecer algumas limitações de simuladores enquanto se revisam conceitos fundamentais do assunto.

3 Introdução

A implantação em maior escala de redes móveis ad-hoc constituem uma interessante alternativa para o fornecimento de dados e redes de telefonia em áreas de difícil acesso ou de implantação problemática, devido à sua versatilidade e adaptabilidade em conjunturas instáveis, sendo ideal também para o fornecimento instantâneo de redes com aplicações civis e militares [4].

Nesse sentido, redes móveis gerenciadas por aeronaves não tripuladas apresentam vantagem na atuação emergencial em contextos de desastres naturais, cenários de guerra e em áreas rurais ou carentes [1]. Tais possibilidades são potencializadas pela velocidade de resposta e pela mobilidade de uma aeronave não tripulada, a qual pode se deslocar sob demanda para potencializar o número de usuários atendidos pela rede, e ajustar seu voo sob para obter melhores condições de canal entre a antena e o objeto alvo [4]. Esta mobilidade pode ser usada, por exemplo, em zonas de poluição, terrenos complexos ou inacessíveis e em condições atmosféricas desfavoráveis [3].

Portanto, a implantação e a pesquisa desta modalidade de fornecimento de rede se constituem numa importante e versátil alternativa a ser utilizada em determinadas situações, caracterizadas pela difícil implantação de sistemas de comunicação tradicionais.

Além disso, faz-se relevante nesse contexto o aprimoramento de novos modelos computacionais, com finalidade de minimizar e superar problemas ou limitações existentes quanto ao uso desta modalidade de rede e quanto à utilização de aeronaves remo-

tamente pilotadas, com ênfase nos efeitos nos canais requeridos durante a aplicação do sistema.

4 Metodologia

A principal metodologia no início do projeto foi a revisão de aspectos fundamentais sobre redes móveis ad-hoc. Em seguida, a ferramenta mais pertinente ao trabalho foi a plataforma de eventos discretos OMNeT++ e algumas de suas capacidades pré-disponíveis, a fim de se reconhecer a plataforma e suas aplicações, bem como analisar suas potencialidades e limitações por meio da implantação de alguns protocolos específicos.

5 Resultados e discussão

Foram aplicados alguns modelos de mobilidade e protocolos como o AODV (devido à sua natureza reativa, a fim de simular a adaptabilidade de uma frota de drone, e capacidade acomodação de grandes números de nós) em módulos móveis, inseridos em modelos reais de cenários simplificados, com edição de algumas características de atenuação, perda e interferência, por exemplo. Nesse sentido, embasados pela comparação quanto à funcionalidade com outras linhas de simuladores, como o SUMO e o Veins, principalmente quanto à modelagem de cenário, buscou-se a realização de simulações para detecção de algumas limitações dos simuladores, a fim de nortear o desenvolvimento de novas funcionalidades ou o aperfeiçoamento de outras.

Nesse sentido, o principal ponto destacado foi a necessidade de aperfeiçoamento das ferramentas de modelagem de canal para melhorar a proximidade a situações de uso real, o que norteará as próximas etapas do projeto. Além disso, também foi identificada necessidade de aprimoramento da variedade e protocolos ad-hoc disponíveis na plataforma, a fim de desenvolver seus modelos de maneira análoga ao AODV pré-disponibilizado, tanto para diversificar as aplicações do simulador quanto para exercer atividades comparativas entre protocolos em situações específicas.

A simulação final obtida, a qual ainda apresenta finalidades de melhoramento, foi configurada com base no protocolo UDP da camada de transporte, o qual embora apresente reduzida confiabilidade, possui velocidade pertinente à simulação com veículos remotamente pilotados. Este protocolo funcionou em conjunto com o algoritmo de IPv4 da camada de rede para minimizar a latência da rede, por meio A fim de compensar eventuais erros de transmissão, na camada de enlace foi aplicado o CSMA/CA com uso de acknowledgments, a fim de reduzir colisão de sinais e também, por meio de um critério limite do número de retransmissões realizadas, auxiliar o protocolo AODV a reagir eficientemente à indisponibilidade de certos nós pelos mais diversos fatores físicos (obstáculos, alcance de transmissão).

O alcance dos sinais de cada módulo também foram estabelecidos a fim de aumentar a verossimilhança desta e de futuras simulações, assim como a interferência entre mensagens, por meio da análise de chegada de mensagens simultâneas em um mesmo nó de rede. Alguns obstáculos foram acrescentados ao cenário e configurados por meio de um arquivo .xml, com dimensões, material, posição e características físicas especificadas

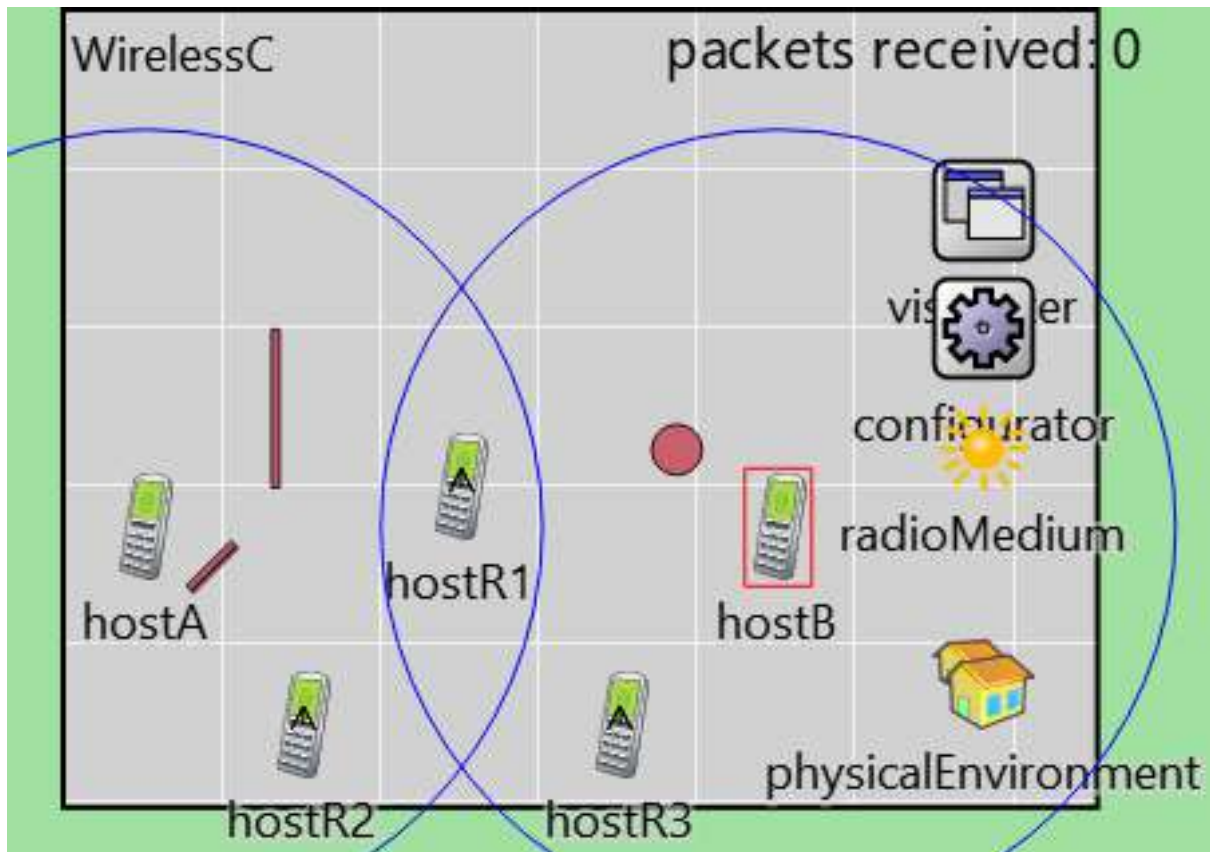


Figura 1: Imagem da rede

neste. Neste caso, um fator de aprimoramento a ser realizado na simulação é a atenuação nos objetos, pois os mesmos bloqueiam completamente quaisquer mensagens que eventualmente colidam com eles.

O rádio utilizado nas simulações foi regulado por default para utilizar a modulação BPSK, como evolução de um modelo de rádio ideal, sem modulação específica, que garantia a transmissão e recepção de quaisquer sinais enviados. O rádio BPSK também teve parâmetros como consumo de energia, largura de banda, frequência central e características como sensibilidade e limites mínimos de energia de sinal para detecção configurados. Neste ambiente, apenas a mobilidade do tipo linear foi utilizada nos nós da rede, mas com intenções de aplicar movimento randômico para os veículos e modelar colisões e desvios realizados pelos drones frente aos obstáculos encontrados no meio. Por fim, embora o consumo de energia por meio dos módulos tenha sido modelado de acordo com o estado em que os mesmos se encontravam (transmitindo, recebendo, em espera), a fonte de energia dos veículos ainda foi ideal durante as simulações.

Todas as configurações foram realizadas nas extensões .ned dos arquivos .cc, com estas sendo configuradas conforme interessa no arquivo de inicialização .ini conforme objeto a ser estudado. A discussão e objetivos principais das simulações realizadas, em especial da última, consistiram na obtenção de embasamento teórico dos conceitos de comunicação envolvidos, tanto de redes quanto modelagem de aspectos físicos e de rádio, e por fim na implementação dos respectivos algoritmos no simulador para entendimento

prático, de maneira coerente com o propósito e tema do projeto, a fim de estimular o domínio de alguns desses algoritmos fornecidos pela plataforma do simulador quando usados no contexto de redes móveis ad-hoc. Para isso, foram obtidas estatísticas do envio de mensagem entre os módulos, com o protocolo AODV em funcionamento, em cenário livre de obstáculos e com obstáculo, com intuito de analisar o funcionamento do algoritmo. Percebe-se nas imagens abaixo a alternância de uso entre os módulos R1 e R2 pela rede devido aos obstáculos.

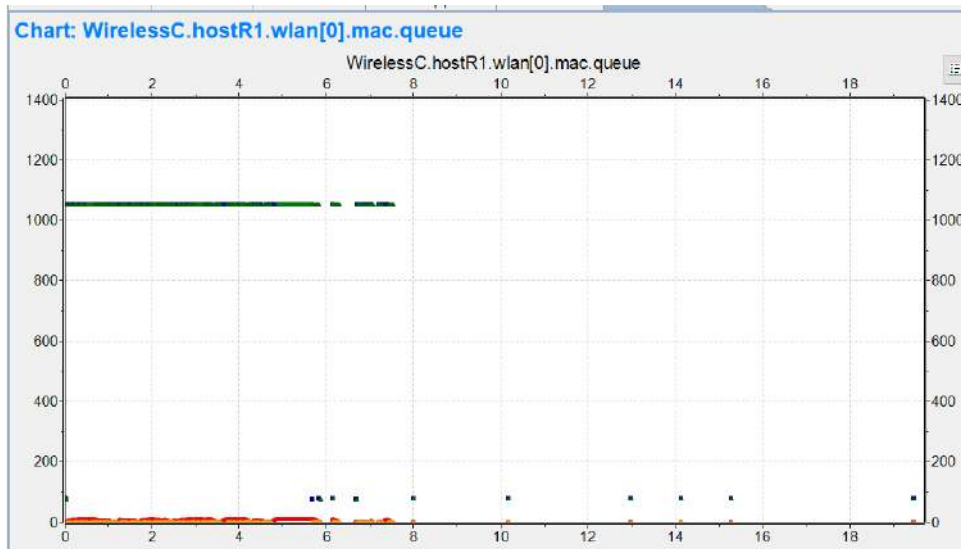


Figura 2: Queueing do módulo R1

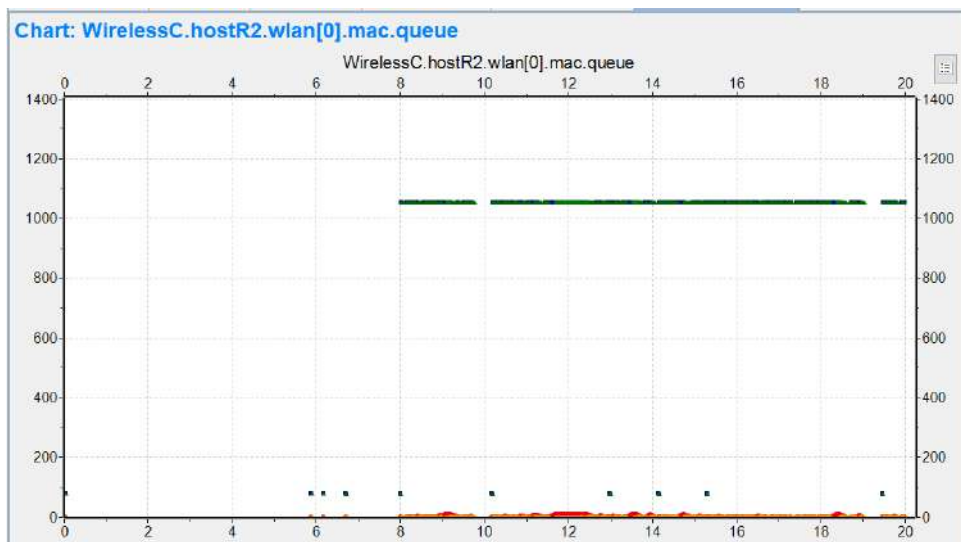


Figura 3: Queueing do módulo R2

6 Conclusão

A implantação e estudo de redes móveis ad-hoc é essencial para a diversificação e aumento da versatilidade quanto ao fornecimento de redes de comunicação. Essas características são muito desejadas à medida que são mais demandadas aplicações múltiplas das inovações tecnológicas em diversas conjuturas.

Desta maneira, o uso do simulador OMNeT++ contribui para o estudo dessa pertinente implementação, e também para a detecção de aprimoramentos quanto às ferramentas e metodologias utilizadas, a fim de promover modelos mais reais e eficientes de canal, protocolos e mobilidade dos módulos da rede. Tais melhoramentos serão essenciais à continuidade do desenvolvimento do algoritmo.

7 Referências Bibliográficas

- [1] Zhe Wang, Lingjie Duan and Rui Zhang: Adaptive Deployment for UAV-Aided Communication Networks. IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, 2019, VOL. 18, NO.9, 4531-4543.
- [2] Daniel Lihui Guy Guangyu Pei, Henry Ly, Mario Gerla, Beichuan Zhang and Xiaoyan Hong, UAV Aided Intelligent Routing for Ad-Hoc Wireless Network in Single-area Theater. 2000 IEEE WIRELESS COMMUNICATIONS AND NETWORKING CONFERENCE, Conference Record (Cat. No.00TH8540), 2000, VOL.3
- [3] Yi Zhou, Nan Cheng, Ning Lu, and Xuemin (Sherman) Shen: Multi-UAV-Aided Networks: Aerial-Ground Cooperative Vehicular Networking Architecture. IEEE VEHICULAR TECHNOLOGY MAGAZINE, 2015, VOL. 10, ISSUE 4, 36-44.
- [4] Meng Hua, Yi Wang, Zhengming Zhang, Chunguo Li, Yongming Huang and Luxi Yang. Power-Efficient Communication in UAV-Aided Wireless Sensor Networks. IEEE COMMUNICATIONS LETTERS, 2018, VOL. 22, NO. 6, 1264-1267

RELATÓRIO DE ATIVIDADES

Título do Projeto: Exploração de Módulos Paralelo-Híbrido de Bioinformática para Ambientes GPU de Supercomputação

Nome do bolsista:

Guilherme Freire da Silva Dornelas

Nome do orientador:

D.Sc. Carla Osthoff Ferreira de Barros (Tecnologista Sênior – CENAPAD/LNCC)

Nome do coorientador:

D.Sc. Kary Ann del Carmen Ocaña Gautherot (Tecnologista Pleno – LABINFO/LNCC)

Tipo de bolsa: PIBIC

Período do relatório: 01/08/2020 a 31/07/2021

OBJETIVOS

A presente proposta visa analisar o desempenho de aplicações CPU e GPU de bioinformática em ambientes de processamento de alto desempenho.

O projeto apresenta três principais frentes:

- Migrar e acoplar os programas Beast, Beast2 para o SDumont.
- Explorar esses programas quando acoplados à biblioteca de alto desempenho BEAGLE 3 nos ambientes CPU e GPU do SDumont.
- Realizar análises de desempenho e escalabilidade, visando otimizar o uso do ambiente computacional.

INTRODUÇÃO

Plataformas de Computação de Alto Desempenho (CAD) como clusters, grades, nuvens ou supercomputadores permitem executar de uma maneira eficiente tarefas com grande demanda de processamento. O supercomputador Santos Dumont (SDumont) possui uma arquitetura híbrida com CPU multi-core e dispositivos com a chamada arquitetura many-core, GPU e MIC. SDumont é usado de forma intensiva por vários grupos de pesquisa brasileiros das mais diversificadas áreas científicas. Fruto da colaboração entre os centros de pesquisa CENAPAD e LABINFO pertencentes ao LNCC são realizadas diversas pesquisas no apoio às análises computacionais envolvendo bioinformática, biologia computacional e análises de CAD.

O presente trabalho visa levantar um estudo de desempenho de aplicações de filogenômica e evolução molecular computacional em multi-GPU. Dessa maneira, apresenta uma análise comparativa do software BEAST/BEAGLE executado em 1 nó com CPU multi-core e 1 GPU do SDumont e usando dados biológicos de diferente natureza. No presente trabalho apresentamos resultados de desempenho escalando BEAGLE/BEAST até 8 GPU, os quais se mostram mais eficientes.

O uso de um ou mais GPU pode se mostrar mais eficiente do que o uso de CPU multi-core para analisar grandes conjuntos de dados de nucleotídeos. Para dados de entradas com alinhamento de várias partições como no vírus Dengue é altamente recomendável dividir os cálculos de probabilidade em vários dispositivos GPU.

METODOLOGIA

BEAST é uma aplicação de análise filogenômica baseada em inferência Bayesiana. A biblioteca BEAGLE 3 torna mais eficiente a paralelização em escala fina de cálculos métodos de Markov chain Monte Carlo (MCMC) realizados pelo BEAST.

BEAST 1.8, BEAGLE 3 e as respectivas bibliotecas foram instaladas no ambiente do SDumont. Foram levantados cenários, para as diferentes chamadas CPU, GPU e multi-CPU/GPU acoplando a devida parametrização do BEAST e BEAGLE e levando em consideração as características e natureza dos dados. Os dados de entrada foram arquivos de sequencias biológicas do vírus da Dengue no formato XML.

RESULTADOS

Dados moleculares do vírus da Dengue em formato XML foram extraídos do diretório benchmark do BEAST 1.8 e usados nas análises. O parâmetro chainLength do BEAST 1.8 está relacionado ao cálculo das cadeias MCMC. O incremento no valor do chainLength fornece consistência as análises bayesianas, mostrando-se proporcional ao incremento no tempo computacional. As execuções foram realizadas usando CPU 24 threads, CPU 40 threads, 1 GPU, CPU 40 threads/1 GPU, CPU 40 threads/8 GPU, e 8 GPU.

O melhor desempenho foi obtido usando 8 GPU. As execuções usando um valor baixo no parâmetro chainLength = “100000” e exploratório e requer menos gasto computacional. Quando usado um valor maior de chainLength = “10000000” tende a gerar uma maior consistência nos resultados, os cálculos de probabilidade requeridos tornam-se mais exaustivos levando a um gasto computacional maior. Nos nossos resultados, o incremento do chainLength influencia no maior tempo computacional obtido, mesmo assim ambas as execuções de chainLength apresentaram um comportamento muito similar no uso dos recursos CPU e GPU.

Para os cálculos foi usado como variabilidade o parâmetro chainLength fixado em “100000” e “10000000”. Os experimentos sugerem que as características como tamanho dos dados e configuração de parâmetros no BEAST, como o chainLength, influenciam no tempo computacional. As execuções realizadas fixando o uso de recursos em 8 GPU apresentam melhor desempenho quando comparadas ao demais ambientes multi-core e 1 GPU.

PUBLICAÇÕES EM EVENTO CIENTÍFICOS

Trabalhos completos publicados em anais de congressos

Ocaña, K.A.C.S.; Coelho, M. ; Freire, G. ; Osthoff, C. High-Performance Computing of BEAST/BEAGLE in Bayesian Phylogenetics using SDumont Hybrid Resources. In: Brazilian e-Science Workshop (BreSci 2020), 2020, Cuiabá, Mato Grosso. Anais do XIV Brazilian e-Science Workshop, 2020.

Resumos expandidos publicados em anais de congressos

Freire, G. ; Coelho, M. ; Ocaña, K.A.C.S. ; Osthoff, C. Exploração de Módulos Paralelo Híbrido de Bioinformática para Ambientes GPU de Supercomputação. In: VI Escola Regional de Alto Desempenho do Rio de Janeiro, 2020. VI Escola Regional de Alto Desempenho do Rio de Janeiro, 2020.

Resumos e pôster publicados em anais de congressos

The Event for High Performance Computing, Machine Learning and Data analysis (ISC, <https://www.isc-hpc.com/>). Developing Efficient Scientific Gateways for Bioinformatics in Supercomputer Environments supported by Artificial Intelligence. 2021. (Congresso). Kary Ocaña, Micaella Coelho, Rafael Terra, Guilherme Freire, Mayconn Santos, Lucas Cruz, Marcelo Galheigo, André Carneiro, Bruno Fagundes, Diego Carvalho, Douglas Cardoso, Esteban Meneses, Luiz Gadelha, Carla Osthoff. 2021.

CONCLUSÕES

O presente estudo viabiliza a exploração e análise de desempenho do BEAST/BEAGLE em ambientes de CAD com a especificação do ambiente computacional que leve a um desempenho mais eficiente. Dessa maneira, permite que usuários possam usufruir dessas informações e realizar execuções garantindo um uso racional do ambiente do SDumont.

Análises de desempenho do BEAST/BEAGLE em múltiplas configurações de CPU/GPU no SDumont sugerem o uso da configuração híbrida CPU 40 threads e 8 GPU como a mais eficiente. Sobre a variabilidade no número de chainLength fixados em “100000” e “10000000” como esperado o tempo computacional gasto na execução é incremental, mas o comportamento é muito similar.

Como trabalhos futuros visamos uma análise de desempenho em vários nós em paralelo, visando uma melhora na execução GPU, explorando ambientes multi-GPU. No entanto uma nova versão do BEAST tenha que ser explorada, especificamente aquela que apresente consistência com a chamada em GPU e multi-nós.

BEAST/BEAGLE está integrado ao Portal-Bioinfo (<https://bioinfo.lncc.br/>) como uma aplicação de bioinformática. Os resultados apresentados apoiam o uso de portais científicos verdes e mais eficientes. Alguns artigos relacionados a esse projeto foram apresentados nos eventos BreSci2020, Erad2020 e Isc2021.

REFERÊNCIAS BIBLIOGRÁFICAS

- Ocana, K., Coelho, M., Freire, G., and Osthoff, C. (2020). High-performance computing of beast/beagle in bayesian phylogenetics using sdumont hybrid resources. In 14º BreSci – Brazilian e-Science Workshop.
- C.-L. Hung, Y.-S. Lin, C.-Y. Lin, Y.-C. Chung, e Y.-F. Chung, “CUDA ClustalW: An efficient parallel algorithm for progressive multiple sequence alignment on Multi-GPUs,” *Comput Biol Chem*, vol. 58, pp. 62– 68, May 2015.
- Yin, Z., Lan, H., Tan, G., Lu, M., Vasilakos, A. V., and Liu, W. (2017). Computing platforms for big biological data analytics: Perspectives and challenges. *Computational and Structural Biotechnology Journal*, 15:403–411.
- Jin, Z. and Bakos, J. D. (2013). Extending the beagle library to a multi fpga platform. *BMC Bioinformatics*, 14(1):25.
- J. P. Huelsenbeck e F. Ronquist, “MRBAYES: Bayesian inference of phylogenetic trees,” *Bioinformatics*, vol. 17, no. 8, pp. 754–755, Aug. 2001.

Relatório de Atividades

1 Dados Gerais

Título do projeto: Auto localização robótica e mapeamento com aprendizado de máquina

Nome do Bolsista: Gustavo Alves Casqueiro

Nome do orientador: Jauvane Cavalcante de Oliveira

Tipo de bolsa e período do relatório: PIBITI–CNPq/LNCC, Edital 2020/2021, 10/2020 - 08/2021

2 Objetivos

Este projeto teve por objetivo o estudo do funcionamento do sistema de SLAM e VSLAM integrados à plataformas robótica móveis, e o estudo da implementação de um método barato e eficiente capaz de solucionar o problema da detecção de fechamento de loop, para que se corrija a deriva nos cálculos estimados da Odometria Visual deste sistema, a partir de uma Rede Neural Convolutiva e Recorrente.

Além disso, também foi estudada e implementada a técnica chamada Recuperação de Imagens com Base em Conteúdo (em inglês, *Content-Based Image Retrieval* — CBIR) baseado em CNN — neste caso, um Autoencoder eliminador de ruídos. O objetivo consiste em realizar uma avaliação comparativa abrangente entre sistemas CBIR baseados em Hand-Crafted Features — Bag of Visual Features (BoVF) utilizados com classificadores MLP e SVM e sistemas CBIR baseados em CNN. Este estudo ajudará no progresso de pesquisas futuras em recuperação de imagens usando aprendizado profundo e em outras abordagens, como um SLAM baseado em semântica, onde o CBIR se demonstra bastante útil no domínio de Localização e Mapeamento Simultâneo (SLAM), pois pode manipular uma grande quantidade de dados em tempo real. Além disso, ele também poderá auxiliar como referencia na implementação da Rede Neural Híbrida que está sendo construída em parceria com uma Dissertação de Mestrado em andamento, desenvolvido no Laboratório de Robótica e Inteligência Computacional (RoboLAB) do Instituto Militar de Engenharia (IME).

3 Introdução

Um problema fundamental abordado pela Robótica Móvel em tarefas de navegação autônoma de plataformas robóticas é o problema de Localização e Mapeamento Simultâneos (em inglês, *Simultaneous Localization and Mapping* — SLAM) que constitui-se na resolução de tarefas complexas, tais quais: mapeamento e localização. Estas tarefas não podem ser exercidas de forma independente. [1]

O problema de SLAM, tanto no âmbito civil e militar, possibilitou o surgimento de uma gama de aplicações para Veículos Terrestres não Tripulados (em inglês, *Unmanned Ground Vehicles* — UGV), como por exemplo carros autônomos. [1]

Uma das principais tarefas exercidas pelos UGV é a sua localização. Com ela, a plataforma robótica móvel consegue navegar em um ambiente de interação, rastreando seus passos, detectando obstáculos e evitando-os de forma apropriada para que a execução de uma determinada tarefa seja finalizada com sucesso, isto é, através da coleta de dados e informações neste ambiente. Para tal, utiliza como método a Odometria fundamentada na visão — a Odometria Visual (em inglês, *Visual Odometry* — VO). [1]

Um problema com este método é a grande quantidade de soluções que utilizam recursos caros e complexos, como sensores e lasers. Portanto, existe uma demanda por soluções que utilizam recursos alternativos, que sejam mais baratos e menos complexos. A Visão Computacional (em inglês, *Computer Vision* — CV) pode ser utilizada com o SLAM, formando o VSLAM (em inglês, *Visual SLAM*), o qual é uma possível solução. Com o VSLAM, surge a necessidade de técnicas como Reconhecimento de Padrões (em inglês, *Pattern Recognition* — PR) e Aprendizado de Máquina (em inglês, *Machine Learning* — ML) baseado em Aprendizado Profundo (em inglês, *Deep Learning* — DL) para solucionar este problema. [1]

Em vista disto, este projeto de Iniciação Científica, em parceria com uma Dissertação de Mestrado em andamento, desenvolvido no Laboratório de Robótica e Inteligência Computacional (RoboLAB) do Instituto Militar de Engenharia (IME), tem por objetivo final a implementação de um método barato e eficiente capaz de solucionar o problema de detecção de fechamento de loop, para que se corrija a deriva nos cálculos estimados da Odometria Visual deste sistema a partir de uma Rede Neural Convolutiva e Recorrente.

4 Metodologia

A metodologia foi dividida entre Referências Bibliográficas, que contém as explicações das tecnologias estudadas e da técnica prática aplicada, e os detalhes da implementação em si.

4.1 Referência Bibliográfica

Os assuntos abordados são: Rede Neural Híbrida, Autoencoder e Recuperação de Imagens com Base em Conteúdo.

4.1.1 Rede Neural Híbrida

A solução proposta para a implementação do sistema que detecta o fechamento de looping no SLAM é uma rede neural profunda híbrida, chamada Long-Term Recurrent Convolutional Network (LRNC), também conhecida na literatura como Convolutional Neural Network Long Short-Term Memory (CNN LSTM). Esta rede neural ainda está na fase de planejamento e desenvolvimento, e futuramente será integrada com o sistema de VSLAM, as bibliotecas do ROS e a Nvidia Jetson Nano que será acoplada ao robô. [2]

Segue abaixo uma ilustração que explica o funcionamento da rede neural utilizada:

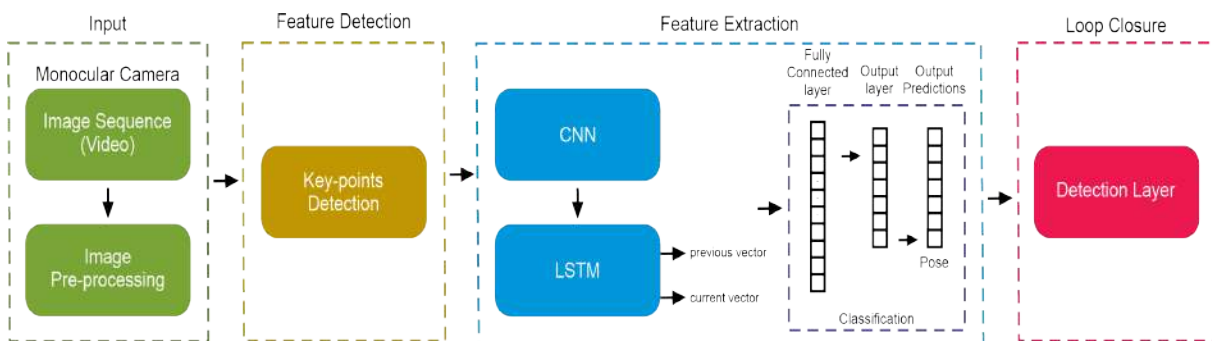


Figura 1: Esquema lógico da LRNC proposta para o problema. Fonte: [2]

A primeira etapa, Input, consiste na obtenção da sequência de imagens por via da câmera acoplada no robô, e na realização do pré-processando dessas imagens para que sejam melhor trabalhadas pelas etapas seguintes. Em Feature Detection, será feita a detecção e descrição de características das imagens pré-processadas através de um Descritor Binário Local. Em Feature Extraction, que é onde estão de fato as Redes Neurais Convolucionais e Recorrentes, será feita a alimentação das características obtidas na etapa anterior no formato de filtros convolucionais, e onde ocorrerá a tarefa de classificação. Finalmente, na última etapa, com o auxílio do sistema de VSLAM, será feito a detecção do fechamento de looping. [2]

Para que a Rede Neural desenvolvida seja viável de ser treinada e utilizada, foram utilizadas algumas técnicas para melhorar a sua performance, como camadas esparsas na CNN e pesos binários nas arestas. Baoyuan Liu et al. mostrou em seu trabalho a grande redução de complexidade computacional que se ganha ao utilizar a CNN esparsa, com a eliminação de parâmetros redundantes que não são significantes no resultado final, com uma perda de apenas 1% de precisão [3]. Também visando uma melhoria na etapa do treinamento e execução da rede neural, Courbariaux et al. nos mostra que uma alternativa para contornar um dos maiores problemas da Deep Learning, que é o da grande capacidade computacional geralmente exigida para seus algoritmos funcionarem, consiste na utilização de pesos e ativações binários. Com isso, muitas operações que antes eram feitas aritmeticamente, podem agora ser solucionadas com operações binárias, que são significativamente mais rápidas, e portanto aceleram e tornam mais viável o uso das Deep Neural Networks (DNN) para diversos problemas, como a rede neural híbrida criada. [4]

A implementação da adaptação de modelo híbrido de arquitetura de Rede Neural Artificial — a CNN LSTM para a solução do problema de detecção de fechamento de loop, está sendo desenvolvida em parceria com uma Dissertação de Mestrado em andamento, desenvolvido no Laboratório de Robótica e Inteligência Computacional (RoboLAB) do Instituto Militar de Engenharia (IME). Uma etapa fundamental desta adaptação que consiste na reformulação de filtros convolucionais de uma arquitetura de Rede Neural Convolucional tradicional através de um Descritor Binário Local poderá ser consultado posteriormente, em artigo submetido no periódico científico internacional *Journal of Intelligent & Robotics Systems* (JINT) com classificação Qualis B1 e *Impact factor* 2.259.

4.1.2 Autoencoder

Os autoencoders são Redes Neurais de aprendizado não-supervisionado, cujo objetivo é copiar o input para o output. A partir do que recebem como entrada, eles a codificam na chamada representação espaço-latente (a parte da codificação), e a partir desta representação eles conseguem reconstruir a imagem original (decodificação). Essa codificação na verdade é uma compressão da imagem, então imagens similares serão comprimidas de forma igual, e o resultado final da saída será igual também.

Esta estrutura é útil para uma série de tarefas. Um exemplo é o de remoção de ruído nas imagens: ao receber uma imagem com ruído, o chamado denoisy autoencoder (que foi treinado com imagens similares sem imperfeições) extrai as características mais relevantes da imagem na codificação, deixando de lado características irrelevantes, e a partir de seu código a imagem original sem perturbações é reconstruída. Segue abaixo um esquema representando o funcionamento deste algoritmo:

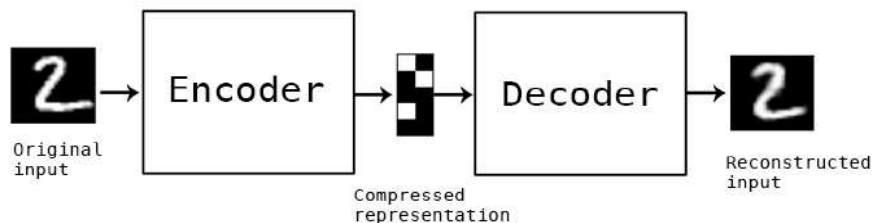


Figura 2: Esquema representativo de um autoencoder. Fonte: [5]

4.1.3 Content-Based Image Retrieval

O Content-Based Image Retrieval é uma técnica que se utiliza da primeira parte de um denoisy auto-encoder, a codificação. Um CBIR faz a recuperação de imagens baseado apenas em seu conteúdo e suas características, como cores, formatos, textura entre outros, sem necessitar de rótulos predeterminados para a imagem. A partir de um banco de dados de imagens disponíveis para serem recuperadas e de uma imagem de input, o CBIR retorna as imagens similares disponíveis no banco. [6]

O seu funcionamento se dá, inicialmente, com a extração e armazenamento das características das imagens do dataset que conterá as imagens disponíveis para serem buscadas. Após esta etapa, para se fazer a query de uma imagem específica, se extrai as suas características de forma similar que no treinamento do CBIR, e então recupera-se as imagens com as características mais similares, empregando uma função de distância para determinar o quão parecidas elas são. [6] Segue abaixo um fluxograma que explica a sequência de eventos envolvidos no treinamento e na recuperação de imagens em um CBIR:

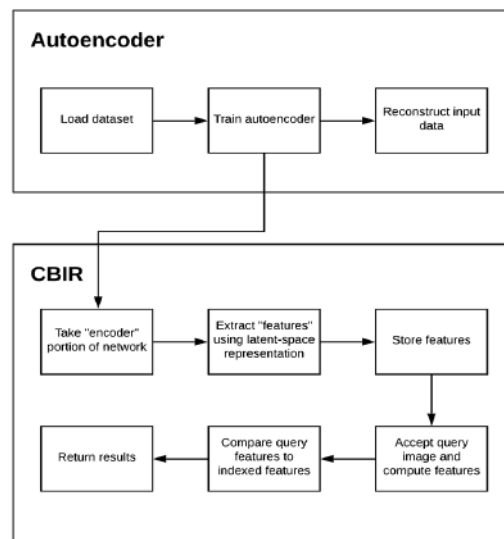


Figura 3: Fluxograma do funcionamento do CBIR com denoisy autoencoder. Fonte: [7]

O CBIR é uma ferramenta muito útil para o uso em sistemas baseados em SLAM, devido à grande capacidade de manipulação de imagens em tempo real. A construção do CBIR poderá ajudar como uma referência para avaliar a implementação da Rede Neural Híbrida proposta, bem como também ajudará no progresso de pesquisas futuras em recuperação de imagens usando aprendizado profundo e em outras abordagens, como um SLAM baseado em semântica.

4.2 Implementação do CBIR

Para a construção do CBIR foi utilizada a plataforma Google Colab, com a linguagem Python. As principais bibliotecas utilizadas foram a Numpy (para manipulação do formato dos dados), Opencv (para a manipulação de imagens) e Keras (para a construção dos algoritmos de machine learning e deep learning).

O dataset utilizado foi o MNIST: Modified National Institute of Standards and Technology database (MNIST), que contém 70000 imagens de algarismos escritos a mão, de 28x28 pixels.

Inicialmente, foi construído o autoencoder eliminador de ruídos. O treinamos com o dataset MNIST, importado diretamente do keras.datasets, que contém 60000 imagens (as quais serão disponíveis para serem recuperadas através das queries no CBIR). Depois, utilizando a primeira metade do autoencoder, a parte da

codificação, construímos o sistema do CBIR, que recupera e mostra em tela as imagens recuperadas a partir de uma entrada, e mostra os scores computados relacionados a essas imagens.

5 Resultados e Discussão

Para demonstração do resultado do experimento, foi utilizada uma imagem arbitrária para fazer a query, e ver os resultados buscados. A imagem buscada foi a seguinte:



Figura 4: Imagem utilizada para fazer a query

Os resultados retornados estão demonstrados em figura abaixo:



Figura 5: Imagens do dataset recuperadas

Para estes resultados, foi utilizado o dataset MNIST. Posteriormente, outros datasets mais complexos podem ser utilizados para os mesmos experimentos e análises, contribuindo de maneira mais significativa para os objetivos propostos. Algumas sugestões são:

JAFFE: Japanese Female Facial Expression [8], Extended CK+: Extended Cohn-Kanade [9], FEI: FEI Face Database [10], CIFAR-10: Canadian Institute For Advanced Research (CIFAR-10) [11], FER-2013: Facial Expression Recognition 2013 [12].

Estes datasets anteriores seriam utilizados com o intuito de avaliar melhor o sistema de CBIR construído. Após isso, experimentos futuros vão focar mais em trabalhar com datasets relacionados com a área do SLAM e do Loop Closure Detection, como:

New College and the City Centre datasets originally used by FAB-MA [13], LipIndoor and LipOutdoor datasets [14], Bovisa dataset [15], KITTI dataset [16]

6 Conclusões

Considerando o que foi proposto neste projeto de Iniciação Científica, foi apresentado o estudo sobre o SLAM e o VSLAM, e foi detalhada a solução que está sendo proposta da Rede Neural Híbrida, que detecta o fechamento de looping. Esta rede neural ainda está em desenvolvimento, e há um considerável trabalho para a descoberta não apenas dos melhores parâmetros e configurações para melhorar a sua performance, mas também para torná-la viável econômica e computacionalmente, utilizando a menor quantidade de recursos possíveis de forma que se mantenha acima de um nível desejado de precisão.

Este tipo de análise demanda muitos esforços, portanto ferramentas e sistemas que permitem fazer comparações abrangentes são essenciais para que o projeto tenha um bom andamento, como é o caso do sistema de CBIR construído. Trabalhos futuros poderão explorar formas de utilizar este sistema para encontrar as melhores configurações possíveis para a Rede Neural Híbrida.

Gostaria de agradecer inicialmente ao CNPq e ao LNCC, por terem me proporcionado a bolsa que tornou possível a realização deste trabalho. Também agradeço aos meus orientadores Jauvane Cavalcante

de Oliveira, Paulo Fernando Ferreira Rosa e Alexandra Miguel Raibolt da Silva, por me acompanharem e me guiarem tão bem na pesquisa.

Referências

- [1] A. Raibolt, A. Angonese, P. Rosa, and G. Casqueiro, “Towards loop closure detection for slam applications using bag of visual features: Experiments and simulation,” [submitted].
- [2] A. Raibolt, A. Angonese, and P. Rosa, “Comparative evaluation of feature descriptors through bag of visual features with multilayer perceptron on embedded gpu system,” in *2020 Latin American Robotics Symposium (LARS), 2020 Brazilian Symposium on Robotics (SBR) and 2020 Workshop on Robotics in Education (WRE)*, 2020, pp. 1–6.
- [3] B. Liu, M. Wang, H. Foroosh, M. Tappen, and M. Pensky, “Sparse convolutional neural networks,” 06 2015, pp. 806–814.
- [4] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, “Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or -1,” 02 2016.
- [5] “Building autoencoders in keras,” <https://blog.keras.io/building-autoencoders-in-keras.html>, accessed: 2021-08-05.
- [6] Y. Liu, D. Zhang, G. Lu, and W.-Y. Ma, “A survey of content-based image retrieval with high-level semantics,” *Pattern Recognition*, vol. 40, pp. 262–282, 01 2007.
- [7] “Autoencoders for content-based image retrieval with keras and tensorflow,” <https://www.pyimagesearch.com/2020/03/30/autoencoders-for-content-based-image-retrieval-with-keras-and-tensorflow/>, accessed: 2021-08-05.
- [8] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, “Coding facial expressions with gabor wavelets,” in *Proceedings of the 3rd. International Conference on Face and Gesture Recognition*, 1998, pp. 200–205.
- [9] T. Kanade, J. Cohn, and Y. Tian, “Comprehensive database for facial expression analysis,” *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 46–53, 2000.
- [10] Z. Zheng, H. Zheng, Z. Yu, Z. Gu, and B. Zheng, “Photo-to-caricature translation on faces in the wild,” 11 2017.
- [11] A. Krizhevsky, “Learning multiple layers of features from tiny images,” pp. 32–33, 2009. [Online]. Available: <https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf>
- [12] L. Wang, S. Xu, X. Wang, and Q. Zhu, “Eavesdrop the composition proportion of training labels in federated learning,” 10 2019.
- [13] M. Ramezani, Y. Wang, M. Camurri, D. Wisth, M. Mattamala, and M. Fallon, “The newer college dataset: Handheld lidar, inertial and vision with ground truth,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.
- [14] D. Mascharka and E. Manley, “Lips: Learning based indoor positioning system using mobile phone-based sensors,” in *2016 13th IEEE Annual Consumer Communications Networking Conference (CCNC)*, 2016, pp. 968–971.
- [15] D. Valiente, A. Gil, L. Payá, J. Sebastián, and Ó. Reinoso, “Robust visual localization with dynamic uncertainty management in omnidirectional slam,” *Applied Sciences*, vol. 7, p. 1294, 12 2017.
- [16] J. Xie, M. Kiefel, M.-T. Sun, and A. Geiger, “Semantic instance annotation of street scenes by 3d to 2d label transfer,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Relatório de Atividades

Título: Gerenciamento de grandes volumes de dados nos sistemas Web do SINAPAD

Bolsista: Lucas Fernandes Calheiros

Orientador: Antônio Tadeu Azevedo Gomes

Tipo da bolsa: PIBIT/LNCC

Período: Out/2020 a Ago/2021

Áreas prioritárias (Portaria MCTI 1.122/2020):

Tecnologias Habilitadoras (Inteligência Artificial)

-

Sub-área do conhecimento (Tabela CNPq):

1.03.03.00-6 – Metodologia e Técnicas da Computação (Engenharia de Software)

1 Objetivos

O objetivo principal deste projeto é: **investigar alternativas arquiteturais para os sistemas Web do SINAPAD, considerando abordagens modernas de gerenciamento de grandes volumes de dados, e prototipar soluções baseadas na(s) alternativa(s) mais promissora(s).**

2 Introdução

Como parte de seus objetivos, o sistema SINAPAD oferece 3 sistemas Web:

- O sistema de gerenciamento de projetos, de utilização de recursos e de produções científicas associados ao supercomputador Santos Dumont (<http://sdumont.lncc.br> e <https://intranet.sdumont.lncc.br>);
- O portal de modelagem molecular Dockthor (<http://dockthor.lncc.br>); e
- O portal de bioinformática BioInfo (<http://bioinfo.lncc.br>).

Os 3 sistemas empregam basicamente as mesmas tecnologias de desenvolvimento Web (Bootstrap, AngularJS, JQuery) e de bancos de dados (MySQL). Mais importante, esses sistemas compartilham uma estrutura arquitetural semelhante, com aplicação de *design patterns* consagrados na literatura de desenvolvimento Web, como *Model-View-Controller* [1].

Todos os sistemas Web do SINAPAD podem ser hoje considerados *sistemas intensivos de dados*, isto é, sistemas que manipulam um grande volume de dados e que apresentam atributos de qualidade de desempenho como escalabilidade e disponibilidade. Esses sistemas também podem ser considerados como *reativos*, no sentido de demandarem atributos de qualidade como usabilidade e tempo de resposta. Para lidar com essa combinação “volume de dados X reatividade”, a equipe SINAPAD desenvolveu adaptações específicas às tecnologias listadas acima, adaptações estas que foram evoluindo em consonância com a evolução dessas tecnologias. Um efeito negativo dessa evolução acoplada foi a degradação arquitetural dos sistemas Web do SINAPAD, que se tornaram demasiado complexos de se manter.

Com o advento de sistemas específicos de gerenciamento de grandes volumes de dados (como Spark e Cassandra) e de processamento de eventos (como Kafka), a equipe SINAPAD tem a oportunidade de rever a organização arquitetural dos sistemas Web do SINAPAD, de modo a melhor atender os atributos de qualidade supracitados. Além disso, o suporte dessas tecnologias abre a possibilidade de uso de soluções de Inteligência Artificial que apoiem as tomadas de decisão dos usuários desses sistemas no nível de gestão, como por exemplo na atualização das políticas de filas do supercomputador Santos Dumont.

Para alcançar o objetivo do projeto, propomos o seguinte conjunto de atividades e vislumbramos os resultados esperados correspondentes:

1. Estudar arquiteturas baseadas em microsserviços / Conhecimento adquirido;
2. Estudar arquiteturas dirigidas por eventos / Conhecimento adquirido;
3. Analisar tecnologias existentes de gerenciamento de grandes volumes de dados / Conhecimento adquirido;
4. Analisar tecnologias existentes de processamento de eventos / Conhecimento adquirido;
5. Propor alternativas arquiteturais para os sistemas Web do SINAPAD / Relatório técnico;
6. Prototipar continuamente os sistemas Web do SINAPAD com base nas tecnologias estudadas e alternativas arquiteturais propostas / Pacote de software.

3 Metodologia

Como parte das atividades 2 e 4, começamos com o estudo do Kafka,¹ uma plataforma de processamento de eventos altamente escalável e com suporte à criptografia de mensagens e fácil integração com sistemas gerenciadores de bancos de dados como PostgreSQL e MySQL.

¹<https://kafka.apache.org/>

Posteriormente, dentro do contexto da atividade 5, remodelamos a arquitetura do sistema de gerenciamento do Santos Dumont (ilustrada na Figura 1) para uma nova arquitetura com uma estrutura mais distribuída e também mais eficiente (ilustrada na Figura 2), adotando o Kafka como componente central da arquitetura. Como passo seguinte, pela atividade 1, iniciamos o estudo de arquiteturas baseadas em microsserviços, empregando a tecnologia de containers Docker² como estudo de caso e mapeando assim os componentes da nova arquitetura em diferentes containers (como ilustrado na Figura 3).

Após a implementação dos containers, iniciamos a atividade 6, prototipando a parte do sistema de gerenciamento do SDumont responsável pela apresentação de gráficos de monitoramento e de estatísticas das filas e de projetos do SDumont. Para tanto, adotamos o Grafana,³ uma ferramenta para consultar e visualizar métricas graficamente. Empregamos uma abordagem de separação da base de dados original do sistema em bases específicas por funcionalidade, de modo que tabelas continuamente alimentadas pelo Kafka foram criadas exclusivamente para propiciar a geração de gráficos no Grafana.

Ainda como parte do plano de trabalho original, estava prevista a análise de tecnologias existentes de gerenciamento de grandes volumes de dados (atividade 3), mas infelizmente essa atividade não pôde ser conduzida por conta das outras atividades terem tomado um grande tempo de estudo, principalmente pelo fato de termos inicialmente utilizado Windows em vez de Linux (ambiente onde está implantado o sistema atual de gerenciamento do SDumont) como plataforma de desenvolvimento. Contudo, esta atividade continua prevista numa eventual renovação da bolsa.

²<https://www.docker.com/>

³<http://grafana.com>

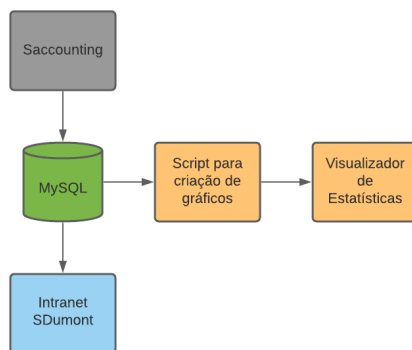


Figura 1: Arquitetura existente do sistema de gerenciamento do SDumont. O componente “Saccounting” é uma aplicação Python que coleta periodicamente do escalonador de tarefas do SDumont informações sobre as aplicações científicas executadas pelos usuários do SDumont.

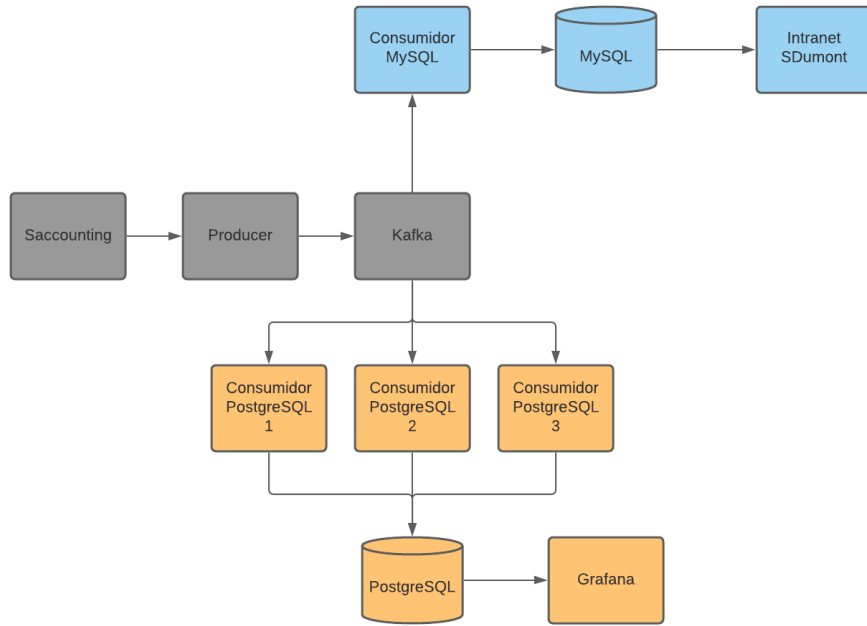


Figura 2: Arquitetura proposta.

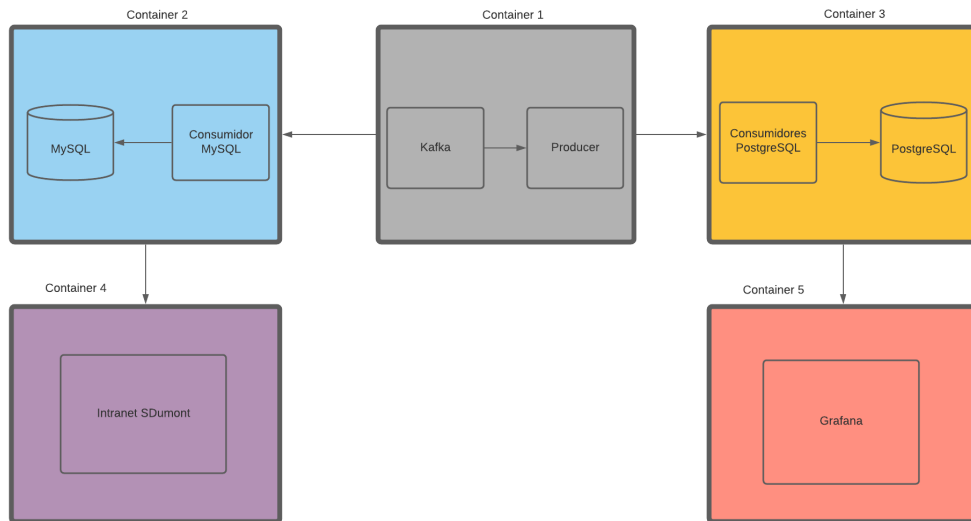


Figura 3: Partição dos Containers Docker.

4 Resultados

Após os estudos e aplicação das novas tecnologias, obtivemos uma arquitetura melhor compartimentada, e também mais eficiente em comparação à arquitetura antiga. Mais à frente pretende-se refinar a implementação dos gráficos de monitoramento e de apresentação de estatísticas das filas e de projetos do SDumont no sistema de produção, utilizando a arquitetura proposta.

5 Conclusões

No nosso entendimento, o projeto evoluiu bastante, já que nos foi possível compreender porque o sistema atual é lento e inflexível na geração dos gráficos de monitoramento e de estatísticas, uma vez que se vale de consultas SQL desnecessariamente complexas. Com a alteração proposta na arquitetura, identificamos ser possível obter e processar os dados de utilização do SDumont de maneira muito mais flexível e eficiente, fragmentando a base de dados e usando componentes especializados, interligados via um barramento de mensagens, para cada uma das tarefas presentes no sistema de gerenciamento do SDumont. Sentimos também termos evoluído bastante no entendimento de conceitos e na manipulação de tecnologias que não são usualmente vistos na graduação.

Referências

- [1] Trygve Reenskaug. Models-views-controllers. Technical report, Xerox PARC, 1979.

Relatório de Atividades

1. Dados gerais

Título do projeto: Projeto e Implementação de *Workflows* Científicos Reprodutíveis de Alto Desempenho

Bolsista: Lucas da Cruz Silva

Orientador: Luiz Manoel Rocha Gadelha Júnior

Coorientador: Carla Osthoff e Kary Ocaña

Tipo de bolsa: Iniciação Científica

Período do relatório: 01/09/2020 - 31/08/2021

2. Objetivos

O presente trabalho se desenvolve a partir de um projeto multidisciplinar que envolve o Laboratório de Bioinformática (LABINFO) e o Centro Nacional de Processamento de Alto Desempenho (CENAPAD) do Laboratório Nacional de Computação Científica (LNCC), com o objetivo de acoplar ao Bioinfo-Portal, um portal de Bioinformática hospedado no LNCC, um *workflow* científico de sequenciamento RNA (RNA-Seq). Dessa forma, através da motivação de facilitar o gerenciamento (definição, execução e monitoração) de atividades extensivas, complexas e computacionalmente custosas, este trabalho tem por objetivo, modelar, executar e analisar *workflows* científicos com tecnologias e ambientes de Processamento de Alto Desempenho (PAD), tomando como caso de estudo o experimento de bioinformática de sequenciamento RNA.

3. Introdução

Em diversas áreas da ciência existem experimentos que podem ser computacionalmente intensivos, como é o caso dos experimentos de bioinformática de sequenciamento RNA. A técnica de RNA-Seq trata da análise de Expressão Diferencial de Genes (EDG) e sequenciamento genômico, ou seja, que permite estudar o comportamento de um conjunto de transcritos de uma célula em uma dada condição fisiológica ou de desenvolvimento, tal como o câncer. A tecnologia de RNA-Seq demonstra um grande avanço nos estudos da transcritômica, no entanto, existem ainda muitos desafios relacionados à complexidade, produção e consumo de grande volume de dados e uso de programas e *software* de alto custos computacionais. Tipicamente, podem levar muitas horas ou mesmo dias para que um estudo de caso de RNA-Seq seja processado. Sob essa condição, diversas áreas da ciência têm se apoiado a procedimentos e uso de infraestruturas computacionais no decorrer dos anos [Cordeiro et al. 2013].

Computação de Alto Desempenho (CAD), ciência de dados e aprendizado de máquinas são exemplos de áreas da computação que se aliam a bioinformática visando desenvolver estratégias para prover um baixo tempo e custo computacional. Outra tecnologia que cabe citar são os *workflows* científicos, que desenvolvem uma forma abstrata de representar esses experimentos, modelando, computacionalmente, e encadeando as etapas experimentais, isso permite que a gerência de execução e análise sejam mais bem estabelecidas. E isso leva a reforçar a reprodutibilidade, confiabilidade e escalabilidade do experimento. Para atingir esses três requisitos de experimentos científicos, é necessário utilizar ferramentas que forneçam a infraestrutura para definir, executar e monitorar o *workflow*, permitindo que toda a modelagem abstrata desenvolvida pelo cientista seja traduzida a nível de máquina.

Sistema de Gerência de *Workflows* Científicos (SGWfC) ou mesmo determinadas linguagens de programação são capazes de orquestrar a execução de *workflows* e até acoplar *workflows* computacionalmente intensivos em ambientes de CAD. Nesse âmbito existem SGWfC baseados em sistema *web*, como o Galaxy e pacotes estatísticos do R e Bioconductor, como DESeq2 e EdgeR, que são usados nos estudos de EDG e para automação das tarefas os SGWfC distribuídos e paralelos Kepler, Pegasus e as linguagens Swift e Python/Parsl.

Este trabalho, até a presente data, apresenta uma análise comparativa da modelagem e desempenho computacional da nova versão do *workflow* científico ParsIRNA-Seq, com seis atividades

[Cruz et al. 2021], comparada a versão anterior, de três atividades [Cruz et al. 2020]. Essas execuções dessas versões foram acopladas a ambientes de supercomputação. A versão atual, que é apresentada como “versão otimizada”, é composta por seis atividades principais, onde as modificações foram realizadas para utilização da estratégia de divisão e conquista para melhorar o desempenho na execução da atividade HTSeq, pois com sua atualização, o programa HTSeq passou a permitir execuções paralelas em múltiplos *cores*. Embora tenha sido necessário incluir mais três novas atividades para a paralelização do HTSeq, a nova versão “otimizada” do ParsIRNA-Seq pode levar as execuções a alcançarem um ganho em tempo computacional maior do que 65% em relação a versão anterior. A grande vantagem da versão atual se dá pela execução paralela e distribuída em múltiplos nós, pois na versão anterior não haviam ganhos de desempenho significativos nas execuções multinós. O ambiente computacional utilizado para as análises de desempenho do ParsIRNA-Seq é o supercomputador Santos Dumont (SDumont).

Este relatório está organizado da seguinte forma: a Seção 1, apresenta informações gerais sobre o relatório; a Seção 2 traz o resumo; na Seção 3 a presente introdução; a Seção 4 traz a metodologia adotada na modelagem do *workflow* de RNA-Seq desenvolvido e sua descrição; na Seção 5 são apresentados os resultados e análises experimentais; a Seção 7 todas as publicações, apresentações e prêmios em eventos científicos, frutos desta pesquisa; e, por fim, a Seção 6 traz as considerações finais e os trabalhos futuros.

4. Metodologia

4.1. Modelagem do *workflow* de RNA-Seq

Para modelar computacionalmente um *workflow* científico, inicialmente, é requerido que as atividades ou etapas do experimento sejam bem mapeadas, a fim de fazer uma abstração de todos os procedimentos que o cientista realizaria sem a utilização de um ambiente computacional. A modelagem requer o conhecimento de quantas atividades são realizadas, qual a ordem dessas atividades e as dependências entre elas. O *workflow* de RNA-Seq inicialmente desenvolvido em [Cruz et al. 2020], denominado ParsIRNA-Seq é composto por três atividades principais: Bowtie, HTSeq e DESeq (Figura 1(a)). Na atual versão do ParsIRNA-Seq, o *workflow* passa a ser composto por seis atividades (Figura 1(b)), as quais foram adicionadas Sort, Split_Picard e Merge_HTSeq [Cruz et al. 2021]. Todas incluídas em função da melhora de desempenho da atividade HTSeq.

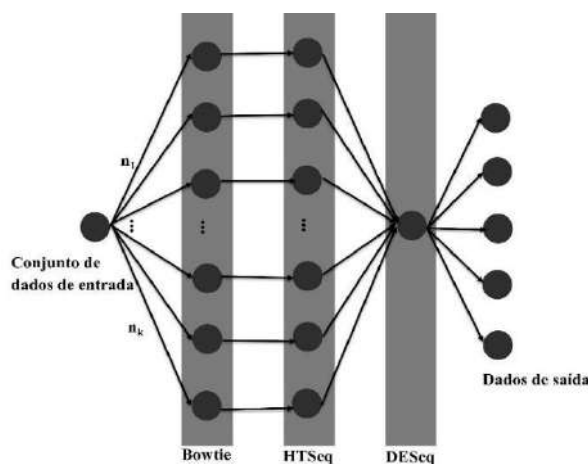
As atividades são executadas na seguinte sequência: A atividade 1, Bowtie, faz o mapeamento das leituras curtas do genoma [Langmead and Salzberg 2012]. A atividade 2, Sort, faz a ordenação das leituras. A atividade 3, Split_Picard, faz a divisão dos arquivos de leituras, dividindo cada arquivo de amostra em várias subpartes. A atividade 4, HTSeq, faz a contagem das leituras mapeadas de cada gene [Anders et al. 2014], nesse caso, as contagens serão realizadas em relação a cada subparte da amostra. A atividade 5, Merge_HTSeq, vai combinar todas as subpartes da amostra em uma única; por fim, a atividade 6, DESeq, aplica estatísticas em cima das contagens para a análise da EDG [Love et al. 2014].

Toda a modelagem computacional do ParsIRNA-Seq foi feita utilizando a biblioteca Parsl, da linguagem de programação Python, que é capaz de agir como um gerenciador de atividades dentro dos mais variados ambientes computacionais (*notebooks*, nuvem, *clusters*, supercomputadores etc.) [Babuji et al. 2019]. Além disso, é capaz de abstrair ao usuário toda a complexidade envolvida para inicialização e processamento de *workflows* dentro desses ambientes, se fazendo dessa forma uma grande ferramenta a cientistas das mais diversas áreas.

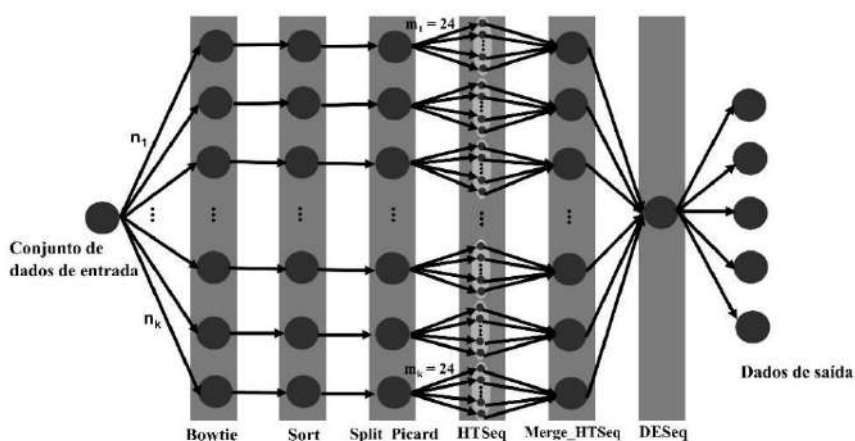
4.2. Configuração de ambiente computacional

O ambiente computacional utilizado para as execuções e as análises de desempenho é o supercomputador SDumont, que está entre as 500 máquinas mais poderosas do mundo. Ele possui uma capacidade de processamento de 5.1 Petaflop/s, com 34.688 CPU *multicores* distribuídas em

1.132 nós computacionais que são interligados por uma rede de interconexão *Infiniband* FDR/HDR. Os nós computacionais possuem duas CPUs Ivy Bridge Intel Xeon E5-2695v2 (12c @2.4GHz) e 64GB de memória RAM e uma GPU Nvidia K40. As execuções foram realizadas em nós computacional de duas CPUs Intel Xeon E5-2695v2 Ivy Bridge, 24 núcleos (12 por CPU) e 64GB de memória RAM. Todos os *softwares*, algoritmos, dependências de bioinformática (Bowtie2, Samtools, Picard, HTSeq e DESeq2) e os componentes do Parsl/Python foram alocados e instalados no ambiente do SDumont.



(a) Primeira versão do ParslRNA-Seq (três atividades).



(b) Segunda versão do ParslRNA-Seq (seis atividades).

Figura 1. Modelagem conceitual do *workflow* científico de RNA-Seq.

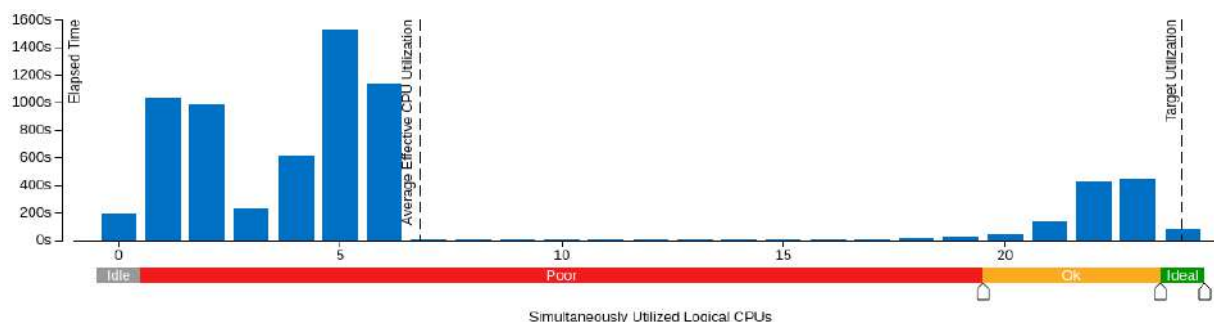
4.3. Dados de entrada

Os dados de entrada pertencem a um experimento real de RNA-Seq, foram extraídos do repositório público *Gene Expression Omnibus* (GEO), são eles: o genoma de referência; o arquivo de características genômicas no formato GTF (*Gene Transfer Format*); os arquivos de sequenciamento no formato fastq e as condições experimentais da amostra. Esses dados de RNA-Seq se referem à via metabólica de sinalização transcricional Wnt, a que tem sido reportada com um papel regulatório no coração. Um total de seis arquivos de sequenciamento, divididos em dois grupos, três de controle e três de condição Wnt sequenciadas no *Mus Musculus* (rato) foram usados como dados de entrada pelo ParslRNA-Seq. Há de se notar que o número de arquivos de sequenciamento e a divisão dos grupos podem variar conforme o experimento. Então, inicialmente, a análise de desempenho está ligada somente a esses seis arquivos, onde dois deles possuem o tamanho de 1.8GB e os outros 1.9GB, 2.0GB, 2.5GB e 3.0GB, totalizando 13GB.

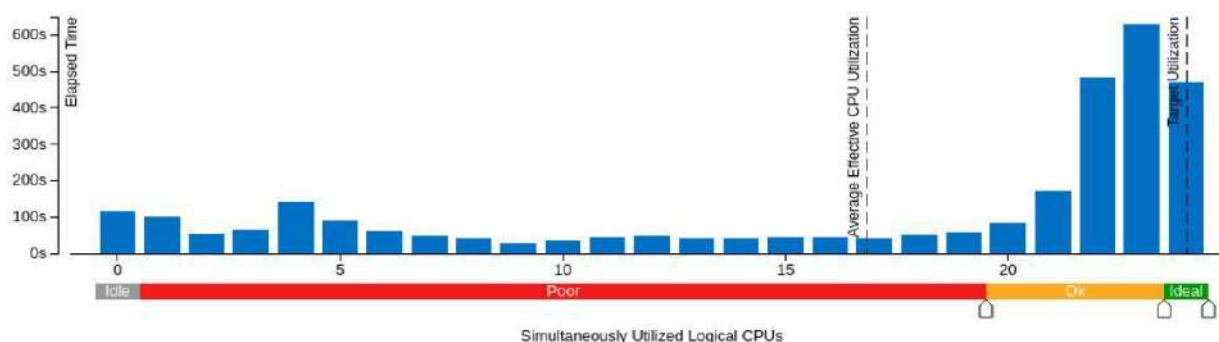
5. Resultados e discussão

A seção que segue apresenta uma análise de desempenho computacional comparando as versões do *workflow* ParsIRNA-Seq, utilizando para isso o perfilador VTune da Intel. As condições adotadas para análise se baseiam em [Cruz et al. 2020], onde são feitas análises utilizando o recurso de *multithreading* local do Parsl dentro de um nó computacional. Adicionalmente, são feitas análises para um cenário de execução em múltiplos nós, apresentando uma comparação do Tempo Total de Execução (TTE) do *workflow* entre as duas versões.

5.1 Análise de desempenho do ParsIRNA-Seq utilizando multithreading local



(a) Versão de três atividades.



(b) Versão de seis atividades.

Figura 2. Eficiência na utilização de CPUs usando recurso de *multithreading* local.

Como destacado na subseção 4.1 é imprescindível levar em consideração que as atividades da nova versão foram incluídas somente em função da melhora de desempenho do *workflow*. E isso pode ser observado pela Figura 2, onde é apresentado o histograma da eficiência na utilização das CPUs. No cenário de execução proposto foi utilizado um nó computacional e com a utilização do Parsl foi possível realizar uma paralelização de tarefas dentro do nó. A grande diferença que se dá entre os histogramas da Figura 2 é que: na Figura 2(a) há somente paralelização *multithreading* dos processos do software Bowtie e o HTSeq está utilizando somente um núcleo de CPU para executar cada amostra. Então, apesar da execução paralelizada de tarefas realizadas pelo Parsl ainda existe um número considerável de CPUs ociosas na maior parte da execução do *workflow*; agora, na Figura 2(b) é possível observar uma maior distribuição na utilização no número de núcleos de CPUs, aumentando, consideravelmente, o número de CPUs utilizadas na maior parte do tempo de execução e levando a um fator ideal de utilização de recursos computacionais. Isso se deve ao fato de ter mais uma atividade fazendo execução *multithreading*, a atividade *Sort*, e principalmente pela execução paralelizada em múltiplos *cores* do HTSeq.

5.2 Análise de desempenho do ParsIRNA-Seq para um cenário de múltiplos nós

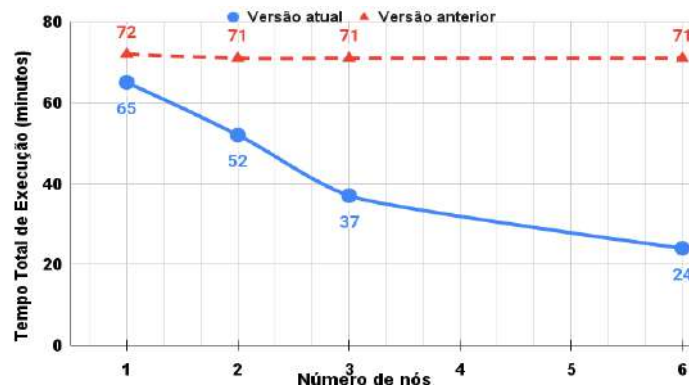


Figura 3. TTE do ParsIRNA-Seq dentro de um cenário distribuído em múltiplos nós.

Com a redução de núcleos de CPUs ociosas, por consequência, há uma redução no TTE do *workflow* de 72 para 65 minutos, quando o número de nós é igual a um, como pode ser observado na Figura 3. É imprescindível observar que a redução do TTE não ocorre de forma tão significativa pois o aumento da paralelização, aumenta também as condições de corrida para utilização dos recursos computacionais, para um cenário de um nó computacional. Por essa razão, não pode ser observado ainda nessa análise o máximo desempenho do ParsIRNA-Seq dentro de um ambiente paralelo e distribuído.

A principal razão de executar esse *workflow* dentro de um cenário paralelo e distribuído é fazer com que cada amostra seja processada dentro de um nó, dessa forma, teoricamente, seria alcançado um melhor desempenho. No entanto, ao analisar a primeira versão do ParsIRNA-Seq (três atividades) dentro de um cenário multinó, não foram observados ganhos expressivos no TTE. Como destacado na Figura 3, a linha tracejada indica a execução da primeira versão do *workflow* e o que se constata é que nem mesmo aumentando o número de nós o TTE reduz. Isso ocorre, pois, apesar de haver distribuição das amostras entre os nós, há uma execução não paralelizada no HTSeq e, além disso, há também uma barreira para sincronização de dados na altura da atividade DESeq. Esses fatores farão a amostra de maior tamanho segurar, praticamente, todo o tempo de execução do *workflow* na atividade HTSeq. Diferentemente, na segunda versão há uma distribuição de processamento entre os núcleos para o HTSeq. O que não impossibilita que a amostra de maior tamanho segure a execução do *workflow*, mas esse tempo é consideravelmente menor, como pode ser observado na linha contínua da Figura 3, que representa a execução da segunda versão do *workflow*. Portanto, o melhor TTE dentro do cenário multinó pode ser observado pela segunda versão do ParsIRNA-Seq, na Figura 3, em 6 nós, onde o TTE marca 24 minutos.

6. Publicações em eventos científicos

6.1 Trabalhos completos publicados em anais de congressos

Cruz, L., Coelho, M., Terra, R., Carvalho, D., Gadelha, L., Osthoff, C., & Ocaña, K. (2021). *Workflows Científicos de RNA-Seq em Ambientes Distribuídos de Alto Desempenho: Otimização de Desempenho e Análises de Dados de Expressão Diferencial de Genes*. In *Anais do XV Brazilian e-Science Workshop, July 18, 2021, Evento Online, Brasil*. SBC, Porto Alegre, Brasil, pages 57-64. DOI: <https://doi.org/10.5753/bresci.2021.15789>.

Terra, Rafael; Coelho, Micaella; Cruz, Lucas; Garcia-Zapata, Marco; Gadelha, Luiz; Osthoff, Carla; CARVALHO, Diego; OCAÑA, Kary. Gerência e Análises de Workflows aplicados a Redes Filogenéticas de Genomas de Dengue no Brasil. In *Anais do XV Brazilian e-Science Workshop*, 15. , 2021, Evento Online. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2021 . p. 49-56. ISSN 2763-8774. DOI: <https://doi.org/10.5753/bresci.2021.15788>.

6.2 Resumos expandidos publicados em anais de congressos

Cruz, L., Coelho, M., Gadelha, L., Ocaña, K., and Osthoff, C. (2020). Avaliação de Desempenho de um *Workflow* Científico para Experimentos de RNA-Seq no Supercomputador Santos Dumont. *In Anais Estendidos do XXI Simpósio em Sistemas Computacionais de Alto Desempenho*, outubro 21, 2020, Evento *Online*, Brasil. SBC, Porto Alegre, Brasil, *pages* 86-93. DOI: https://doi.org/10.5753/wscad_estendido.2020.14093.

Cruz, L., Coelho, M., Gadelha, L., Ocaña, K., and Osthoff, C. (2020). Desempenho de *Workflows Multithreading* no Supercomputador SDumont em Experimentos de RNA-Seq. *In Anais da VI Escola Regional de Alto Desempenho do Rio de Janeiro*, novembro 30, 2020, Evento *Online*, Brasil. SBC, Porto Alegre, Brasil, 54-56. DOI: <https://doi.org/10.5753/eradrj.2020.14520>.

6.3 Resumos e pôster publicados em anais de congressos

The Event for High Performance Computing, Machine Learning and Data analysis (ISC, <https://www.isc-hpc.com/>). *Developing Efficient Scientific Gateways for Bioinformatics in Supercomputer Environments supported by Artificial Intelligence*. 2021. (Congresso). Kary Ocaña, Micaella Coelho, Rafael Terra, Guilherme Freire, Mayconn Santos, Lucas Cruz, Marcelo Galheigo, André Carneiro, Bruno Fagundes, Diego Carvalho, Douglas Cardoso, Esteban Meneses, Luiz Gadelha, Carla Osthoff. 2021.

6.4 Prêmios

Menções Honrosa pela Segunda Colocação do *Workshop* de Iniciação Científica em Arquitetura de Computadores e Computação de Alto Desempenho (WSCAD-WIC 2020), XXI Simpósio em Sistemas Computacionais de Alto Desempenho.

7. Conclusão

Este trabalho apresentou as duas modelagens do *workflow* de RNA-Seq que foram desenvolvidas bem como uma análise comparativa sobre os desempenhos computacionais de cada uma versão do *workflow*, concluindo que a segunda versão, de seis atividades, é a que hoje mais se adequa ao cenário paralelo e distribuído do supercomputador SDumont, pois nela há uma melhor utilização dos recursos computacionais disponíveis. A nova versão apresenta uma melhora de 66,19% no tempo computacional, em relação a versão anterior para uma execução paralela e distribuída. O ParsRNA-Seq se encontra disponível no GitHub¹ para toda a comunidade científica e futuramente, será acoplado ao Bioinfo-Portal, um portal hospedado no LNCC, voltado ao fortalecimento das pesquisas envolvendo o uso da bioinformática. Serão feitos estudos sobre o desempenho dos dados de entrada e saída, visando otimizar também as operações de escrita e leitura ao longo da execução do *workflow*. Também é visado o refinamento do ParsRNA-Seq de modo a atender de uma forma melhor o público que utiliza serviço de nuvem. E, por fim, também serão realizados estudos para executar múltiplos experimentos de RNA-Seq em paralelo no SDumont.

Como apresentado, no desenvolvimento do projeto foram alcançados resultados mais do que satisfatórios o que inclui geração de códigos dos *workflows* disponíveis para a comunidade científica de bioinformática, apresentação oral nos principais fóruns em diversas comunidades de CAD, Ciência de Dados, Bioinformática e e-Ciência, publicação de trabalhos completos e resumos nos principais eventos brasileiros, publicação de pôster em evento internacional, menção honrosa em evento de CAD. Além disso, este projeto está integrado em colaboração a grupos de pesquisa internacionais (INRIA) e projetos multi-institucionais (UNB, Fiocruz, UFRJ), o que demonstra a importância do presente projeto de Iniciação Científica em termos científicos e tecnológicos.

8. Referências bibliográficas

¹ <https://github.com/lucruzz/RNA-seq>

- Anders, S., Pyl, P. T., and Huber, W. (2014). HTSeq—a Python *framework to work with high-throughput sequencing data*. *Bioinformatics*, 31(2):166–169.
- Babuji, Y., Woodard, A., Li, Z., Katz, D. S., Clifford, B., Kumar, R., Lacinski, L., Chard, R., Wozniak, J., Foster, I., Wilde, M., and Chard, K. (2019). Parsl: *Pervasive parallel programming in python*. In *28th ACM International Symposium on High-Performance Parallel and Distributed Computing (HPDC)*
- Cordeiro, D., Braghetto, K. R., Goldman, A., and Kon, F. (2013). Da ciência à e-ciência: paradigmas da descoberta do conhecimento. *Revista USP*, (97):71–81
- Cruz, L., Coelho, M., Gadelha, L., Ocaña, K., and Osthoff, C. (2020). Avaliação de desempenho de um *workflow* científico para experimentos de rna-seq no supercomputador santos dumont. In *Anais Estendidos do XXI Simpósio em Sistemas Computacionais de Alto Desempenho*, pages 86–93, Porto Alegre, RS, Brasil. SBC.
- Cruz, L., Coelho, M., Terra, R., Carvalho, D., Gadelha, L., Osthoff, C., and Ocaña, K. (2021). *Workflows* científicos de rna-seq em ambientes distribuídos de alto desempenho: Otimização de desempenho e análises de dados de expressão diferencial de genes. In *Anais do XV Brazilian e-Science Workshop*, pages 57–64, Porto Alegre, RS, Brasil. SBC.
- Langmead, B. and Salzberg, S. L. (2012). *Fast gapped-read alignment with bowtie 2*. *Nature methods*, 9(4):357.
- Love, M. I., Huber, W., and Anders, S. (2014). *Moderated estimation offold change and dispersion for rna-seq data with deseq2*. *Genome biology*, 15(12):550.

RELATÓRIO DE ATIVIDADES

Título do Projeto: Modelagem e Integração de Bancos de Dados Relacionais na arquitetura do Bioinfo-Portal

Nome do bolsista: Marco Antonio Silva Cabral

Nome do orientador: D.Sc. Kary Ann del Carmen Ocaña Gautherot (Tecnologista Pleno – LABINFO/LNCC)

Nome do Coorientador: B.Sc. Marcelo Monteiro Galheigo (Analista Sênior – SINAPAD/LNCC, Coorientador)

Tipo de bolsa: PIBIC

Período do relatório: 26/04/2021 a 30/07/2021

INTRODUÇÃO

O Projeto de Iniciação Científica (IC) se desenvolve no nível de estágio supervisionado sob coordenação da orientadora Kary Ocaña (LNCC) e com bolsa IC financiada pelo CNPq. Nesse projeto pretende-se implementar a camada baseada na modelagem de Banco de Dados (BD) no Bioinfo-Portal para o armazenamento organizado de informações de proveniência relacionadas às aplicações de bioinformática no Bioinfo-Portal, tal que apoiem a tomadas de decisão inteligentes futuras i.e., tolerância a falhas, alocação e escalonamento de processos, instanciação de máquinas, etc.

Um *gateway* científico é um sistema de arquitetura voltado para a execução automática de aplicações, ferramentas e coleções de dados científicos através de uma interface *web* comum para gerência de processos, execução de tarefas e visualização, podendo estar atrelados à ambientes de Computação de Alto Desempenho (CAD) especializados em recursos distribuídos geograficamente, um exemplo representativo sobre *gateways* se menciona em <http://www.sciencegateway.org/>. A questão importante em *gateways* científicos é orquestrar tarefas em um conjunto distribuído de recursos com tempo e custos minimizados, garantindo integridade e consistência nas informações. Para tornar um *gateway* científico "verde" e "inteligente" informações científicas e de proveniência devem ser processados dos dados de origem e processados pelos *gateways*, os que devem ser armazenados de maneira consistente para serem consultados e analisados. Banco de dados é uma área consolidada fundamentada na teoria do Modelo de Bancos de Dados e nos Sistemas de Gerência de Bancos de Dados (SGBD). BD são usados em diferentes atividades e áreas como: (i) sítios web tal e como Google, Facebook, Amazon; (ii) empresas para manter informações sobre negócios; (iii) pesquisas científicas como Astronomia (Sloan Digital Sky

Survey, Dark Energy Survey, Large Synoptic Survey Telescope), Biodiversidade (Global Biodiversity Information Facility) e Genômica: (GenBank, DNA Databank of Japan, European Molecular Biology Laboratory), entre outros.

O presente Projeto de IC visa focar nas pesquisas relacionadas a implementar funcionalidades de gerência e adaptabilidade das aplicações do *gateway* Bioinfo-Portal (<https://bioinfo.lncc.br/>) em ambientes de CAD do LNCC, pelo qual o problema será atacado em três frentes: (i) modelar um BD para o Bioinfo-Portal e acoplá-lo ao CSGrid do ambiente SINAPAD (<https://www.lncc.br/sinapad/>) e ao supercomputador Santos Dumont (SDumont, <http://sdumont.lncc.br/>); (ii) Aplicar testes de funcionalidade para validar a consistência do BD e (iii) acoplar a arquitetura do item (i) a uma interface web inteligente que consuma informações do banco de dados para serem processados e analisados *a posteriori* pelo equipe do Bioinfo-Portal, aplicando técnicas de aprendizado de máquina. Bioinfo-Portal visa a execução em larga escala de aplicações de bioinformática usando recursos computacionais paralelos e distribuídos a fim de diminuir o grande tempo de processamento das execuções e no apoio às pesquisas da comunidade científica de bioinformática.

OBJETIVOS

Os objetivos do presente Projeto de IC são: (1) implementar um modelo de Banco de Dados conceitual no apoio ao Bioinfo-Portal e suportado por tecnologias de SGBD, acoplado aos recursos computacionais do SINAPAD e da infraestrutura do Santos Dumont, (2) implementar funcionalidades de gerência dos dados de proveniência gerados pelas aplicações do portal por meio de consultas no banco de dados e (3) aplicar testes de funcionalidade para validar a consistência do banco de dados.

METODOLOGIA

Ao ser estabelecido o modelo conceitual do BD do Bioinfo-Portal, foi possível dar início ao desenvolvimento do mapeamento dos dados para a implementação do modelo lógico. O banco de dados foi modelado com o programa dbdesigner. O projeto físico foi iniciado utilizando o PostgreSQL v10 como Sistema de Gerência de Banco de Dados (SGBD) relacional Open Source e pgAdmin v5.2 como plataforma de desenvolvimento e gerência Open Source, muito usada e com mais recursos para o PostgreSQL.

RESULTADOS E DISCUSSÃO

Sobre o desenvolvimento do BD, primeiramente foi necessário o levantamento bibliográfico sobre a gerência e modelos de banco de dados, com foco no estudo e implementação da entidade-relacionamento. A Figura 1 a seguir mostra o modelo desenvolvido e as respectivas entidades, que foram acopladas e desenvolvidas no presente Projeto de IC, modelo realizado pelo aplicativo DbSchema. Por exemplo, a entidade *arquivo (files)* se refere a entrada que será consumida por uma aplicação e está relacionada à entidade *execução (execution)*. A Figura 2 apresenta todas as tabelas (entidades) que foram desenvolvidas para armazenar os dados de proveniência do portal.

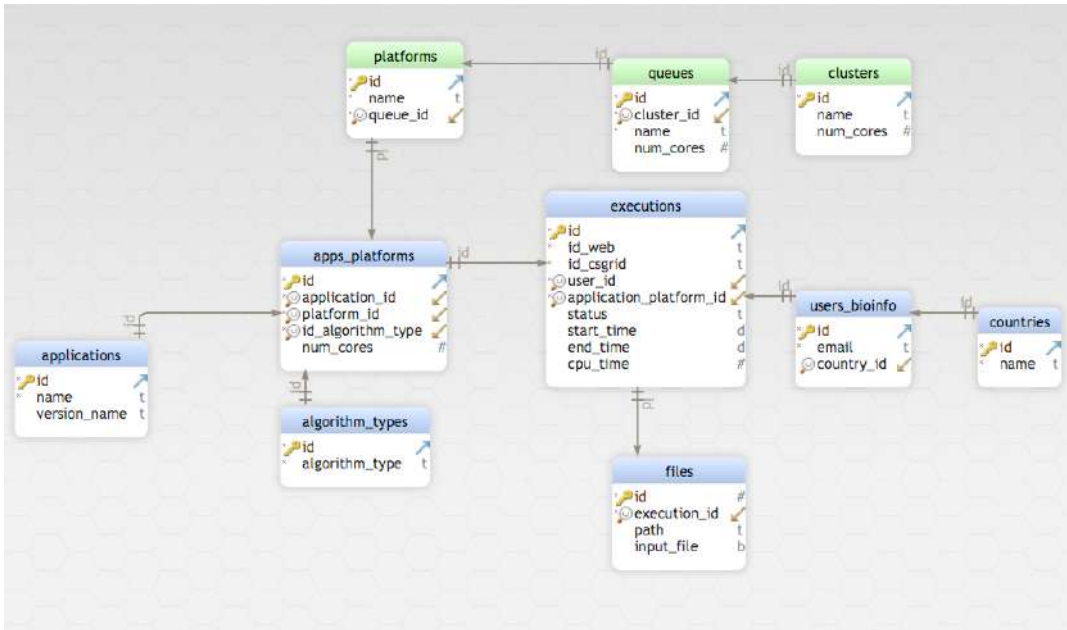


Figura 1. Modelo entidade-relacionamento desenvolvido para o Bioinfo-Portal.

Estudo de Caso

- Mapeamento do modelo conceitual ER: *Arquivo* e *Execução* são as únicas entidades existentes (originais) no Portal-Bioinfo (Figura 1). Todas as demais entidades da nova versão do BD estão disponibilizadas no servidor de desenvolvimento no SINAPAD.

Arquivo (tipo, nomeArquivo, caminho, tamanho); *Execução* (estadoFinal, tempo de execução); *Usado em* (nomeArquivo, estadoFinal, tipo de uso)

- *Arquivo* é responsável por armazenar as informações do dado de entrada, tais como: tipo de entrada, nome e tamanho do arquivo e o local de armazenamento do mesmo
- *Execução* armazena informações de tempo de execução (início e fim) e estado da execução (em espera, executando, finalizado com erro e finalizado sem erro)
- *Usado em* é o relacionamento entre as entidades *Arquivo* e *Execução*

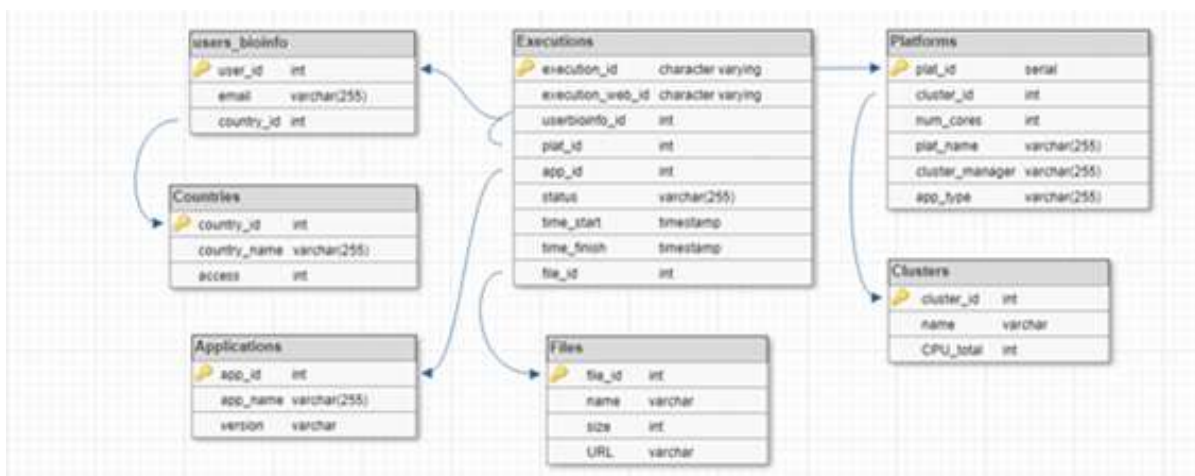


Figura 2. Tabelas do modelo do Portal-Bioinfo sendo testadas no presente projeto

O BD testado neste projeto de IC foi criado e até o presente relatório foram testadas tendo como base as tabelas da Figura 2, criadas na plataforma online dbdesigner. As Figuras 1 e 2 são tentativas de armazenamento dos dados do BD do Bioinfo-Portal, devido a que mais metadados estão sendo gerados e, por tanto, mais entidades precisam serem modeladas e integradas na versão BD final.

Nesse desenvolvimento foram criadas e mapeadas as tabelas da Figura 2, seus relacionamentos, assim como também funções de preenchimento utilizados para facilitar inserções, como nas consultas a seguir com comandos de inserção na tabela *executions*:

```
CREATE OR REPLACE FUNCTION
insert_execution(execution_id varchar, execution_webid varchar, app_id int,
file_id int, plat_id int, status varchar, time_start time, time_finish time)
  RETURNS text AS $$
  BEGIN
    INSERT INTO executions (execution_id, execution_webid, app_id,
file_id, plat_id, status, time_start, time_finish) VALUES (execution_id,
user_bioinfo_id, cluster_id, app_id, file_id, status, time_start,
time_finish);
    RETURN 'Execução cadastrada com sucesso';
  END;
$$ LANGUAGE plpgsql;

Para utilizar a função:
SELECT insert_execution('a','b',1,1,1,'', '22:30:00', '22:32:47');
```

Onde os valores dentro dos parênteses ainda são apenas para teste, preenchendo os atributos *execution_id*(chave primária), *execution_webid*(id proveniente do portal), *app_id*(chave estrangeira), *file_id*(chave estrangeira), *plat_id*(chave estrangeira) *status*(estado da execução), *time_start*(hora de início da execução), *time_finish*(hora do fim da execução).

A partir do acesso aprovado ao VPN do LNCC, foi possível acessar a máquina onde o postgresql está instalado via SSH e implementar o modelo de banco de dados como mostrado na Figura 1 e Figura 2 ao banco de dados Bioinfo-Portal, pelos códigos a continuação, possibilitando fazer alterações no banco, como: excluir, novo banco, tabela e etc.

Acesso a máquina que está instalado o postgresql:

```
ssh -L 5433:postgresql.sinapad.lncc.br:5432 sinapad@postgresql.sinapad.lncc.br
```

Acessando o Banco do Portal:

```
sudo -u postgres psql
```

```

Last login: Thu Jul 8 01:52:47 on ttys000
/Users/marcoantonio/Applications/scripts/runpsql.sh; exit
iMac-de-Marco:~ marcoantonio$ /Users/marcoantonio/Applications/scripts/runpsql.sh; exit
Server [localhost]: postgresql.sinapad.lncc.br
Database [postgres]: Bioinfo
Port [5432]: 5432
Username [postgres]: postgres
psql (9.6.22, server 12.6)
WARNING: psql major version 9.6, server major version 12.
Some psql features might not work.
Type "help" for help.

Bioinfo=# \l
          List of databases
-----+-----+-----+-----+-----+-----+
 Name      | Owner  | Encoding | Collate | Ctype  | Access privileges
-----+-----+-----+-----+-----+-----+
 Bioinfo   | postgres | UTF8     | pt_BR.UTF-8 | pt_BR.UTF-8 |
 bioinfotestemarco | postgres | UTF8     | pt_BR.UTF-8 | pt_BR.UTF-8 |
 postgres | postgres | UTF8     | pt_BR.UTF-8 | pt_BR.UTF-8 |
 template0 | postgres | UTF8     | pt_BR.UTF-8 | pt_BR.UTF-8 | =c/postgres      +
                                     postgres=CTc/postgres
 template1 | postgres | UTF8     | pt_BR.UTF-8 | pt_BR.UTF-8 | =c/postgres      +
                                     postgres=CTc/postgres
(5 rows)

Bioinfo=# █

```

Figura 3: Acesso via SSH ao banco.

CONCLUSÕES

Apresentamos resultados sobre o desenvolvimento e implementação de um banco de dados acoplado à infraestrutura do Bioinfo-Portal, que permitirá armazenar dados de proveniência, por exemplo, das execuções de todas as aplicações do *gateway*. Nesse projeto salientamos que esse modelo de entidade-relacionamento para o Bioinfo-Portal foi implementado e mapeado, além de ser analisado neste projeto, o que até então não existia. Dessa maneira, o desenvolvimento e implementação de um modelo conceitual ER para o Bioinfo-Portal permitirá a garantir a consistência na gerência e acesso aos dados científicos, de proveniência e de desempenho. Esse modelo irá sustentar o acesso aos dados para estudos realizados em paralelo no Bioinfo-Portal, análises de predição via aprendizado de máquina e o desenvolvimento da interface web inteligente.

Como passos próximos estão a alimentação do BD com execuções do Bioinfo-Portal e a criação de três sistemas para integração de novas aplicações, para mapeamento de qual versão das aplicações está sendo usada e para a gerência da inserção de dados e consulta de dados para alimentar os testes de aprendizado de máquina e alocação inteligente e dinâmica dos recursos computacionais.

REFERÊNCIAS

1. Ocaña, K.A.C.S., et al. BioinfoPortal: A scientific gateway for integrating bioinformatics applications on the Brazilian national high-performance computing network. **FGCS**, Rio de Janeiro, v. 107, p. 23, Janeiro 2020.
2. KIM, S.-H. et al. A Science Gateway Cloud With Cost-Adaptive VM Management for Computational Science and Applications. **IEEE Systems Journal**, v. 11, n. 1, p. 173-185, Março 2017. ISSN 1932-8184.

3. Gesing S, Krüger J, Grunzke R, Herres-Pawlis S, Hoffmann A. Using Science Gateways for Bridging the Differences between Research Infrastructures. **Journal of Grid Computing**. 2016;14:545–57.
4. LESK, A. M. Bioinformatics. **Britannica**, Pennsylvania, Fevereiro 2019.

Aplicação de Conceitos Topológicos para Análise de Imagens

Bolsista: Mariana Dória Prata Lima

Instituição do Bolsista: Universidade Federal de Sergipe

Instituição de Bolsa: Laboratório Nacional de Computação Científica (LNCC)

Orientador: Gilson Antônio Giraldo - LNCC

Coorientador: Gastão Florêncio Miranda Junior - UFS

Tipo da Bolsa: PIBIC-LNCC

Período do Relatório: 01/12/2020 a 31/08/2021

13 de Agosto de 2021

1 Objetivo

Este projeto tem como principal objetivo explorar conceitos topológicos para análise de imagens, e interpretação robusta dos dados, partindo da aplicação de conceitos em TDA para a análise de imagens, e consequentemente o reconhecimento de padrões. A análise de imagens será realizada usando imagens de faces do banco de imagens mantido pelo Centro Universitário da FEI [6].

2 Introdução

Sistemas com grandes volumes de imagens apresentam novos desafios para a representação, processamento e extração e informação [2]. No entanto, ainda precisamos solucionar as relações de vizinhança entre as imagens para a representação via grafos e variedades. Para isso, algumas ferramentas para a análise de dados baseada em topologia são usadas para estudar estruturas fundamentais dos dados. Estamos no contexto de *Topological Data Analysis* (TDA), uma área recente em mineração de dados [3]. A topologia associada com a geometria permite uma interpretação robusta dos dados possibilitando reconhecimento de padrões.

No projeto será utilizado banco de faces humanas da FEI. As imagens estão normalizadas com resolução padrão de 360×260 pixels. O objetivo é identificar descobrir padrões e realizar uma interpretação robusta dos dados.

Enfatizamos a utilização de ferramentas baseadas em alguns conceitos topológicos, como a homologia persistente obtida usando a filtragem de complexos simpliciais e o diagramas de persistência. Utilizamos a biblioteca GUDHI [6] e a linguagem Python para a implementação de métodos topológicos básicos.

3 Fundamentação Teórica

Precisaremos enunciar algumas definições para ajudar da compreensão da metodologia proposta para análise de dados.

3.1 Complexos

3.1.1 Complexos simpliciais

[**Simplexos**] Seja u_0, \dots, u_k pontos em \mathbb{R}^d . Um ponto $x = \sum_{i=0}^k \lambda_i u_i$, com cada $\lambda_i \in \mathbb{R}$, $0 \leq \lambda_i \leq 1$ e $\sum_{i=0}^k \lambda_i = 1$, é uma combinação convexa dos pontos u_i . O envoltório convexo é o conjunto de todas as combinações convexas dos u_i . Um simplexo k-dimensional é o envoltório

convexo de $k + 1$ pontos linearmente independentes $\sigma = \text{conv}\{u_0, \dots, u_k\}$. Para as dimensões menores são usadas nomes especiais: vértice para 0-simplexo, aresta para 1-simplexo, triângulo para 2-simplexo e tetraedro para 3-simplexo [1]. Veja a Figura 1.

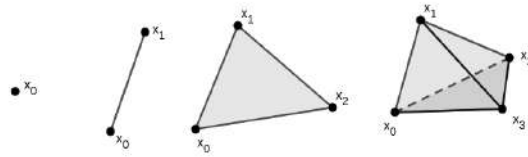


Figure 1: Da esquerda para a direita: um vértice, uma aresta, um triângulo, e um tetraedro.

Em seguida, vejamos as definições de complexo simplicial, pois sobre ele será enunciada a definição de grupo de homologia.

Definição 1. [Complexo Simplicial] Um complexo simplicial é uma coleção finita de simplexes K tal que: (a) $\sigma \in K$ e $\tau \subseteq \sigma$ implica $\tau \in K$; (b) Sejam $\sigma, \sigma_0 \in K$ então $\sigma \cap \sigma_0 = \emptyset$ ou uma face de ambos [1].

Para construir complexos simpliciais a partir dos dados, precisamos abstrair a noção de um complexo simplicial. Vejamos a definição do complexo simplicial abstrato.

Definição 2. [Complexo simplicial abstrato] Um complexo simplicial abstrato é uma coleção finita de conjuntos A tal que se $\alpha \in A$ e $\beta \subseteq \alpha$ implicam em $\beta \in A$ [1].

Vejamos abaixo mais uma definição que será usada para construção do complexo Alpha.

Definição 3. [Nervo] Seja F uma coleção finita de conjuntos. O nervo consiste em todas as subcoleções não vazias cujos conjuntos têm uma interseção comum não vazia. Assim,

$$\text{Nrv}F = \{X \subseteq F \mid \bigcap X \neq \emptyset\}$$

É sempre um complexo simplicial abstrato, não importa quais temos em F . De fato, $\bigcap X \neq \emptyset$ e $Y \subseteq X$, então $\bigcap Y \neq \emptyset$ [1].

Definição 4. [Diagrama de Voronoi] Dado um conjunto finito $S \in \mathbb{R}^d$ e u um ponto de S . A célula de Voronoi trata-se de um conjunto de pontos $V_u = \{x \in \mathbb{R}^d \mid \|x - u\| \leq \|x - v\|, v \in S\}$. O diagrama de Voronoi de S é a coleção de células de Voronoi de seus pontos [1].

[Complexo de Vietoris-Rips] Seja S um conjunto finito de pontos em \mathbb{R}^d e r um número real não negativo. O complexo de Vietoris-Rips de S consiste em todos os subconjuntos de diâmetro no máximo $2r$.

$$\text{Vietoris - Rips}(r) = \{\sigma \subseteq S \mid \text{diam} \sigma \leq 2r\}$$

Sendo o diâmetro de um conjunto no espaço euclidiano o supremo nas distâncias entre seus pontos [1].

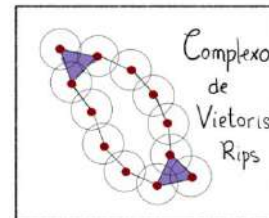


Figure 2: Exemplo do complexo de Vietoris-Rips [5].

[Complexo Alpha] Seja S um conjunto finito de pontos em \mathbb{R}^d e r um número real não negativo. Para cada $u \in S$, consideramos $B_u(r) = u + r\mathbb{B}^d$ a bola fechada com centro u e raio r . A união dessas bolas é o conjunto de pontos a uma distância no máximo r de pelo menos um dos pontos em S . Para decompor a

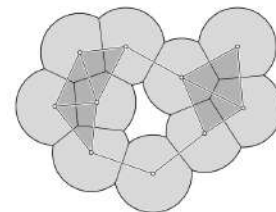


Figure 3: A união dos discos é decomposta em regiões convexas pelo Dia-

união, intersectamos cada bola com a célula de Voronoi correspondente, $R_u(r) = B_u(r) \cap V_u$. Como as bolas e as células de Voronoi são convexas, os $R_u(r)$ também são convexas. Quaisquer dois deles são disjuntos ou se sobrepõem ao longo de uma parte comum de seus limites, além disso, $S = \cup_u R_u(r)$ [1]. Veja a figura abaixo. O complexo alpha é o nervo dessa cobertura, computado como segue:

$$Alpha(r) = \{\sigma \subseteq S \mid \bigcap_{u \in \sigma} R_u(r) \neq \emptyset\}$$

3.2 Homologia

Homologia é um formalismo matemático para caracterizar aspectos topológicos de um espaço [1].

Cadeia de Complexos Seja K um complexo simplicial e p uma dimensão. Uma p -cadeia é uma soma formal de p -simplexos em K . A notação padrão para isso é $c = \sum a_i \sigma_i$, onde o σ_i são os p -simplexos e a_i são os coeficientes. Na topologia computacional, os coeficientes a_i são binários (0 ou 1), e as operações envolvendo esses coeficientes são realizadas módulo 2.

O conjunto de p -cadeia com a operação de adição forma o grupo abeliano de p -cadeias [1]. Denotamos por $(C_p, +)$, ou simplesmente $C_p = C_p(K)$. Para relacionar esses grupos, é definida a borda de um p -simplexo como a soma de suas faces $(p-1)$ -dimensionais. Escrevendo $\sigma = [u_0, \dots, u_p]$ para o simplexo com vértices nos pontos u_0, \dots, u_p , sua borda é dada por:

$$\partial_p \sigma = \sum_{j=0}^p [u_0, \dots, \hat{u}_j, \dots, u_n]$$

, onde \hat{u}_j significa exclusão do elemento u_j .

Assim, para uma p -cadeia $c = \sum a_i \sigma_i$, sua borda é $\partial_p c = \sum a_i \partial_p \sigma_i$. Escrevemos a função $\partial_p : C_p \rightarrow C_{p-1}$ a qual é um homomorfismo, pois comuta com a adição [1].

Uma cadeia de complexos é a sequência do grupo de p -cadeias conectadas pela operação de borda [1]. Denotamos

$$C : \dots \rightarrow C_{p+1} \xrightarrow{\partial_{p+1}} C_p \xrightarrow{\partial_p} C_{p-1} \rightarrow \dots \rightarrow C_1 \xrightarrow{\partial_1} C_0 \xrightarrow{\partial_0} 0.$$

[Grupo de Homologia] Seja A um anel comutativo e C um complexo de cadeias. Seja também B_p a imagem do homomorfismo de borda $\partial_{p+1} : C_{p+1} \rightarrow C_p$ e Z_p o núcleo do homomorfismo ∂_p . O grupo de homologia p -dimensional do complexo C com coeficientes em A é definido pelo conjunto $H_p(C) = Z_p / B_p = \{c + B_p; c \in Z_p\}$ [1].

3.3 Homologia Persistente

A homologia persistente estuda os invariantes topológicos de um espaço através da topologia algébrica, fornecendo informações como o número de componentes conexas por caminho e o número de buracos [1].

[Filtração] Considere K um complexo simplicial. Uma filtração é uma sequência aninhada de subcomplexos, $\emptyset = K_0 \subset K_1 \subset \dots \subset K_n = K$ que apresenta evolução topológica expressa pela sequência de grupos de homologia. Sendo $K_{i-1} \subset K_i$, uma aplicação de inclusão induz um homomorfismo entre os grupos de homologia, $f_* : H_p(K_{i-1}) \rightarrow H_p(K_i)$. A sequência aninhada de complexos, portanto, corresponde a sequências de grupos de homologias conectados por homomorfismo. Assim, $0 = H_p(K_0) \rightarrow H_p(K_1) \rightarrow \dots \rightarrow H_p(K_n) = H_p(K)$, para cada dimensão p . A medida que K_{i-1} passa para K_i , novas classes de homologia podem ser formadas ou perdidas, ou se fundir com outras [1]

[Grupo de homologia persistente] Considere K um complexo simplicial, $\emptyset = K_0 \subset K_1 \subset \dots \subset K_n = K$ uma sequência de complexos, e $K_i, K_j \subseteq K$. Para cada $0 \leq i \leq j \leq n$ temos uma aplicação de inclusão do espaço subjacente de K_i ao de K_j , e portanto um homomorfismo induzido $f_p^{i,j} : H_p(K_i) \rightarrow H_p(K_j)$, para cada dimensão p . O grupo de homologia persistente p -dimensional, denotado por $H_p^{i,j}$, é a imagem do homomorfismo induzido, ou seja, $H_p^{i,j} = \text{im } f_p^{i,j}$ [1].

[Diagramas] O diagrama de Persistência e o diagrama do código de barras são representações do intervalo $[i, j]$, sendo i o tempo de nascimento na filtração de um $u \in H_p(K_i)$ e j o tempo da morte na filtração de um $u \in H_p(K_j)$ com $j > i$, sendo $H_p(K_i)$ e $H_p(K_j)$ grupos de homologia. No nascimento, u não é imagem de H_{p-1} sobre $f_p^{i-1,i} : H_p(K_{i-1}) \rightarrow H_p(K_i)$. Na morte j deve ser o menor índice tal que $f_p^{i,j} = 0$.

O diagrama de persistência representa cada intervalo como um ponto (i, j) no plano. Já o diagrama do código de barras representa através de barras o intervalo $(i, j]$.

4 Metodologia

Primeiramente, foi realizado um estudo dos conceitos matemáticos tais como espaço métrico, espaço topológico, simplexes, homotopia e homologia, com ênfase na homologia persistente que inclui grupos de homologia, filtração de simplexes e diagrama de persistência, presentes em [1, 5].

A partir de uma nuvem de pontos podemos obter informações utilizando alguns algoritmos em TDA. Nesse trabalho, essa nuvem de pontos corresponde ao banco de imagens de faces da FEI. Essas imagens estão normalizadas possuindo uma orientação padrão, tendo sido geradas originalmente com resolução de 360×260 pixels. Neste trabalho, as imagens foram convertidas para tons de cinza (vide Figura 4).

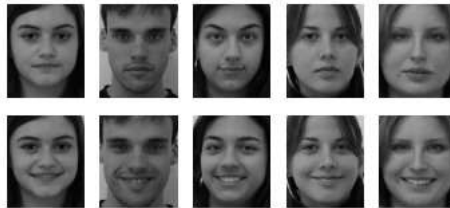


Figure 4: Alguns exemplos de imagem do banco de dados de faces da FEI.

Em nossos testes, as imagens foram redimensionadas para o tamanho 60×50 pixels, desde modo, cada imagens pode ser representada por um elemento em $\mathbb{R}^{60 \times 50}$, estamos no contexto de \mathbb{R}^n , com $n = 3000$. Nesse caso, as técnicas de visualização são limitadas. Para isso, aplicamos uma técnica de redução de dimensionalidade, o Multidimensional Scaling (MDS) [7], que preserva a distâncias entre pontos. Foram utilizadas 60 imagens de faces humanas de ambos os sexos (30 do sexo feminino e 30 do sexo masculino), uma com uma expressão neutra ou não sorridente e a outra com uma expressão facial sorridente.

A próxima etapa foi a implementação utilizando a biblioteca GUDHI e a linguagem Python. Nessa etapa, as nuvens de pontos foram fornecidas como entrada para a criação dos complexos de Vietoris-Rips e o complexo Alpha. A redução de dimensionalidade possibilitou a visualização desses complexos na dimensão 2 e 3 (vide seção 5). Por fim, com a filtração é possível obter o diagramas de persistência e o diagrama o código de barras de cada complexo simplicial, nessa etapa também utilizamos a biblioteca GUDHI.

5 Resultados e Discussão

A nuvem de pontos utilizadas na implementação, corresponde ao banco de imagens da FEI. Primeiro, utilizamos a biblioteca SCIKIT-LEARN [7] para obter a redução de dimensionalidade

desse conjunto de pontos. Com a dimensionalidade reduzida, geramos as filtrações do complexo de Vietoris-Rips e complexo Alpha. Com a biblioteca GUDHI é possível visualizar o diagramas de persistência e o diagrama do código de barras. As imagens das Figuras 1-2 mostram os complexos de Vietoris-Rips e Alpha e seus diagramas para a nuvem de pontos obtida reduzindo a dimensão das imagens para o valor dois via MDS. As Figuras 1-3 são análogas, mas usando redução para dimensão $d = 3$.

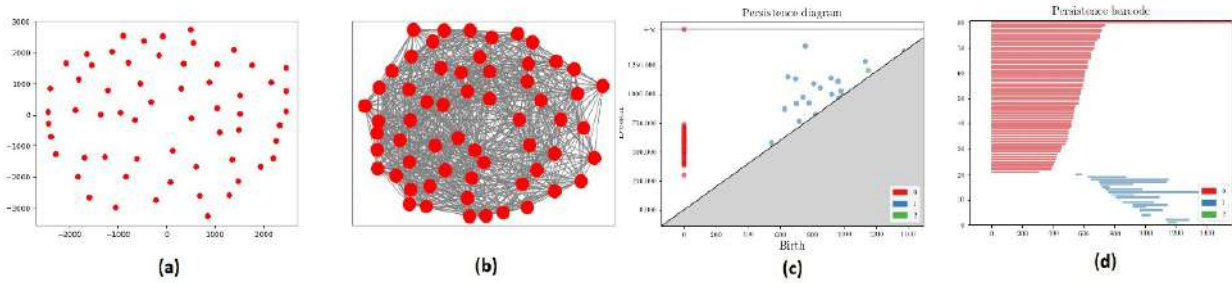


Figure 5: (a)Nuvem de pontos bidimensional; (b)Complexo de Vietoris-Rips; (c)Diagrama de Persistência; (d)Diagrama do código de barras.

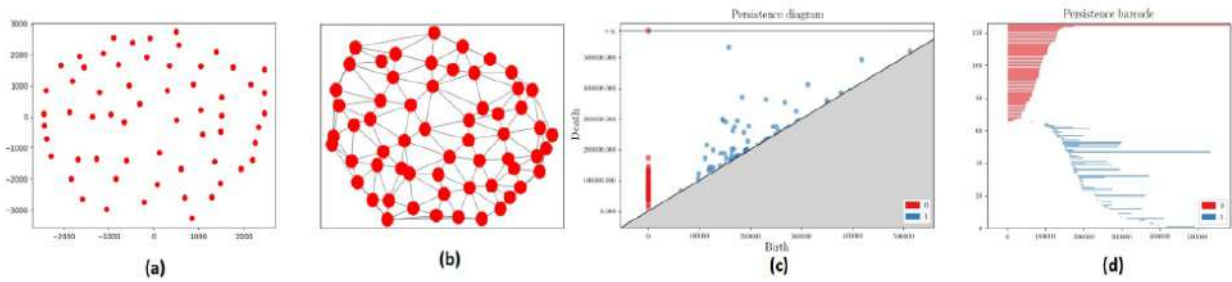


Figure 6: (a)Nuvem de pontos bidimensional; (b)Complexo Alpha; (c)Diagrama de Persistência; (d)Diagrama do código de barras.

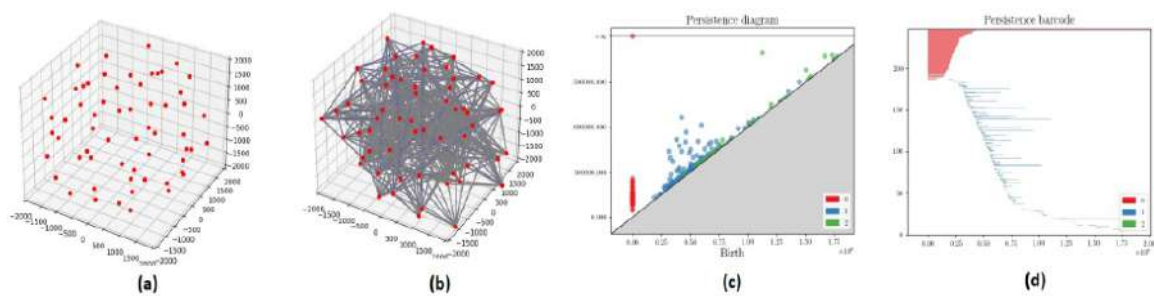


Figure 7: (a)Nuvem de pontos tridimensional; (b)Complexo de Vietoris-Rips; (c)Diagrama de Persistência; (d)Diagrama do código de barras.

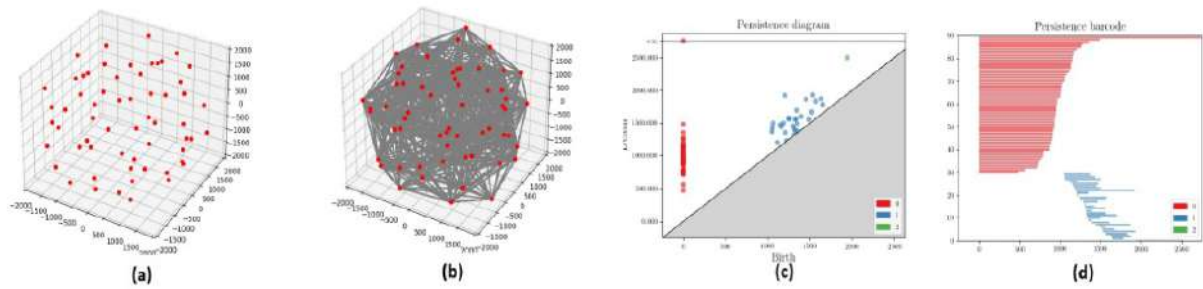


Figure 8: (a)Nuvem de pontos tridimensional; (b)Complexo Alpha; (c)Diagrama de Persistência; (d)Diagrama do código de barras.

6 Considerações Finais

Considerando os resultados nas Figuras 1-4, faz-se necessário realizar um estudo comparativo a respeito dos diagramas de persistência encontrados de forma que possamos escolher a melhor estratégia que represente a estrutura de vizinhanças bem como a redução de dimensionalidade adequada.

Neste sentido, estudo de estabilidade entre diagramas torna-se imprescindível. Para isso, vamos utilizar uma métrica que compare o quão próximos estão dois diagramas de persistências obtidos [8].

References

- [1] E. Herbert and H. John, Computational topology : an introduction, 2008.
- [2] M. Offroy and L. Duponchel, “*Topological data analysis: A promising big data exploration tool in biology, analytical chemistry and physical chemistry,*” *Analytica Chimica Acta*, vol. 910, pp. 1 – 11, 2016.
- [3] F. Chazal and B. Michel, “An introduction to topological data analysis: fundamental and practical aspects for data scientists,” 2017.
- [4] The GUDHI Project, GUDHI User and Reference Manual, 3.2.0 ed. Board, 2020. [Online]. Available: <https://gudhi.inria.fr/doc/3.2.0/> GUDHI Editorial
- [5] R. Carlos Henrique Venturi, Geradores de homologia persistente e aplicações, São Carlos (2019).
- [6] Centro Universitario da FEI, São Bernardo do Campo, São Paulo, Brazil. Link: <https://fei.edu.br/cet/facedatabase.html>
- [7] Scikit-learn, Machine Learning in Python. [Online]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.manifold.MDS.html>
- [8] A. Sarit, Comparison of Persistence Diagrams, Hebrew University, Mount Scopus, Jerusalem, Israel.

Relatório de Atividades

Título: Algoritmos Quânticos

Bolsista: Matheus de Araújo Ferreira

Orientador: Renato Portugal

Tipo de bolsa: PIBIC

Período do relatório: 01/03/2021 à 31/08/2021

Objetivos

O objetivo principal do trabalho é o estudo dos fundamentos e princípios da Computação Quântica, por meio do estudo dos fundamentos de Álgebra Linear utilizados na teoria, da origem da Computação Quântica, da conceituação de circuitos quânticos e portas lógicas quânticas, e do estudo do funcionamento dos algoritmos de Deutsch, Deutsch-Jozsa e Bernstein-Vazirani.

Além disso, o objetivo secundário desta pesquisa é o desenvolvimento de um Trabalho de Conclusão de Curso (TCC) descrevendo o funcionamento do já citado Algoritmo de Deutsch-Jozsa, a ser apresentado à instituição FAETERJ Petrópolis, com entrega prevista para dezembro de 2021.

Introdução

O desenvolvimento do que hoje se conhece por Computação Quântica se iniciou em 1981, por meio do trabalho do físico Richard Feynman. Em palestra no MIT, Feynman argumentou que os computadores disponíveis à época não eram capazes de solucionar os problemas propostos pela Física Quântica com o nível de precisão exigido. Assim, o físico teorizou que um computador desenvolvido segundo os postulados da Mecânica Quântica seria o ideal para resolver os problemas desta área.

A proposta de Feynman incentivou o também físico David Deutsch a desenvolver, em 1985, uma generalização da clássica Máquina de Turing visando sua utilização em um futuro computador quântico. No mesmo ano, Deutsch também desenvolveu o modelo de circuitos quânticos e portas lógicas quânticas, que se tornou base para o desenvolvimento dos algoritmos surgidos posteriormente, como os algoritmos de Shor, Deutsch-Jozsa e Bernstein-Vazirani.

Material e Metodologia

O principal material utilizado foi a apostila “Introdução a Programação de Computadores Quânticos”, desenvolvida por Renato Portugal e Franklin Marquezino. Para o estudo dos fundamentos de Álgebra Linear, também foi

utilizado, como material secundário, o livro “Linear Algebra Done Right”, do autor Sheldon Axler.

A metodologia utilizada foi o estudo dirigido, com reuniões periódicas com o orientador e o grupo de Computação Quântica do LNCC, resolução dos exercícios propostos no material e implementação e simulação dos algoritmos descritos.

Resultados e Discussão

Durante o período acima relatado, foram estudados os seguintes tópicos, conforme estrutura da apostila referenciada anteriormente: A base da computação quântica; Origens e novidades da computação quântica; Qubit, portas lógicas e circuitos quânticos; Paralelismo quântico e o modelo padrão da computação quântica; Algoritmo de Deutsch e Algoritmo de Bernstein-Vazirani.

Foram realizadas reuniões semanais com o orientador e o grupo de Computação Quântica do LNCC, visando o acompanhamento dos trabalhos realizados e o prosseguimento da pesquisa.

Os exercícios previstos na apostila foram devidamente resolvidos. Os exercícios de implementação e testagem de algoritmos quânticos foram desenvolvidos por meio da ferramenta IBM Quantum Composer, parte da plataforma gratuita IBM Quantum.

Também foi iniciado o projeto do Trabalho de Conclusão de Curso supracitado, com a definição da temática, o Algoritmo de Deutsch-Jozsa, e o início do desenvolvimento do trabalho.

Conclusão

A partir da proposição de Feynman e dos estudos iniciais de Deutsch, a Computação Quântica se desenvolveu de maneira exponencial nas últimas décadas.

Apesar de seu surgimento recente em comparação a outros ramos da matemática e da computação, e do nível ainda inicial dos dispositivos existentes, utilizados primeiramente para pesquisa científica, a Computação Quântica já obteve resultados concretos. Gigantes do setor de tecnologia investem da ordem de milhões de dólares anuais em máquinas cada vez mais potentes, visando a utilização destes dispositivos nos mais diferentes setores no futuro, como saúde e meio-ambiente. Além disso, já existem dois casos reivindicados da chamada “supremacia quântica”, onde pesquisadores do Google, em 2019, e da Universidade de Ciência e Tecnologia da China, em 2020, conseguiram executar, em questão de minutos, cálculos de tamanha complexidade que computadores clássicos levariam milhares de anos para resolvê-los.

Assim, a Computação Quântica se apresenta como a fronteira final do desenvolvimento de dispositivos eletrônicos, prometendo revolucionar toda a área de tecnologia e entregar um nível jamais visto de desempenho.

Referências Bibliográficas

R. Portugal and F. Marquezino. **Introdução à Programação de Computadores Quânticos**. In CSBC 2019 – 38° JAI, pages 1–51, Belém – Pará, jul 2019.

AXLER, Sheldon. **Linear Algebra Done Right**. 2. ed. Nova Iorque: Springer, 1997.

Criação Automática de Árvores Arteriais em Territórios Vasculares com Ênfase na Vasculatura Renal

Rodrigo Schmidt

Resumo

No campo de pesquisa cardiovascular, sabe-se que a maioria das doenças de caráter circulatório têm início devido a uma comunicação inadequada entre o escoamento sanguíneo e as células que compõem os órgãos. Portanto, há um grande interesse para entender como ocorrem estas relações, no intuito de obter diagnósticos mais precisos. Uma das formas de estudar essas relações é através da modelagem computacional de sistemas cardiovasculares. Essa modelagem pode ser feita de várias, e uma delas utiliza-se o algoritmo *Constrained Constructive Optimization* (CCO), que permite, através de uma geração sequencial de redes, gerar árvores arteriais automaticamente com bom nível de exatidão, respeitando a fisiologia. A proposta deste trabalho é realizar uma modelagem a nível de arteríolas e capilares, utilizando-se o algoritmo supracitado, aplicada à vasculatura renal. O projeto se encontra avançando conforme o esperado, e durante o período de desenvolvimento foi possível observar que o algoritmo CCO é capaz de gerar automaticamente árvores arteriais, conforme proposto. Além disso, observou-se que os métodos utilizados para determinar a geometria da região de vascularização são capazes de realizar sua função, e podem ser utilizados na construção da geometria para este trabalho.

Sumário

1	Dados Gerais	2
2	Introdução	2
3	Objetivos	3
4	Metodologia	3
5	Resultados e discussão	4
6	Conclusões	5

1 Dados Gerais

Título do Projeto: Criação Automática de Árvores Arteriais em Territórios Vasculares com Ênfase na Vasculatura Renal

Bolsista: Rodrigo Schmidt

Orientador: Pablo Javier Blanco Tipo de Bolsa: Iniciação Científica (PIBIC)

Período de Vigência: 08/02/2021 até 08/09/2021

2 Introdução

A maioria das doenças cardiovasculares humanas têm início nas pequenas artérias, veias e capilares, onde ocorre a comunicação destes vasos com os tecidos que compõem os órgãos. Esta interação mecânico-química entre o sistema cardiovascular e os tecidos é extremamente importante para que os órgãos cumpram suas funções fisiológicas de forma correta. Caso ocorra uma perda de resposta funcional de arteríolas e capilares, algumas anomalias podem ser percebidas em vasos de maior porte, no caso veias e artérias principais, devido ao acoplamento dessa perda funcional acumulada ao longo dos vasos de menor diâmetro.

Como a expressão celular pode ocorrer de diferentes formas, deve ser feita uma abordagem de múltipla escala adequada, englobando a localização dos tecidos, as funções celulares e a estrutura vascular local. Todavia, essa abordagem não é possível com as tecnologias atuais, devido às limitações de resolução dos equipamentos utilizados nas análises. Desta forma, a modelagem computacional surge como uma opção para contornar esse problema. Com o modelo computacional construído, podem ser realizadas simulações matemáticas, obtendo parâmetros que agregam maior conhecimento sobre os fenômenos fisiológicos existentes.

No entanto, esses modelos computacionais devem respeitar os conceitos impostos pela Fisiologia e pela Mecânica dos Fluidos. Sendo assim, uma das formas de construir esses modelos é interpretar imagens médicas com alta resolução

e construir geometrias de acordo com as imagens obtidas. Com essas geometrias, utilizam-se algoritmos de vascularização automática de redes arteriais, que podem ser de dois tipos: fractais e de preenchimento de espaço. No presente trabalho, utilizou-se uma classe particular do segundo grupo de algoritmos, chamada *Constrained Constructive Optimization* (CCO), que é formulada como um problema de otimizar uma função custo, com algumas restrições impostas [1, 2, 3, 4].

3 Objetivos

O objetivo geral do projeto é pesquisar e conhecer algoritmos de vascularização automática, no intuito de aplicá-los na vascularização de domínios, respeitando as regras da Anatomia, Fisiologia e Mecânica dos Fluidos. Entre os objetivos específicos, pode-se destacar:

1. Pesquisar sobre os algoritmos de vascularização automática desenvolvidos pelo grupo de pesquisa HeMoLab;
2. Introduzir a pesquisa na modelagem do Sistema Cardiovascular humano, com ênfase na modelagem e simulação hemodinâmica do escoamento sanguíneo;
3. Pesquisar sobre a utilização dos algoritmos de vascularização automática, utilizando-se a decomposição de domínios complexos em domínios mais simples;
4. Integrar conhecimentos das áreas de Computação Científica e da Fisiologia para abordar a modelagem do funcionamento dos rins.

4 Metodologia

No presente estudo, apresentou-se inicialmente as técnicas de modelagem e de simulação computacional, utilizadas pelo Grupo de Pesquisa HeMoLab, assim como a biblioteca *Virtual Iterative Angiogenesis* (ViTA). Essa biblioteca possui uma grande quantidade de ferramentas para automatização dos algoritmos de vascularização.

Na fase inicial do estudo, foi proposta a utilização da biblioteca supracitada, abordando domínios anatômicos simples, onde todas as variáveis e funções podem ser facilmente controladas. Desta forma, torna-se mais simples o entendimento dos algoritmos. Os domínios utilizados estão disponíveis no grupo de pesquisa citado.

O projeto ainda não está em fase final. Porém, é proposto que esses algoritmos sejam utilizados na vascularização de domínios complexos, semelhantes aos encontrados em órgãos do corpo humano, em particular os rins. No entanto, objetiva-se utilizar técnicas de decomposição de domínios, de modo a simplificar a execução e a geração das redes arteriais.

Os dados sobre os domínios vasculares também estão disponíveis no grupo He-MoLab e já são empregados para vascularizações automatáticas. Além disso, utilizou-se alguns recursos disponibilizados pelo grupo, como por exemplo a biblioteca VItA e outras ferramentas. Deve-se destacar também a colaboração de pesquisadores do *Auckland Bioengineering Institute* (ABI), realizando reuniões semanais entre o bolsista, o orientador e o pesquisador Gonzalo Maso Talou, do ABI, o que garante um ambiente de pesquisa colaborativo e multidisciplinar, agregando maior conhecimento sobre o tema do estudo desenvolvido.

5 Resultados e discussão

Conforme dito anteriormente, o projeto ainda não se encontra na fase final. Portanto, ainda não se obtiveram os resultados finais da aplicação dos algoritmos na vascularização renal. Todavia, os algoritmos já foram utilizados na vascularização de domínios mais simples, de forma a possibilitar a análise dos resultados, bem como avaliar a sensibilidade em cada etapa do processo.

Entre os resultados, destaca-se a vascularização feita em estágios (vasos vermelhos) a partir de uma árvore inicial feita manualmente (vasos azuis). Este resultado é apresentado na Figura 1. Outro resultado que merece destaque é a geração de territórios vasculares que minimizam o funcional do tipo custo que representa o comprimento dos vasos, apresentado na Figura 2. No entanto, deve-se destacar o resultado obtido utilizando-se o algoritmo *Partitioned aDaptive Constrained Constructive Optimization* (PDCCO), que possui as mesmas funcionalidades do CCO. Porém, com esse algoritmo decompõe-se um domínio mais complexo em domínios mais simples, executa a vascularização de cada subdomínio separadamente e mescla as árvores geradas em um único território vascular. Essa vascularização foi feita a partir de uma árvore base, sem a divisão do domínio, que pode ser vista na Figura 3, e depois divide-se em quatro subdomínios vascularizados, que podem ser vistos em na Figura 4. Posteriormente, o algoritmo faz a junção desses subdomínios vascularizados com a árvore base gerada anteriormente. Este resultado pode ser visto na Figura 5.

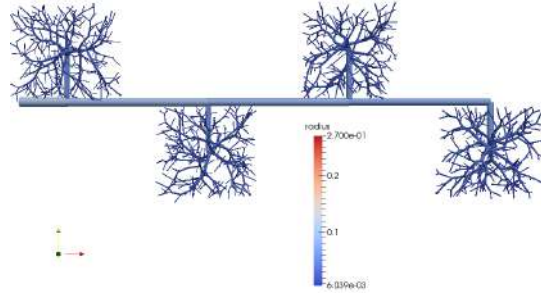


Figura 1: Vascularização em quatro estágios a partir de uma árvore inicial construída manualmente

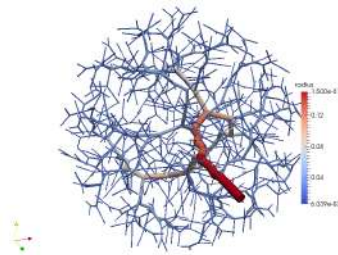


Figura 2: Vascularização com minização do comprimento dos vasos

6 Conclusões

Conforme os resultados obtidos, pode-se concluir que o método proposto é eficaz para realizar uma vascularização automática de redes de vasos, de forma consistente e que respeite as imposições e leis empíricas da Fisiologia e as Leis de Conservação da Mecânica dos Fluidos. Entre essas restrições impostas, cita-se a não vascularização em determinadas regiões determinadas, determinação do fluxo inicial, ângulo de bifurcação determinado de forma adequada, entre outros parâmetros.

Os resultados finais ainda não foram obtidos; todavia, espera-se que com a utilização dos algoritmos CCO e PDCCO possa ser possível realizar a vascularização completa de um rim, representando adequadamente todas as artérias, veias, arteríolas e capilares e compõem o sistema vascular local com eficiência. Além disso, pode-se concluir também que o estudo pode ser aplicado a outros órgãos do corpo humano, desde que respeitadas as restrições fisiológicas, que podem ser obtidas por interpretação de imagens médicas.

Desta forma, conclui-se, por fim, que o estudo realizado está de acordo com os objetivos do projeto, visto que os resultados parciais obtidos estão dentro do

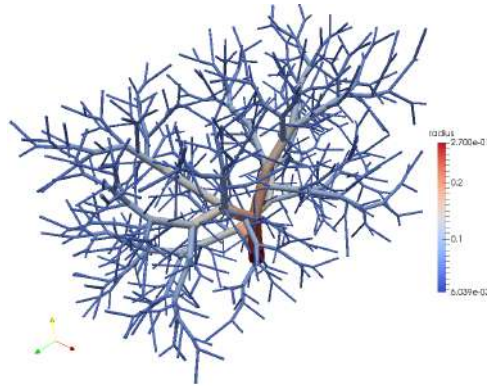


Figura 3: Vascularização da árvore inicial pelo PDCCO

esperado, validando seu desenvolvimento.

Referências

- [1] P. J. Blanco, R. A. B. Queiroz, and R. A. Feijóo. A computational approach to generate concurrent arterial networks in vascular territories. *International Journal for Numerical Methods in Biomedical Engineering*, 2013.
- [2] L. F. M. Cury. Um algoritmo paralelo para a construção de redes vasculares de grande porte. Master's thesis, Laboratório Nacional de Computação Científica, 2021.
- [3] G.D.M. Maso Talou, S. Safaei, P.J. Hunter, and P.J. Blanco. Adaptive constrained constructive optimisation for complex vascularisation processes. *Scientific Reports*, 2021.
- [4] R. A. B. Queiroz. *Construção automática de modelos de árvores circulatórias e suas aplicações em hemodinâmica computacional*. PhD thesis, Laboratório Nacional de Computação Científica, 2013.

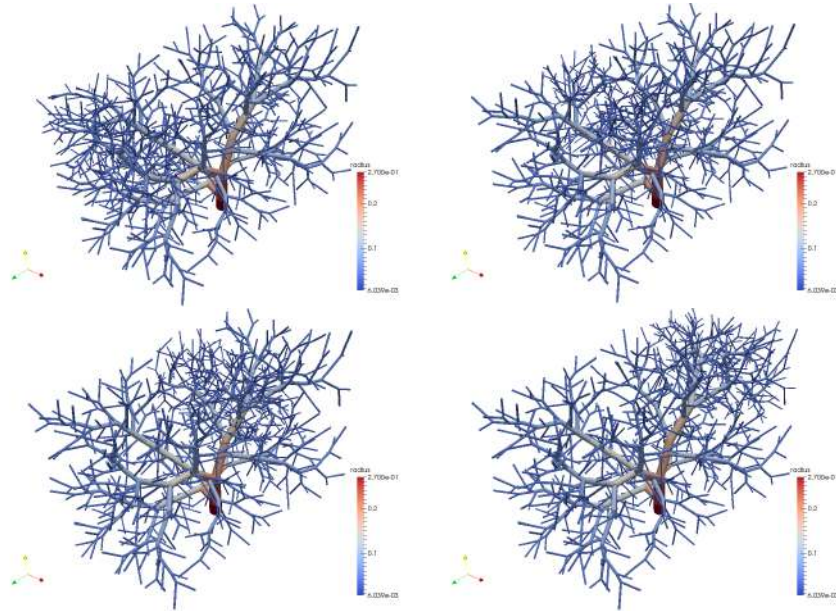


Figura 4: Vascularização dos domínios fracionados em diferentes estágios com o PDCCO

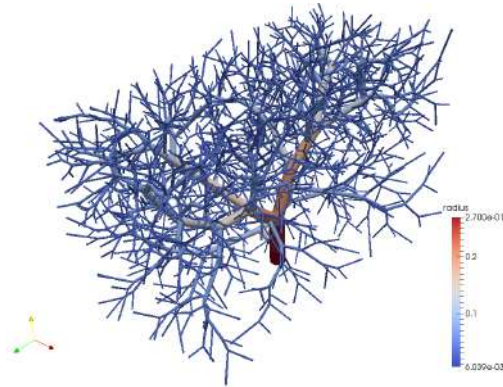
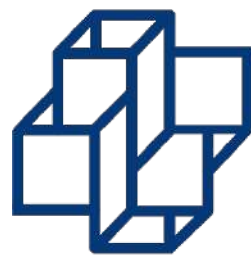


Figura 5: Resultado final da vascularização feita pelo PDCCO



Laboratório
Nacional de
Computação
Científica

Programa Institucional de Bolsas de Iniciação em Desenvolvimento Tecnológico e
Inovação

Edital 2020/2021

Relatório de Atividades

Introdução à Computação Quântica e Criptografia

Autor: Thainá Lucciola Hipolito de Lima
Orientador: Renato Portugal

Rio de Janeiro, RJ
Agosto de 2021

1. OBJETIVOS

Neste trabalho foi proposto um estudo introdutório de conceitos fundamentais da computação quântica, bem como a aplicabilidade destes na programação e seus principais resultados dentro da criptografia. Tivemos como objetivo relacionar esses conceitos com o tema de segurança da informação, discutindo algoritmos que envolvem a quebra da criptografia moderna.

2. INTRODUÇÃO

Com o passar do tempo surgem novas necessidades que guiam o desenvolvimento técnico/científico para suprir as deficiências das tecnologias de cada época. A ascensão da computação conforme conhecemos atualmente teve um fortalecimento com Alan Mathison Turing (1912 – 1954), que idealizou um dispositivo teórico que escrevia e interpretava os símbolos de 0, 1 e nulo, em uma fita infinita, com o auxílio de um cabeçote, respeitando uma função de transição. Os atuais sistemas computacionais são construídos com base na referência matemática da Máquina de Turing, seguindo a arquitetura de Von Neumann e com aperfeiçoamentos e tecnologias de cada fabricante. Como resultado dessa modelagem, limitações de processamento na Máquina de Turing se apresenta atualmente como problemas de desempenho do *hardware* dos computadores atuais. Esses problemas foram o suficiente para levantar o questionamento se ainda existiam modelos computacionais melhores em determinadas tarefas que necessitavam de muito poder computacional.

3. Metodologia

Foi elaborado um relatório com base na pesquisa realizada no semestre em que foi discutido de maneira extensa diversos conceitos, como: Notação de Dirac, Fenômenos quânticos como dualidade partícula-onda, sobreposição quântica e emaranhamento, Qubit e portas lógicas de 1 ou mais qubits (com exemplos de circuitos no *composer* da IBM), paralelismo quântico e três importantes algoritmos da computação quântica.

4. RESULTADOS E DISCUSSÃO

Nesse relatório de atividades temos como foco mostrar o funcionamento geral dos algoritmos quânticos, de forma que não entraremos em detalhes da análise matemática dos algoritmos.

Algoritmo de Bernstein-Vazirani [1]

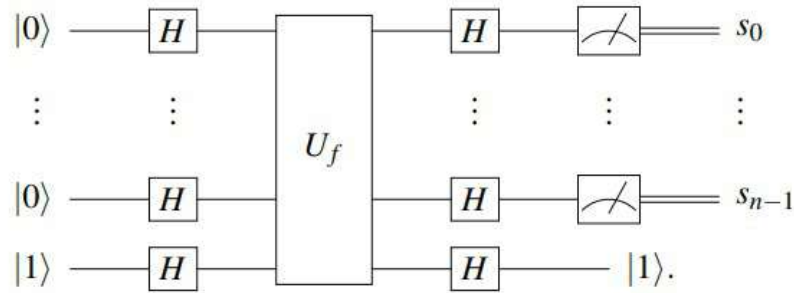
Considere uma cadeia desconhecida de n bits, denominada s . Embora não tenhamos conhecimento prévio sobre s , temos acesso a uma função booleana que possui uma dependência com a cadeia s , denominada oráculo e chamada de f . Queremos encontrar s utilizando f o mínimo possível. Resolvendo de maneira clássica, consultamos o oráculo n vezes e montamos um sistema com n equações e n incógnitas. Já na versão quântica, a ideia do algoritmo é utilizar o oráculo como uma caixa preta, cujo acesso ao subconjunto do contradomínio de f é feito para alguns valores selecionados do domínio, e aproveitar a particularidade da computação quântica para determinar a sequência s com poucos acessos a função.

A cadeia s é definida por $s = s_0s_1\dots s_{n-1}$ e a função f , tal que $f: \{0,1\}^n \rightarrow \{0,1\}$, em que f é definida por $f(x) = x \cdot s = x_0s_0 + \dots + x_{n-1}s_{n-1} \pmod 2$.

Vale-se ressaltar que como não temos acesso a s (queremos descobrir s utilizando o algoritmo), uma terceira pessoa implementa a função f de modo que não seja acessível para gente. Por isso é chamada de caixa preta, porque podemos usar a função, mas não sabemos como ela foi implementada. Dessa forma, definimos que um operador U_f de $n+1$ qubits irá utilizar f e prover o resultado necessário para prosseguir o cálculo. Resolvendo esse problema com a computação quântica, só precisamos consultar o oráculo U_f uma única vez.

U_f é um operador unitário definido por $U_f|x\rangle|j\rangle = |x\rangle|j \oplus f(x)\rangle$, onde \oplus refere-se a operação soma módulo 2, x é uma sequência de n qubits e j é um qubit.

Podemos representar o algoritmo de forma visual da seguinte maneira:



Seguindo a representação visual, temos que o estado inicial é $|0\rangle^{\otimes n}|1\rangle$ e vemos que uma análise teórica da medição do estado final leva aos valores de s com probabilidade 1. Em caso de utilizar o computador quântico, devidos aos erros provenientes do computador, essa probabilidade não atinge o valor 1.

Algoritmo de Grover [1]

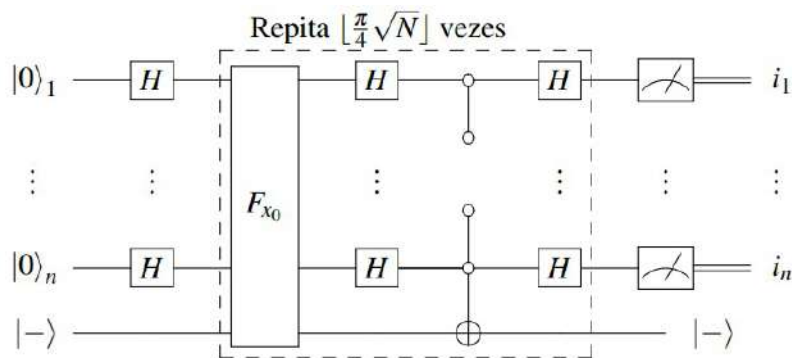
O algoritmo de Grover é um algoritmo quântico utilizado para busca em banco de dados não ordenados, cuja complexidade é $O(N^{1/2})$ [2]. A diminuição da complexidade em comparação a algoritmos clássicos de busca é obtida através do hardware quântico que explora a superposição de estados e testa mais de uma entrada para uma função. Formulando o problema de maneira formal:

Dada uma função booleana $f: \{0,1,2,\dots,N-1\} \rightarrow \{0,1\}$, em que N é uma potência de 2, em que $f(x)=1$ se, e somente se, $x=x_0$ para algum valor fixo x_0 (chave da busca, desconhecida inicialmente, composta de n qubits). Buscamos com o algoritmo de Grover achar x_0 utilizando f (classicamente, testaríamos todas as entradas em ordem crescente até que $f(x)$ retornasse 1). Mas, analogamente ao algoritmo de Bernstein-Vazirani, queremos consultar f o mínimo possível. Parecido com o algoritmo de Bernstein-Vazirani, temos um oráculo que implementa a função f com uma matriz unitária F_{x_0} , utilizando uma porta Toffoli generalizada ativada por x_0 . Dessa forma, como não temos acesso a chave de busca, também utilizamos F_{x_0} sem conhecer sua implementação, da mesma maneira que utilizamos U_f (solicitando a uma “terceira pessoa” a resposta). Podemos então definir a matriz F_{x_0} para o algoritmo de Grover como:

$$F_{x_0}|x\rangle|-\rangle = \begin{cases} -|x_0\rangle|-\rangle, & \text{se } x = x_0, \\ |x\rangle|-\rangle, & \text{caso contrário.} \end{cases}$$

O algoritmo de Grover tem como entrada um inteiro N e a função f(x) e sua saída é x_0 com uma probabilidade menor do que 1, que depende de N. Além disso, o qubit menos significativo da entrada de F_{x_0} é constante e igual a $|-\rangle$, em que $|-\rangle = 2^{-(1/2)} (|0\rangle - |1\rangle)$.

Podemos representar o algoritmo de forma visual da seguinte maneira:



Tendo em vista que i_1, i_2, \dots, i_n são os bits de x_0 , podemos analisar os estados intermediários do algoritmo. O estado inicial é colocado em $|0\rangle^{\otimes n}|-\rangle$ e ao se aplicar as portas Hadamard, chega-se ao seguinte estado [3]:

$$\frac{1}{N^{1/2}} \sum_{x=0}^{N-1} |x\rangle$$

Já a iteração que acontece no quadrado tracejado e após essa iteração chega-se na chave. Apesar de não chegar à chave com probabilidade 1, conforme aumenta-se o N, essa probabilidade aproxima-se de 1. De forma que repetir o processo algumas vezes fornece uma margem de resposta aceitável [3].

Algoritmo de Shor

O uso da criptografia está relacionado à proteção de informação militar, governamental, empresarial e pessoal [4]. Por conta da crescente valorização desses dados, surgiu os *crackers*. O termo é utilizado para nomear aquele que

pratica a quebra de algum sistema de segurança de maneira ilegal a fim de obter alguma vantagem em cima da pessoa que foi alvo do ataque. Os *crackers* especializados em criptografia se dedicam a quebra de chaves e a complexidade dessa tarefa depende da particularidade de cada problema. Os atuais métodos criptográficos, por exemplo, utilizam chaves públicas e privadas. O cálculo da chave privada utilizando a chave pública pode ser realizado através de operações que utilizam a fatoração de números inteiros muito grandes [4], o que dificulta um computador clássico conseguir realizar tais operações.

Entretanto, o matemático estadunidense Peter Williston Shor desenvolveu um algoritmo quântico que leva seu nome para fatorar números primos com complexidade de tempo muito inferior a complexidade demonstrada por computadores clássicos. Dessa forma, esse algoritmo contribui para a quebra de chaves criptográficas e demonstra a vulnerabilidade que todos os sistemas de informação estarão suscetíveis. O algoritmo de Shor encontra os fatores da decomposição de um número N em tempo polinomial do número de bits de entrada[5].

O problema solucionado pelo algoritmo pode ser formulado da seguinte maneira: dado um número N de n bits como entrada e dois números a e b como saída, tal que $1 < a$ e $b < C$, tais que $N = a \cdot b$, como descobrir a e b com a menor complexidade possível. Um algoritmo clássico testa todas as divisões de N por inteiros que variam de 2 até a raiz quadrada de N . Para valores pequenos de N a computação clássica retorna o resultado em um tempo satisfatório. Com N maior, o processo mais utilizado envolve o cálculo de um período sobre divisões de N . Um computador clássico demoraria 100 mil anos para realizar esse cálculo para um número de 1024 bits, em contrapartida um computador quântico levaria 4.5 minutos [5] utilizando o algoritmo de Shor.

4. CONCLUSÃO

Entre as atuais aplicações da computação quântica, uma das maiores vantagens com esse novo paradigma computacional sobre a computação clássica é a possibilidade de solucionar problemas com uma menor complexidade de espaço e de tempo. Essa atenuação temporal, se analisada sob o âmbito da criptografia,

resulta em um número muito maior de tentativas para encontrar uma chave, em um mesmo espaço de tempo, do que é permitido pela computação clássica. Dessa forma, uma tentativa de ataque em algum sistema utilizando um computador quântico, permitiria a quebra da maioria dos sistemas criptografados que usamos atualmente, como por exemplo o sistema bancário. Deste modo, a compreensão e a manipulabilidade de computadores quânticos e seus algoritmos resulta em uma eficaz proteção de dados.

5. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Portugal, P. Marquezino, F. **Programa Quântica.**
- [2] Wright, J. **Lecture 4: Grover's Algorithm.** 2015.
- [3] Dillenburg, R. **O Algoritmo de Grover.** Porto Alegre, RS, 2014.
- [4] Vieira, L. **Um estudo passo a passo do Algoritmo de Shor.** Uberlândia, MG, 2019.
- [5] Pamplona, S. **Aplicação da computação quântica na resolução de problemas computacionais.**

Relatório parcial das metas alcançadas na pesquisa.

Título: Fatoração rápida via Método das Curvas Elípticas

Área CNPq: 1.03.02.00-0 Matemática da Computação

Bolsa: Iniciação Científica

Bolsista: Thaís Henter de Melo Dias

Orientador: Fábio Borges de Oliveira

Instituição: Laboratório Nacional de Computação Científica (LNCC)

1 Introdução

Com o advento das tecnologias de comunicação e o compartilhamento de dados em nuvens de computadores na internet a segurança de vários métodos de chave pública depende da dificuldade em fatorar um módulo criado a partir da multiplicação de grandes números primos. O algoritmo RSA é um bom exemplo disso, onde temos dois grandes números primos (p e q) e os multiplicamos para criar um módulo (N). A dificuldade é então ser capaz de encontrar p e q , se soubermos N . Os dois métodos principais que podemos usar para esta fatoração são o método General Number Field Sieve (GNFS) e o método da curva elíptica (ECM). Neste trabalho iremos abordar o método ECM para estudarmos suas vantagens e limitações como método para fatoração de números inteiros gigantes.

2 Objetivos

O objetivo principal é o desenvolvimento de um código em Python que utilize o método da curva elíptica (ECM) baseado na curva de Lenstra para fatoração rápida de inteiros. Os objetivos específicos seguem abaixo:

- Desenvolver habilidades com as disciplinas de teoria dos números e álgebra abstrata [3] e [6].
- Desenvolver habilidades com a programação em Python [4].
- Desenvolver habilidades com curvas elípticas na forma de Weierstrass [7], [9] e [1].
- Criar um código para operar com pontos de uma dada curva elíptica E em um corpo de Galois GF .
- Desenvolver o código para o ECM.
- Realizar testes e submeter seus resultados em um artigo de final de projeto.

3 Metodologia

Neste primeiro semestre de estudos, buscamos bibliografias sobre o tema, tivemos aulas expositivas semanais sobre teoria dos números, álgebra abstrata e curvas elípticas, bem como introdução à linguagem de programação Python. Posteriormente, iniciamos o estudo da curva de Lenstra, que é de suma importância para o ECM devido a sua natureza algébrica. Os objetivos desse primeiro semestre eram prover base teórica para a melhor compreensão do ECM.

4 Resultados e discussão

No decorrer da pesquisa obtive conhecimentos sobre as curvas elípticas, tópico geralmente não abordado em um curso universitário comum com a profundidade a qual foi apresentada, na qual estudei suas definições, alguns teoremas associados às curvas e equações reduzidas de Weierstrass para curvas em corpos algébricos de características diferentes de 2 e 3.

As curvas elípticas têm uma forte aplicação na criptografia devido ao pequeno tamanho de suas chaves, desde a criptografia de mensagens (ECC) até a autenticação digital (ECDSA) passando por esquemas de troca de chaves (SIDH), imbuídas de uma função racional conhecida como isogenias, elas se tornam resistentes aos ataques de um computador quântico, o que nos mostra o grau de importância das curvas.

A disposição desse estudo é saber como usar essas curvas para fatoração de números inteiros gigantes e analisar sua viabilidade, se são realmente eficazes e, caso positivo, como devemos nos proteger de ataques. O método de fatoração

mais rápido conhecido é o *General Number Field Sieve* (GNFS) o qual fatora um inteiro N com tempo de execução de $O(\exp(c + o(1)) \sqrt[3]{(\ln N)(\ln \ln N)^2})$ [7] e [5], mas se N tem um menor fator primo p então o tempo de fatoração via curva de Lenstra, ECM, é $\exp(c\sqrt{(\log p)(\log \log p)})$ passos [7], o que o torna um algoritmo de fatoração sub-exponencial. ECM é o melhor método para fatorar inteiros na faixa de 200 bits com fatores primos de até 40 bits [2]. Uma combinação dos métodos GNFS e ECM permite a criação de um algoritmo com fatoração rápida em função do tamanho, em bits, do número dado.

5 Metas Futuras

Os próximos passos são:

- Estudo mais detalhado da curva de Lenstra e sua relação com a fatoração de números inteiros.
- Implementação computacional do método ECM via Python.
- Análise do código e seus resultados.

6 Referências bibliográficas

[1] Cohen, Henri, Frey, Gerhard et al. 2006. Handbook of Elliptic and Hyper-elliptic Cryptography. New York: CRC Press.

[2] Gaj, K., et al. 2006. Implementing the Elliptic Curve Method of Factoring in Reconfigurable Hardware. Lecture Notes in Computer Science, 4249.

[3] Garcia, A., Yves, Lequain. 2008. Elementos de Algebra. Rio de Janeiro: IMPA.

[4] Langtangen, Hans P. 2016. A Primer on Scientific Programming with Python. London: Springer.

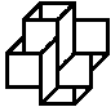
[5] Lara, Pedro C. da S., Oliveira, Fabio B. 2008. O Algoritmo de Fatoração GNFS. XXXI Congresso Nacional de Matemática Aplicada e Computacional.

[6] Santos, Jos e Plínio de O. 2010. Introdução á Teoria dos Numeros. Rio de Janeiro: IMPA.

[7] Silverman, Joseph H. 2000. The Arithmetic of Elliptic Curves. London: Springer.

[8] Stallings, William. 2015. Criptografia e segurança de rede: Princípios e práticas. São Paulo: Pearson.

[9] Washington, Lawrence C. 2008. Elliptic Curves - Number Theory and Cryptography. New York: CRC Press.



Laboratório
Nacional de
Computação
Científica

Projeto de Pesquisa de Iniciação Científica

Programa Institucional de Bolsas
de Iniciação Científica - PIBIC

Inversão de Dados Sísmicos

Bolsista: Vinícius Theobaldo Jorge
Orientador: Marcio Rentes Borges

Petrópolis-RJ
12 de agosto de 2021

Sumário

1	Identificação	1
2	Objetivos	1
3	Introdução	1
4	Metodologia	2
4.1	Método de Diferenças Finitas	3
5	Resultados e Discussão	3
6	Conclusões	5

1 Identificação

- **Título do projeto:** Inversão de Dados Sísmicos
- **Bolsista:** Vinícius Theobaldo Jorge
 - Aluno do curso de graduação de Geologia da UFRJ.
 - **E-mail:** *vinicius.tj@gmail.com*
- **Orientador:** Marcio Rentes Borges
 - Pesquisador Adjunto do Laboratório Nacional de Computação Científica (LNCC).
 - **E-mail:** *mrborges@lncc.br*
- **Bolsa:** PIBIC
- **Período do relatório:** julho de 2020 a agosto de 2021

2 Objetivos

O objetivo principal deste projeto foi a inversão de dados sísmicos utilizando uma abordagem estocástica (?). Entretanto, este o objetivo se mostrou bastante ambicioso e foi atingido apenas parcialmente com a construção de um programa em linguagem FORTRAN desenvolvido para aproximar a solução da equação da onda em domínios tridimensionais heterogêneos, usando o método de diferenças finitas.

3 Introdução

A geofísica desempenha papel crucial na descoberta de novos reservatórios de petróleo e, ainda hoje, o principal uso de dados sísmicos é identificar a geometria dos refletores. Isso é possível porque as ondas sísmicas são refletidas nas interfaces entre materiais de diferentes propriedades acústicas (?). Entretanto, recentemente, estudos sísmicos tem sido usados para a caracterização das rochas e monitoramento dos reservatórios (sísmica 4D), pela transformação de dados de reflexão em propriedades de rochas. Portanto, a inversão de dados sísmicos é uma ferramenta essencial na determinação das propriedades elásticas do subsolo.

O problema de inversão de dados geofísicos pode ser resolvido de forma determinística ou estocástica. Devido ao fato dos dados serem incompletos a inversão é um problema tipicamente mal posto (não possui solução única) e, portanto, na abordagem determinística alguma regularização deve ser imposta na função objetivo introduzindo uma “suavização” não realista na solução. Para quantificar essa ausência de unicidade de solução, métodos estocásticos podem ser utilizados. Por exemplo, a introdução dinâmica dos dados nos modelos pode ser formalizada em termos de métodos Bayesianos e simulações de Monte Carlo via cadeias de Markov (MCMC - *Markov Chain Monte Carlo methods*, veja ??). A inferência Bayesiana é conveniente na quantificação da informação inserida no modelo a partir de diversas fontes. Já os métodos MCMC por sua

vez, fornecem uma estrutura computacional para o processo de amostragem necessário à inferência Bayesiana.

No processo de inversão de dados sísmicos, utilizando uma abordagem estocástica (?), é imprescindível o uso de um simulador do comportamento das ondas sísmicas em subsuperfície. Desta forma, foi desenvolvido um código computacional para desempenhar este papel.

4 Metodologia

O modelo físico considerado é um meio elástico heterogêneo pelo qual o som é propagado através de pequenas vibrações elásticas. O modelo matemático para descrever as deformações em um meio elástico é baseado na segunda lei de Newton:

$$\rho \frac{\partial^2}{\partial t^2} \vec{u} = \nabla \cdot \boldsymbol{\sigma} + \rho \vec{f}, \quad (1)$$

e na lei constitutiva linear (Lei de Hooke) que relaciona o tensor de tensões ($\boldsymbol{\sigma}$) e o campo de deslocamentos (\vec{u}):

$$\boldsymbol{\sigma} = K \nabla \cdot \vec{u} I + G \left[\nabla \vec{u} + (\nabla \vec{u})^T - \frac{2}{3} \nabla \cdot \vec{u} I \right], \quad (2)$$

onde, I é o tensor identidade, ρ é massa específica do meio, \vec{f} representa as forças de corpo, K é o módulo elástico e G o módulo de cisalhamento.

Assumindo que o segundo termo à direita da Lei de Hooke, Eq. (2), representando as deformações que dão origem as tensões de cisalhamento, e as forças de corpo podem ser negligenciados, temos, a partir das Eq. (1) e Eq. (2):

$$\rho \frac{\partial^2}{\partial t^2} \vec{u} = \nabla (K \nabla \cdot \vec{u}). \quad (3)$$

Em seguida, introduzindo a pressão como:

$$p = -K \nabla \cdot \vec{u} \quad (4)$$

e dividindo Eq. (3) por ρ , temos

$$\frac{\partial^2}{\partial t^2} \vec{u} = -\frac{1}{\rho} \nabla p. \quad (5)$$

Agora, tomando a divergência desta última equação, usando o resultado dado pela Eq. (4) e adicionando um termo de fonte (φ), temos a equação linear da onda

$$\frac{\partial^2}{\partial t^2} p(\vec{x}, t) = K \nabla \cdot \left[\frac{1}{\rho(\vec{x})} \nabla p(\vec{x}, t) \right] + \varphi(\vec{x}, t). \quad (6)$$

Finalmente, assumindo que o gradiente da massa específica (ρ) pode ser negligenciado, temos que:

$$\frac{\partial^2}{\partial t^2} p(\vec{x}, t) = c^2(\vec{x}) \nabla^2 p(\vec{x}, t) + \varphi(\vec{x}, t), \quad (7)$$

onde, $c(\vec{x}) = \sqrt{K(\vec{x})/\rho(\vec{x})}$ é a velocidade de propagação, K é o módulo elástico e φ o termo de fonte. Em conjunto com a Eq. (7), condições iniciais e de contorno devem ser fornecidas.

Foi implementado e testado um código computacional, escrito em FORTRAN 90, para aproximar a solução da equação (7), em um domínio tridimensional, utilizando o método de diferenças finitas.

4.1 Método de Diferenças Finitas

Para aproximar a solução da Eq. (7) foi utilizado o método de diferenças finitas centradas de segunda e quarta ordens no espaço (ambos de segunda ordem no tempo). O domínio temporal $[0, T]$ é discretizado por um número Nt finito de pontos igualmente espaçados: $0 = t^0 < t^1 < \dots < t^{Nt-1} < t^{Nt} = T$, com $\Delta t = t^n - t^{n-1}$. De forma similar, discretizamos as componentes espaciais, onde $\Delta x = x_i - x_{i-1}$, $\Delta y = y_j - y_{j-1}$ e $\Delta z = z_k - z_{k-1}$. Definindo $P_{i,j,k}^n$ como uma aproximação para $p(x_i, y_j, z_k, t^n)$, onde os índices i, j, k se referem aos pontos da malha nas direções x, y, z , respectivamente, enquanto índice n representa o número do passo de tempo da simulação. A seguir, apresentamos a aproximação obtida usando o método de segunda ordem:

$$\begin{aligned} P_{i,j,k}^{n+1} = & C \left(\frac{\Delta t}{\Delta x} \right)^2 [P_{i+1,j,k}^n - 2P_{i,j,k}^n + P_{i-1,j,k}^n] + \\ & C \left(\frac{\Delta t}{\Delta y} \right)^2 [P_{i,j+1,k}^n - 2P_{i,j,k}^n + P_{i,j-1,k}^n] + \\ & C \left(\frac{\Delta t}{\Delta z} \right)^2 [P_{i,j,k+1}^n - 2P_{i,j,k}^n + P_{i,j,k-1}^n] + \\ & 2P_{i,j,k}^n - P_{i,j,k}^{n-1} - \Delta t^2 \delta(i, i_\varphi) \delta(j, j_\varphi) \delta(k, k_\varphi) \varphi^n \end{aligned} \quad (8)$$

onde $C = c_{i,j,k}^2 = \left(\frac{K(x_i, y_j, z_k)}{\rho(x_i, y_j, z_k)} \right)^2$. Os pontos $i_\varphi, j_\varphi, k_\varphi$ indicam a posição da fonte nos nós da malha, portanto, $\delta(\alpha, \alpha_\varphi) = 1$ caso $\alpha = \alpha_\varphi$ e $\delta(\alpha, \alpha_\varphi) = 0$, caso contrário, onde $\alpha = i, j, k$.

5 Resultados e Discussão

Nos relatórios anteriores, diversos experimentos em uma, duas e três dimensões foram apresentados. Estes experimentos comprovaram a efetividade do código gerado.

5 RESULTADOS E DISCUSSÃO

Neste último relatório, apresentamos dois experimentos bidimensionais para mostrar a implementação dos “geofones” ao código que tem por objetivo registrar a passagem das ondas em pontos específicos do domínio. Tais registros (sismogramas) são usados no processo de inversão de dados.

Os cenários de simulação são idênticos para os dois problemas considerados a menos dos campos de velocidades que são mostrados na Figura 2. Consideramos um domínio bidimensional de $1000m \times 1000m$, discretizado em uma malha de 500×500 pontos, no qual uma fonte sísmica foi posicionada na região central da superfície $(x, y) = (500m, 1000m)$ e 50 geofones instalados na superfície, conforme mostrado na Figura 1.

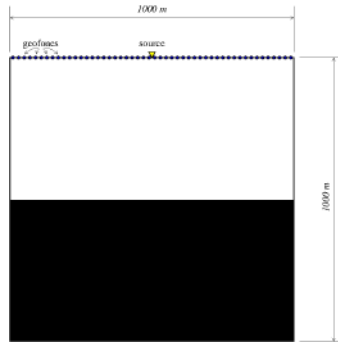


Figura 1: Domínio.

Matematicamente, a fonte é descrita por

$$\varphi(\vec{x}, t) = [1 - 2\pi(\pi f_c t_d)^2] e^{-\pi(\pi f_c t_d)^2}, \quad (9)$$

onde $f_c = f_r/3\sqrt{\pi}$ é a frequência de corte, $t_d = t - 2\sqrt{\pi}/f_r$ e f_r a frequência da onda (neste exemplo, $f_r = 60Hz$). Nas fronteiras do domínio foram consideradas condições de contorno absorventes.

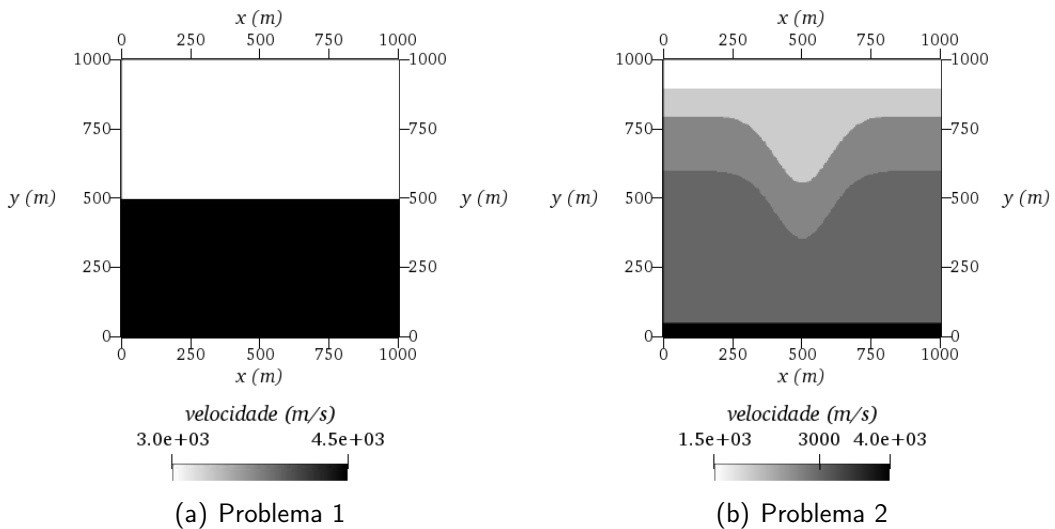


Figura 2: Velocidades de propagação dos domínios utilizados.

As Figuras 3 e 4 mostram a propagação da onda sísmica para os Problemas 1 e 2, respectivamente. É possível observar os fenômenos de transmissão e reflexão toda vez que a onda atinge uma camada material com velocidade de propagação diferente.

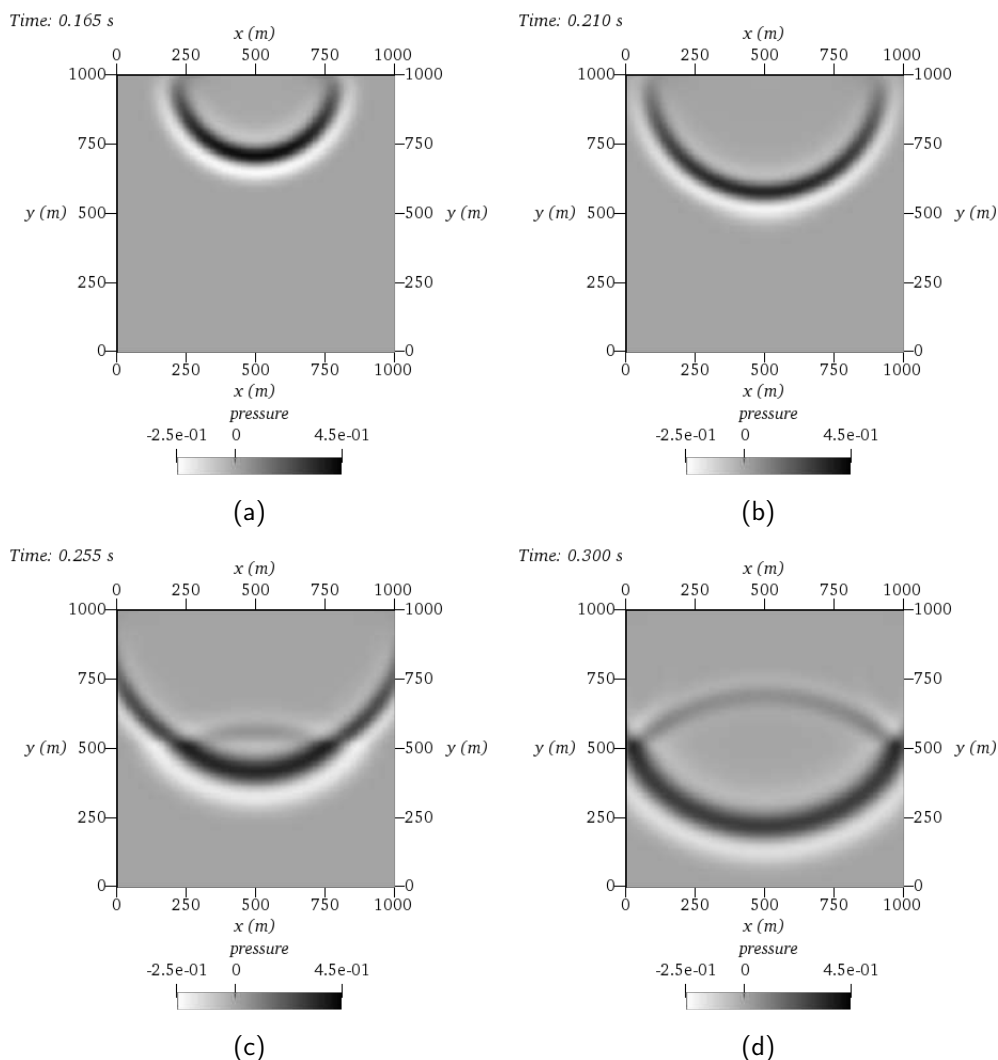


Figura 3: Pressões em diferente instantes de tempo para o Problema 1.

As pressões registradas nos geofones ao longo do tempo de simulação (sismogramas) são apresentadas na Figura 5. Podemos ver a maior complexidade dos registros do sismograma do Problema 2 causada pelo seu campo de velocidades.

6 Conclusões

O método de diferenças finitas implementado em nosso código computacional proporcionou uma boa aproximação da solução da Eq. (7), capturando o comportamento físico esperado para diversos cenários testados em 1, 2 e 3 dimensões. Com isso obtivemos um programa que está qualificado para usar dados reais de estudos relacionados a toda cadeia produtiva do petróleo.

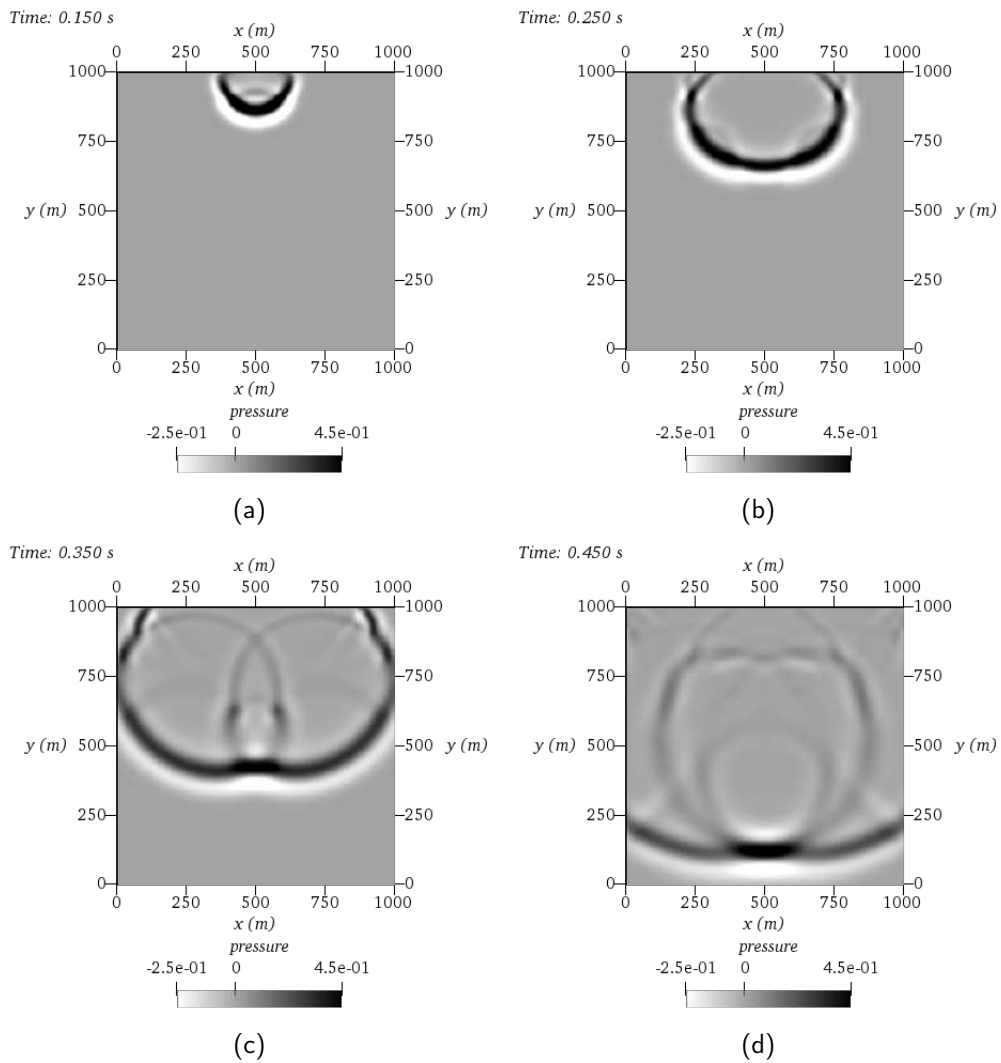


Figura 4: Pressões em diferente instantes de tempo para o Problema 2.

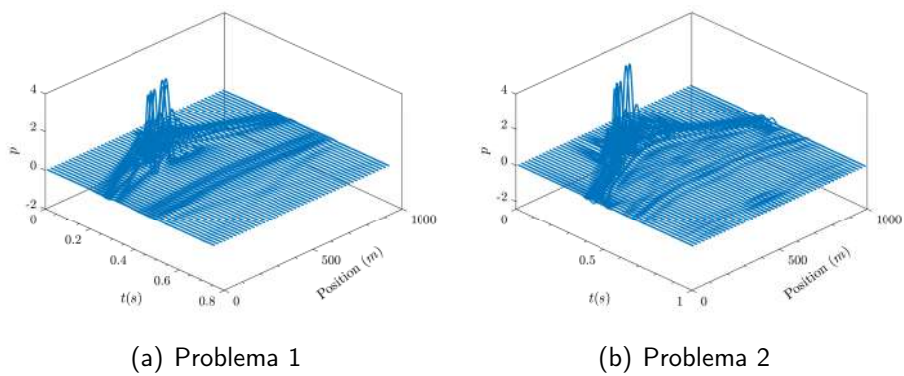


Figura 5: Sismogramas.

Dado que este é o relatório final desse projeto de iniciação científica, alguns pontos merecem destaque. O primeiro é que tivemos êxito na realização das tarefas inicialmente propostas, cumprindo o prazo estabelecido para a execução das mesmas. Ressalta-se também a quantidade considerável de novos conhecimentos adquiridos por mim e pelo meu orientador ao término deste projeto, sendo que a maior parte destes não são abordados nas disciplinas de graduação em Geologia, com isso foi me proporcionado uma apresentação deste trabalho de iniciação científica em um congresso brasileiro no ano passado, diversas jornadas de iniciação científica tanto no LNCC quanto na Universidade na qual eu estudo e não esquecendo o fato de que este projeto do qual participei vai me proporcionar um trabalho de conclusão de curso. Por fim, é importante destacar a qualidade dos resultados obtidos, o qual esperamos disponibilizar, com fins didáticos, para uso da comunidade acadêmica, para que novos trabalhos se desenvolvam a partir do nosso. Agradeço a toda comunidade acadêmica do LNCC, em especial ao meu orientador Márcio Borges por toda a oportunidade e pelo estudo desenvolvido ao longo desses 3 anos.

Laboratório Nacional de Computação Científica Coordenação de Ciência da Computação (CCC)

Relatório de atividades

Programa Institucional de Bolsas de Iniciação Científica PIBIC

Uso do Modelo Roofline para Melhorias no Desempenho e no Consumo de Energia em Sistemas HPC

Vitor Carlos Soares de Sá

Orientador - Prof. Bruno Schulze
Coorientadora - Profa. Mariza Ferro

Período Agosto/2020 a Julho/2021

Agosto 2021

Sumário

1	Introdução	1
2	Objetivos	1
3	Metodologia	1
4	Experimentos e Resultados	3
4.1	Identificando Características de Desempenho	3
4.2	Sugerindo a Arquitetura mais Eficiente	5
4.3	Resultados Preliminares	5
5	Conclusões	6

1 Introdução

A Computação de Alto Desempenho (HPC) vem auxiliando no avanço da ciência possibilitando que simulações mais complexas e com grandes volumes de dados sejam processadas e analisadas de forma cada vez mais precisas e em um tempo viável de execução. Um dos domínios que vem alcançando resultados surpreendentes devido a essa convergência de desempenho computacional e grande volume de dados disponíveis é a Inteligência Artificial (IA), especialmente em sua sub-área de Aprendizado de Máquina (AM), a qual vem sendo cada vez mais utilizada em ambientes de supercomputadores, permitindo que aplicações, antes impossíveis, sejam executadas em tempo viável. Entretanto, como consequência direta do aumento e uso do poder computacional, vem o aumento no consumo de energia elétrica, pois esses ambientes computacionais requerem muita energia.

Portanto, este trabalho de pesquisa se insere na busca por soluções para conter este problema, visando uma computação verde e ecologicamente viável. A proposta é que mediante a compreensão e caracterização do desempenho dessas aplicações científicas, seja possível realizar o uso eficiente dos recursos computacionais e reduzir o consumo de energia elétrica. Por meio da utilização da metodologia adotada neste trabalho, o modelo *Roofline*, foi possível identificar os aspectos que limitam o desempenho das aplicações, sugerir otimizações de *software* e *hardwares* ideais para sua execução. Para cumprir com o objetivo proposto por este trabalho, foram definidas uma série de metas, que foram desenvolvidas por meio de duas fases de experimentos (Seção 4.1 e 4.2). Os resultados foram satisfatórios e permitem a compreensão dos limites das aplicações, dos fatores que possibilitam sua otimização, e podem sugerir *hardwares* que aumentem sua eficiência de execução.

2 Objetivos

O objetivo geral deste projeto é o estudo do comportamento de aplicações científicas em relação à arquitetura de *hardware* visando melhorias no equilíbrio entre o desempenho e a economia de energia. Além disso, viabilizar o uso eficiente dos recursos computacionais e o desenvolvimento de aplicações em busca de uma computação verde. Os objetivos específicos são: 1) Monitorar de forma precisa e eficiente a execução das aplicações; 2) Compreender as aplicações científicas quanto aos requisitos de hardware mais importantes para resolução dos problemas modelados os quais utilizam diferentes métodos matemáticos; 3) Compreender as arquiteturas computacionais quanto a capacidade computacional e a eficiência energética; 4) Identificar os fatores que limitam o desempenho das aplicações, utilizando medidas de Intensidade Aritmética e compreender como estes fatores se relacionam com o consumo de energia.

3 Metodologia

Com base nos objetivos apresentados na Seção 2 foram propostas as seguintes metas para a realização do projeto:

Meta 1: Levantamento bibliográfico sobre o estado da arte.

Foi realizada uma extensa revisão bibliográfica sobre o modelo *Roofline* [Williams et al. 2009, Ilic et al. 2014, Yang et al. 2018, Lo et al. 2014]. Ferramentas de implementação do modelo *Roofline*, como o *Intel Advisor* [Marques et al. 2017] e o *NVIDIA Nsight* [Ding and Williams 2019]. A ferramenta de modelagem de desempenho *Extra-P* [Graf 2017]. O conjunto de ferramentas de infraestrutura de medição *Score-P* [Mey et al. 2012].

Meta 2: Definição das aplicações a serem utilizadas para o estudo.

Na primeira fase deste projeto foram avaliadas aplicações voltadas para HPC (conjunto de benchmarks do NAS-HPC [Frumkin et al. 2009]) e estudos iniciais com aplicações de AM (CNN, SVR e K-means). Nesta fase foram selecionadas novas aplicações voltadas apenas para AM. A seleção dessas aplicações foi feita com base nas mais utilizadas pelos usuários do supercomputador SDumont e também no uso mundial, conforme relatório do Kaggle 2020 [Kaggle 2020]. Foram estudados os algoritmos de árvore de decisão baseada em classificação C4.5 [Quinlan 1993], implementado em linguagem C, Python e por meio da biblioteca scikit-learn [Pedregosa et al. 2011]; e o algoritmo de árvore de classificação e regressão CART [Breiman et al. 1984].

Meta 3: Monitoramento da execução das aplicações, coleta de contadores de *hardware*, desempenho e consumo de energia e identificação dos parâmetros mais relevantes.

Monitorar a execução de aplicações em diferentes arquiteturas pode não ser uma tarefa trivial, pois os contadores de *hardware* podem variar entre os modelos de processadores e as diferentes arquiteturas. Ainda, as ferramentas disponíveis para o monitoramento podem não coletar todos os parâmetros relacionados ao desempenho e ao consumo de energia. Desta forma, foram estudadas um conjunto de ferramentas de monitoramento de desempenho e energia, verificando suas vantagens, limitações e como essas ferramentas apresentam os dados para análise. Foi avaliado ainda, a facilidade de instalação, a sobrecarga de recursos causada pelo monitoramento (*overhead*), a utilização e a interpretação destes resultados. Dentre as ferramentas estudadas e utilizadas durante o projeto, temos: *Perf*¹, por ser uma ferramenta de análise de desempenho que permite a coleta de informações dos registradores de *hardware*. A ferramenta *Likwid* [Eitzinger et al. 2011], utilizando a métrica *perfctr*, que consiste em obter os dados diretamente dos contadores de *hardware*. Os principais parâmetros que foram utilizados para obter informações relevantes para o desenvolvimento da pesquisa utilizando a ferramenta *Likwid* foram: FLOPS_DP (informações de desempenho de ponto flutuante), ENERGY (informações de consumo de energia), MEM (informações de uso de memória RAM, largura de banda), L2 e L3 (informações de uso de memória cache, respectivamente), DATA (oferece as informações necessárias para mensurar o consumo de memória cache em nível L1). Dentre outras ferramentas como *Intel VTUNE*, *nvprof*, *NVIDIA Nsight* e *Intel Advisor*.

Meta 4: Compreensão das aplicações científicas e arquiteturas quanto à exigência dos recursos computacionais e eficiência energética. Identificação dos fatores que limitam o desempenho das aplicações utilizando medidas de intensidade aritmética.

Para a realização desta meta foi utilizado como base o modelo *Roofline* e com ele foi desenvolvido um conjunto de experimentos detalhados na Seção 4. A escolha do modelo *Roofline* [Williams et al. 2009, Ilic et al. 2014, Yang et al. 2018, Lo et al. 2014] foi motivada por ele se mostrar promissor para alcançar os objetivos propostos, oferecendo um modelo visual que relaciona o desempenho do processador ao tráfego de memória, apontando os fatores que limitam o desempenho da aplicação (gargalos computacionais) e possíveis formas de otimização. O seu diferencial é unir todas essas informações em um único ambiente, diferindo dos *benchmarks* tradicionais, os quais analisam separadamente o desempenho da aplicação e a capacidade computacional do *hardware*, gerando informações pertinentes sobre o funcionamento do *hardware* e do *software*.

A implementação de um modelo *Roofline* não é uma tarefa trivial, pois depende da coleta de parâmetros práticos (parâmetros da aplicação) e teóricos (parâmetros do máximo atingível pela arquitetura), que podem variar a metodologia de coleta desses parâmetros de acordo com a arquitetura sendo utilizada. Existem algumas ferramentas que implementam o modelo *Roofline*, sendo uma das principais o *Intel Advisor*. Essa ferramenta além de analisar o desempenho da aplicação em relação ao *hardware*, dá ênfase para análise de eficiência de vetorização e *multithreading*. No gráfico do *Advisor* (Figura 1), cada *loop*/função da aplicação representa um círculo, sendo 3 tipos diferentes de círculos (amarelos, vermelhos e verdes). Isso auxilia na análise do custo computacional de cada círculo,

¹https://perf.wiki.kernel.org/index.php/Main_Page

destacando então os vermelhos como custosos, amarelo como custo moderado e verde como baixo custo. É possível, ainda, avaliar o quão otimizada se encontra a aplicação, relacionando a distância do ponto ao teto de desempenho (quanto mais longe, mais espaço para ganho de desempenho). O teto acima do ponto representa o limitador de desempenho daquele ponto em específico, onde cada teto representa seu próprio limite de *hardware*, caso não seja feita a otimização representada pelo próximo teto. É possível observar que a parte do gráfico onde a tonalidade é mais clara, é considerada intensiva em memória, tonalidade moderada é considerada um equilíbrio entre memória e processamento, já a tonalidade mais escura representa intensivo em processamento. Os pontos são deslocados entre essas áreas de acordo com sua intensidade aritmética, quanto maior, mais intensivo em processamento.

4 Experimentos e Resultados

Os experimentos foram executados em uma arquitetura com processador i7-8700, clock de 3.20 GHz, pico teórico de 307.2 GFlops, 64GB de memória RAM e o pico teórico de largura de banda em 42.56GB. Para a terceira fase de experimentos, além da arquitetura descrita acima, também foi utilizada uma arquitetura semelhante, com exceção do tamanho da memória RAM, que invés de 64GB é 16GB, com o intuito de avaliar mudanças de comportamento para aplicações que possuem gargalos de memória RAM.

4.1 Identificando Características de Desempenho

Esta seção de experimentos tem como objetivo avaliar, por meio da utilização do modelo *Roofline*, diferentes formas de implementação de um algoritmo e analisar as mudanças no desempenho e nos requisitos computacionais da aplicação baseada na implementação de técnicas de otimização ou em uma característica de implementação adotada (mudança na linguagem de implementação, *multithreading*, multiprocessamento, dentre outras). Para esta fase de experimentos foi utilizado o algoritmo de AM: árvore de decisão baseada em classificação, implementado: na linguagem C do algoritmo C4.5, sem nenhuma técnica de otimização; na linguagem Python do algoritmo C4.5, sem nenhuma técnica de otimização; e na linguagem Python do algoritmo CART, utilizando a biblioteca scikit-learn, que utiliza a técnica de otimização de multiprocessamento. Todos os algoritmos desta fase de experimentos foram treinados em uma base de dados sintética de 10.000 exemplos e 21 atributos.

Na Figura 1 é apresentado o resultado para o algoritmo C4.5 implementado em *Python*, e que não é adotada nenhuma técnica de otimização. Por meio da tonalidade de cores de fundo de seus principais pontos, podemos caracterizar esta aplicação como intensiva em memória, e que possui baixíssimo requisito computacional voltada a processamento. Seu gargalo é a memória DRAM, entretanto, com base na distância entre os pontos e o teto de memória DRAM, pode-se notar que este algoritmo está fazendo um uso ineficaz de memória e processamento, o que mostra que há possibilidades para otimizações dentro dessas áreas e possibilitando assim, aumentar significativamente o seu desempenho.



Figura 1: Resultado do algoritmo de Árvore de Decisão C4.5 implementado em *Python*.

Na Figura 2 é apresentado o resultado do algoritmo C4.5 implementado em C, no qual não foi adotada nenhuma técnica de otimização. Com base nos requisitos computacionais de seus principais pontos, pode-se caracterizar esta implementação como equilibrada entre memória e processamento. Entretanto, esta implementação tende muito mais a memória do que processamento, chegando a ficar muito próximo da linha que separa essas duas caracterizações. Seu principal gargalo computacional é a memória DRAM, que assim como a implementação em *Python*, também faz um uso ineficaz de memória e processamento. Porém, este algoritmo possui tanto um desempenho quanto uma intensidade aritmética maior que a implementação em *Python*. Como ambas possuem as mesmas características de implementação, o que se conclui é que este ganho de desempenho é devido à diferença de linguagem de implementação, pois o C prevalece sobre o *Python* em questões de desempenho e tempo de execução.

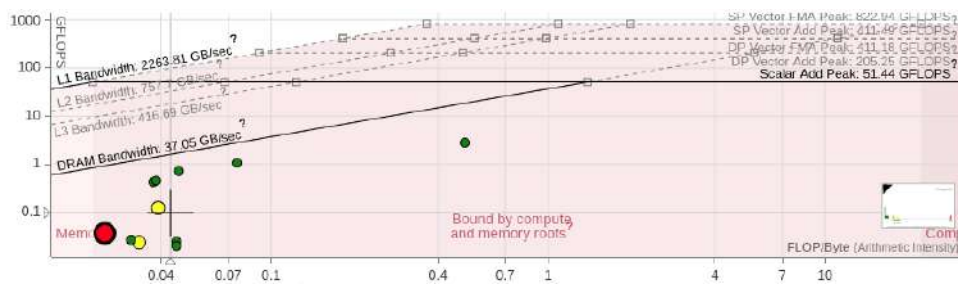


Figura 2: Resultado do algoritmo de Árvore de Decisão C4.5 implementado em C.

Já na Figura 3 é possível observar o resultado da execução do algoritmo CART implementado em *Python* utilizando a biblioteca do scikit-learn, e que possui características de multiprocessamento. Esta implementação é definida como equilibrada entre memória e processamento de acordo com o *Roofline*. Entretanto, este algoritmo tende mais a memória do que para processamento. O principal fator que limita seu desempenho é a memória DRAM, onde possui uma pequena lacuna para otimizações baseadas em DRAM. Neste caso, por meio de um melhor uso de memória cache compartilhada é possível melhorar significativamente o desempenho desta aplicação. Além disso, uma arquitetura com uma alta disponibilidade de memória seria o ideal para a execução do algoritmo em seu estado atual. O fato de possuir multiprocessamento provou ser uma otimização crucial quando se busca otimizar uma aplicação, pois fez com que essa implementação obtivesse um desempenho e uma intensidade aritmética muito maior que as demais implementações.

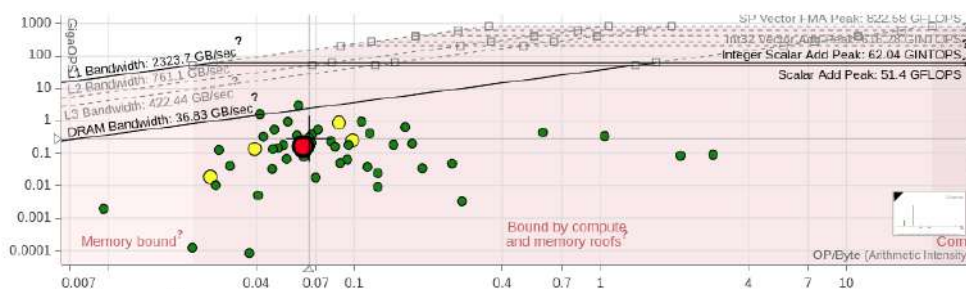


Figura 3: Resultado do algoritmo de Árvore de Decisão CART implementado por meio da biblioteca do *SKLearn* (*Python*).

Com base na avaliação dos resultados obtidos nesta seção, foi possível identificar a diferença do desempenho de diversas formas de implementação de um algoritmo. Também é possível afirmar o quanto uma característica de implementação pode afetar os requisitos computacionais de determinada aplicação. Por exemplo, na Figura 3, que após aplicar uma otimização de multiprocessamento, a aplicação passou a ser mais intensiva em processamento do que a aplicação sem otimização Figura 1. Além disso, também foi possível identificar que a linguagem de implementação adotada contribuiu para a mudança no desempenho da aplicação.

4.2 Sugerindo a Arquitetura mais Eficiente

Esta fase de experimentos consiste em desenvolver uma metodologia, a partir do modelo *Roofline*, para sugerir a arquitetura mais eficiente para a execução de uma aplicação científica com base em seu principal gargalo computacional. Com base nos resultados obtidos no estudo apresentado em [de Sá et al. 2020], é possível identificar um certo padrão entre os algoritmos de AM, caracterizado pelo gargalo em memória DRAM. Com base nesse padrão identificado por meio do modelo *Roofline*, será avaliado o seu uso para sugerir qual é a arquitetura mais eficiente para a execução de uma aplicação científica. Portanto, para esta fase de experimentos foram utilizadas duas arquiteturas distintas (apresentadas na Seção 4), que possuem diferenças significantes de memória DRAM. Os experimentos foram realizadas com o algoritmo de árvore de classificação e regressão (CART) implementados em *Python* e sem adotar nenhuma técnica de otimização, treinado em uma base de dados sintética de 10.000 exemplos e 21 atributos. Para as arquiteturas usadas nos experimentos desta seção a arquitetura que possui mais memória RAM será denominada de Arch1 e a que possui menos memória RAM de Arch2.

Na Figura 4 é possível identificar dois resultados dentro de um mesmo ambiente, sendo representado por um quadrado o resultado da execução do algoritmo na Arch1 e o representado por um círculo o resultado da execução do algoritmo na Arch2. É possível identificar uma diferença nítida de desempenho entre às duas arquiteturas computacionais, fazendo com que o resultado da Arch1 fique bem mais próximo ao teto de memória DRAM que o resultado da Arch2. Este resultado significa que o algoritmo possui menos lacunas de otimização para esta área, provando que o algoritmo aumentou sua eficiência de utilização de memória DRAM, possibilitando um desempenho maior em todos os seus *loops* e funções..

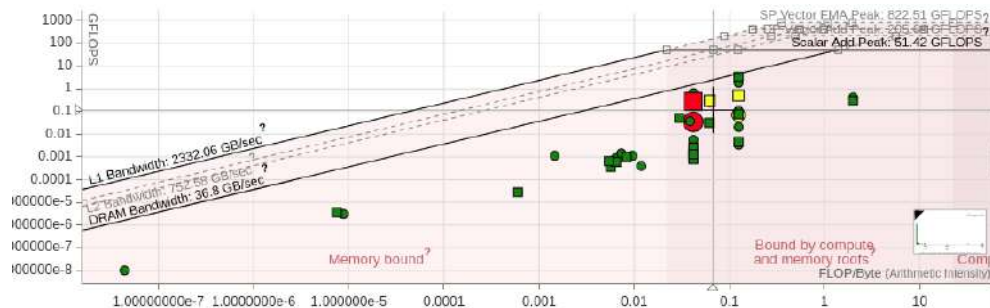


Figura 4: Resultado do algoritmo de Árvore de Classificação e Regressão (CART) implementado em *Python*.

De acordo com os resultados obtidos nesta seção, foi possível observar que por meio do análise do modelo *Roofline*, pode-se identificar o gargalo de uma aplicação científica, e com isso, sugerir uma arquitetura que melhor atenda a sua necessidade, melhorando significativamente seu desempenho.

4.3 Resultados Preliminares

Vitor Sá, Vinícius Klôh, Bruno Schulze & Mariza Ferro. (2020). Análise de Desempenho e de Requisitos Computacionais Utilizando o Modelo Roofline: Um estudo para Aplicações de Inteligência Artificial e do NAS-HPC. XXI Simpósio em Sistemas Computacionais de Alto Desempenho (WSCAD - WIC 2020). Disponível em: https://sol.sbc.org.br/index.php/wscad_estendido/article/view/14085/13933

Vinícius Klôh, Matheus Gritz, Vitor Sá, Bruno Schulze & Mariza Ferro. (2020). *Characterization of the Scientific Applications Requirements: An Hybrid Approach Using Machine Learning and Roofline*. Relatório Técnico. Disponível em: <http://doi.org/10.5281/zenodo.3961639>

Indicado aos Melhores Trabalhos da Jornada de Iniciação Científica e Tecnológica do LNCC/PIBIC 2020.

Indicado ao Prêmio Nacional Destaque na Iniciação Científica e Tecnológica PIBIC 2020 (ainda sob avaliação). <http://www.destaqueict.cnpq.br/web/pdict/inicial>.

Neste momento encontra-se em preparação um artigo para submissão na revista eletrônica de iniciação científica em computação (<https://sol.sbc.org.br/journals/index.php/reic/about>) e um artigo que será submetido ao WSCAD-WIC 2021 no dia 16/08.

5 Conclusões

Neste trabalho foram apresentados os resultados da pesquisa realizada com o objetivo de estudar o comportamento de aplicações científicas, em especial de AM, em arquiteturas de *hardware* utilizadas em ambientes HPC, visando identificar os fatores que limitam o seu desempenho. Para alcançar esses objetivos foram definidos um conjunto de metas, as quais foram apresentadas neste relatório, bem como a metodologia adotada para o seu desenvolvimento e os principais resultados alcançados. As principais contribuições a serem destacadas como resultados desse trabalho são: os resultados da Seção 4.1 contribuem para a avaliação do impacto que determinados tipos de otimização têm no desempenho e nas mudanças de características da aplicação. Além disso, foi demonstrado como a linguagem de implementação altera o desempenho de uma aplicação. Com os resultados da Seção 4.2 foi possível observar que pela identificação dos principais limites de desempenho apontados pelo modelo *Roofline* podemos desenvolver uma metodologia para sugerir as arquiteturas ideais para executar de maneira eficiente uma determinada aplicação científica.

Em trabalhos futuros, será feita uma análise sobre os padrões de consumo de energia do algoritmo, buscando utilizar a metodologia *Roofline* para isso e também desenvolver um *Roofline* para consumo de energia. Além disso, utilizando os parâmetros coletados pelo *Roofline* e as outras ferramentas como o *Likwid* e *Perf*, desenvolver um modelo teórico que permita estimar o desempenho e o consumo de energia de uma aplicação de AM. O objetivo desta atividade é, a partir das coletas dos parâmetros, investigar qual a influência do crescimento do número de exemplos e do número de atributos sobre o consumo de energia dos algoritmos e seu desempenho computacional. Também, dos tipos e do número de cada tipo de atributo. A compreensão da influência desses parâmetros sobre o tempo de execução e o consumo de energia são fundamentais para desenvolver um modelo teórico para estimar o consumo de energia em AM. Com base nesses resultados, usando os parâmetros de *hardware* coletados e essas características relacionadas aos conjuntos de dados espera-se obter um modelo teórico que permita estimar o consumo de energia dos algoritmos.

Referências

- [Breiman et al. 1984] Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984). *Classification and regression trees*. CRC press. 2
- [de Sá et al. 2020] de Sá, V., Klôh, V., Schulze, B., and Ferro, M. (2020). Análise de desempenho e de requisitos computacionais utilizando o modelo *roofline*: Um estudo para aplicações de inteligência artificial e do nas-hpc. In *WSCAD 2020 - WIC* (). 5
- [Ding and Williams 2019] Ding, N. and Williams, S. (2019). An instruction *roofline* model for gpus. In *2019 IEEE/ACM Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS)*, pages 7–18. 1

- [Eitzinger et al. 2011] Eitzinger, J., Hager, G., Wellein, G., and Meier, M. (2011). Poster: Likwid: lightweight performance tools. pages 29–30. 2
- [Frumkin et al. 2009] Frumkin, M., Jin, H., and Yan, J. (2009). Implementation of nas parallel benchmarks in high performance fortran. 2
- [Graf 2017] Graf, A. (2017). Modeling cache locality with extra-p. 1
- [Ilic et al. 2014] Ilic, A., Pratas, F., and Sousa, L. (2014). Cache-aware roofline model: Upgrading the loft. *IEEE Computer Architecture Letters*, 13(1):21–24. 1, 2
- [Kaggle 2020] Kaggle (2020). State of data science and machine learning 2020. Technical report. 2
- [Lo et al. 2014] Lo, Y., Williams, S., Straalen, B., Ligocki, T., Cordery, M., Wright, N., Hall, M., and Oliker, L. (2014). Roofline model toolkit: A practical tool for architectural and program analysis. pages 129–148. 1, 2
- [Marques et al. 2017] Marques, D., Duarte, H., Ilic, A., Sousa, L., Belenov, R., Thierry, P., and Matveev, Z. A. (2017). Performance analysis with cache-aware roofline model in intel advisor. In *2017 International Conference on High Performance Computing Simulation (HPCS)*, pages 898–907. 1
- [Mey et al. 2012] Mey, D., Biersdorf, S., Bischof, C., Diethelm, K., Eschweiler, D., Gerndt, M., Knüpfer, A., Lorenz, D., Malony, A., Nagel, W., Oleynik, Y., Feld, C., Saviankou, P., Schmidl, D., Shende, S., Wagner, M., Wesarg, B., and Wolf, F. (2012). *Score-P: A Unified Performance Measurement System for Petascale Applications*, pages 85–97. 1
- [Pedregosa et al. 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830. 2
- [Quinlan 1993] Quinlan, J. R. (1993). *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. 2
- [Williams et al. 2009] Williams, S., Waterman, A., and Patterson, D. (2009). Roofline: an insightful visual performance model for multicore architectures. *Commun. ACM*, 52(4):65–76. 1, 2
- [Yang et al. 2018] Yang, C., Gayatri, R., Kurth, T., Basu, P., Ronaghi, Z., Adetokunbo, A., Friesen, B., Cook, B., Doerfler, D., Oliker, L., Deslippe, J., and Williams, S. (2018). An empirical roofline methodology for quantitatively assessing performance portability. pages 14–23. 1, 2

Laboratório Nacional de Computação Científica
Coordenação de Ciência da Computação (CCC)

Relatório de atividades

Programa Institucional de Bolsas de Iniciação Científica PIBIC

Análise de Desempenho e Características de Paralelização dos
Algoritmos de Aprendizado de Máquina

Vitor Gonçalves Vieira

Orientador - Prof. Bruno Schulze
Coorientadora - Profa. Mariza Ferro

Período Setembro/2020 a Agosto/2021

Agosto 2021

Sumário

1	Introdução	1
2	Metodologia	1
3	Resultados e Discussão	3
3.1	Trabalhos Publicados	5
4	Conclusões e trabalhos Futuros	6

Dados Referentes à Bolsa

- **Bolsista:** Vitor Gonçalves Vieira
- **Coordenador/Orientador:** Bruno Schulze (CCC-LNCC)
- **Coorientadora:** Mariza Ferro (LNCC)
- **Período de Vigência da Bolsa:** Setembro de 2020 até Agosto de 2021
- **Projetos Relacionados:** *Green AI: Towards an ecologically viable machine learning* - CLIMAT-AMSUD (código: 21-CLIMAT-07), *Towards a Sustainable Artificial Intelligence - SuSAIn* times associados INRIA

1 Introdução

O Aprendizado de Máquina (AM)¹ é uma importante sub-área da Inteligência Artificial (IA) que pode ser definida como a arte de programar um computador, para que ele possa aprender com os dados [Gron 2017]. O AM teve enormes progressos na última década, com inúmeras aplicações práticas inseridas em atividades do nosso dia a dia. A necessidade de modelos de IA e AM na solução de problemas complexos é evidente, porém, uma questão tem sido negligenciada, o consumo de energia desses algoritmos que muitas vezes utilizam alta capacidade computacional para o seu treinamento. Prevê-se que até 2040 metade do consumo mundial de energia elétrica será atribuído a instalações de computação [Villani et al. 2018], e que atualmente *data centers* já são responsáveis por 3% do consumo global de energia ².

Mesmo com a relevância dos algoritmos de AM, pouco se sabe a respeito dos seus requisitos computacionais e consumo energético em diferentes arquiteturas computacionais. A maioria das pesquisas nesta área se concentram em melhorar a precisão preditiva dos algoritmos (*accuracy*), independente do custo econômico e ambiental atrelado a isso - *Red AI* [Schwartz et al. 2020]. A eficiência do algoritmo, ou seja, seus requisitos computacionais, seu desempenho e consumo de energia somente mais recentemente vem sendo considerados em busca de uma IA e AM ecologicamente viável, mais sustentável e inclusivo - *Green AI* [Schwartz et al. 2020]. Por exemplo, os requisitos de um algoritmo de Rede Neural Artificial (RNA) podem ser muito diferentes dos requisitos de um algoritmo de Árvores de Decisão (DT³ ou AD). Ou seja, as diferentes tarefas de AM, envolvem diferentes algoritmos, os quais podem ter requisitos computacionais e consumo energético bastante diversos. Além disso, com a crescente necessidade pela computação de alto desempenho para o treinamento dos modelos de AM, é necessário compreender se esses algoritmos possuem um desempenho satisfatório nessas arquiteturas e se podem ser otimizados para uma melhor eficiência energética.

Assim, o objetivo geral desse projeto de pesquisa é a identificação dos requisitos de software e hardware de diferentes aplicações de IA, mais especificamente aplicações de AM. Além disso, será realizada uma avaliação sobre a influência dos principais hiperparâmetros e dos conjuntos de dados no desempenho e na energia, em busca de um AM mais ecológico e sustentável. Para alcançar este objetivo geral, alguns objetivos específicos foram definidos: 1) Estudar diferentes classes de algoritmos de AM e seus respectivos algoritmos; 2) Monitorar de forma precisa e eficiente a execução dos algoritmos de AM durante as fases de treinamento. Serão avaliados aspectos de desempenho computacional, consumo de energia e eficiência preditiva (p. ex. *accuracy*); 3) Identificar os principais requisitos de hardware para execução das diferentes técnicas de AM (uso de CPU, E/S, memória e comunicação); 4) Identificar os *hotspots* de algoritmos de AM e como essas partes do código influenciam no desempenho e consumo de energia; 5) Investigar a influência dos hiperparâmetros dos algoritmos e dos tipos conjuntos de dados no consumo de energia de algoritmos de AM.

2 Metodologia

Para cumprir os objetivos acima descritos foi adotada uma metodologia para o desenvolvimento da pesquisa baseada em metas.

¹Do termo em inglês Machine Learning.

²independent.co.uk/climate-change/news/global-warming-data-centres-to-consume-three-times-as-much-energy-in-next-decade-experts-warn-a6830086.html

³Do termo em inglês, Decision Trees.

Meta 1: Levantamento bibliográfico sobre AM.

Revisão bibliográfica sobre AM [Mitchell 1997], algoritmos de DT C4.5 [Quinlan 1986] e CART [Breiman et al. 1984] e Florestas Randômica (RF⁴, ou FR) [Breiman 2001], ambos para tarefas de classificação e regressão. Esses algoritmos foram selecionados devido a sua simplicidade e por serem os algoritmos mais utilizados pela comunidade de ciência de dados [Kaggle 2020].

Meta 2: Caracterizar diferentes modelos de algoritmos de AM, aferir seu consumo energético e emissão de CO₂e.

Essa meta consiste em: i) identificar as principais funções e subfunções dos algoritmos; ii) realizar o treinamento dos algoritmos de AM com diferentes conjuntos experimentais em duas arquiteturas distintas; iii) medir o desempenho e o consumo de energia de cada parte do código, identificando as funções que tem maior impacto sobre o desempenho e consumo de energia dos algoritmos (o que denominamos de hotspots);

Para coletar o consumo energético foi utilizada a ferramenta pyRAPL. Com esta ferramenta faz-se necessário a inserção de diretivas de instrumentação no código para que seja possível a coleta do consumo de energia dos trechos de interesse. Sabe-se que este tipo de abordagem normalmente ocasiona um *overhead* (sobrecarga que a ferramenta causa durante o monitoramento do código) significativo no resultado. Portanto, para avaliar o *overhead* da pyRAPL foram realizados experimentos analisando o tempo de execução do algoritmo de DT para a tarefa de classificação com e sem a utilização da ferramenta pyRAPL. Os totais com o overhead, variam de 0,28% a aproximadamente 15%, de acordo com o tamanho do conjunto de dados e a quantidade de atributos utilizados para o treinamento do algoritmo.

Para o cálculo da emissão de CO₂e foram utilizadas as Equações 1 e 2, baseadas no trabalho de [Strubell et al. 2019]. Na Equação 1 é calculada a Energia Total (ET) em KWh, onde *PUE* (*Power Usage Effectiveness* [Avelar et al. 2012]) representa a eficiência energética, *t*, o tempo em horas, *P_CPU*, *P_GPU* e *P_MEM*, respectivamente, potência em Watts da CPU, GPU e memória RAM. Essas métricas foram coletadas com a ferramenta perf⁵. Na Equação 2 é calculada a emissão de CO₂e, os parâmetros utilizados são: a constante de CO₂e do país, a qual de acordo com [Miranda 2012] é de 0,125 Kg/KWh para o Brasil, e a ET.

$$ET = \frac{PUE \times t \times (P_CPU + P_GPU + P_MEM)}{1000} \quad (1)$$

$$CO_2e = 0,125 \times ET \quad (2)$$

Meta 3: Estudo da influência dos hiperparâmetros dos algoritmos no desempenho e consumo energético e dos conjuntos de dados

Complementando a meta 2, foi definido um estudo do comportamento dos algoritmos de DT e RF, alterando seus hiperparâmetros, e como essa alteração influencia no consumo de energia e o impacto direto sobre sua precisão preditiva. Para essa meta foi definido o conjunto experimental apresentado na Figura 1 onde são apresentadas as variações de hiperparâmetros para os algoritmos de DT classificação e regressão gerando 16 configurações experimentais diferentes. Para a árvore de decisão classificação e regressão, os hiperparâmetros escolhidos para serem variados são - Figura 1 (a) e (b): *max_depth* and *min_samples_split*. O primeiro define a profundidade máxima da árvore. Se não modificado (padrão), os nós são expandidos até que todas as folhas sejam puras ou até mesmo que todas as folhas contenham menos que *min_sample_split* que por padrão é 2 (número mínimo de amostras necessárias para um nó interno).

Para a RF foram realizadas 27 configurações experimentais (Figura 2) para as tarefas de classificação e regressão. Os hiperparâmetros ajustados são: *n_jobs* (padrão é nenhum = 1), que é o número máximo de trabalhos em execução em paralelo; *n_estimators* (padrão o valor = 100), que é o número de árvores na floresta; e *max_depth* (nenhum como padrão), que é o valor que define os nós expandidos, até que todas as folhas sejam puras ou até que todas as folhas contenham menos do que *min_samples_split*. Vale ressaltar que o valor padrão do parâmetro *max_features* (que representa o número de atributos a serem considerados ao procurar a melhor divisão), é diferente para as duas tarefas (classificação e regressão). Para a classificação, o padrão é *max_features* = *sqrt* (número de features) e, para a regressão, o padrão é *max_features* = número de features. Apesar de *max_features* não estar entre os parâmetros variados, foi necessário configurá-lo na RF regressora, como o valor padrão do RF classificador, ou seja, *max_feature* = *sqrt*, o que vai permitir uma comparação justa entre as duas tarefas. Todos os outros parâmetros são padrão.

⁴Do termo em inglês, Random Forest

⁵Os detalhes de utilização da ferramenta perf estão disponíveis em [Klöh et al. 2019]

Config	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Config	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16				
max_depth	8				16				33				66				max_depth	9				18				37				74							
min_sample_split	2	4	8	16	2	4	8	16	2	4	8	16	2	4	8	16	min_sample_split	2	4	8	16	2	4	8	16	2	4	8	16	2	4	8	16	2	4	8	16

(a) DT classificação

(b) DT regressão

Figura 1: Configurações hiperparâmetros DT classificação e regressão.

Config	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27			
max_depth	8			30			60			8			30			60			8			30			60					
min_sample_split	50	100	150	50	100	150	50	100	150	50	100	150	50	100	150	50	100	150	50	100	150	50	100	150	50	100	150	50	100	150
n_jobs	1									-7									-1											

(a) RF classificação

Config	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27			
max_depth	9			36			60			9			36			60			9			36			60					
min_sample_split	50	100	150	50	100	150	50	100	150	50	100	150	50	100	150	50	100	150	50	100	150	50	100	150	50	100	150	50	100	150
n_jobs	1									-7									-1											

(b) RF regressão

Figura 2: Configurações hiperparâmetros RF classificação e regressão.

3 Resultados e Discussão

Para o desenvolvimento dos experimentos foram utilizados diferentes bases de dados. O conjunto experimental 1 (Figura 3a) tem 14 variações nas bases de dados sintéticas com diferentes quantidades de exemplos e atributos e um conjunto apenas com atributos categóricos ou numéricos. No conjunto experimental 2 (Figura 3b) foram utilizados três bases de dados sintéticas, também com variações no número de exemplos e na quantidade e tipo de atributos. Estes conjuntos sintéticos foram gerados com a ferramenta MOA [Bifet et al. 2010]. O conjunto experimental 3 (Figura 3c) são bases de dados reais, sendo a de classificação Higgs dataset⁶ e regressão a Seoul bike trip duration⁷, ambas com 5 milhões de exemplos.

Base de dados sintética		
Nome do Conjunto	Quantidade de Exemplos	Quantidade atributos
Conjunto 1	25000	5
Conjunto 2	25000	10
Conjunto 3	25000	21
Conjunto 4	100000	5
Conjunto 5	100000	10
Conjunto 6	100000	21
Conjunto 7	500000	5
Conjunto 8	500000	10
Conjunto 9	500000	21
Conjunto 10	1000000	5
Conjunto 11	1000000	10
Conjunto 12	1000000	21
Conjunto 13	1000000	21 atributos categóricos
Conjunto 14	1000000	21 atributos numéricos

(a) Conjunto experimental 1

Conjunto	Exemplos	Atributos		
		Total	Númerico	Nominal
1	5 mil	5	4	1
		10	9	1
		21	20	1
2	10 mil	5	4	1
		10	9	1
		21	20	1
3	25 mil	5	4	1
		10	9	1
		21	20	1

(b) Conjunto experimental 2

Tarefa	Exemplos	Atributos
Classificação	5 milhões	28
Regressão		24

(c) Conjunto experimental 3

Figura 3: Descrição dos três diferentes conjunto de dados experimentais.

Todos os experimentos foram implementados com a ferramenta Scikit-Learn [Pedregosa et al. 2011], usando o algoritmo baseado no CART [Breiman 2001] para DT, implementados na linguagem de programação Python⁸ e com o Pandas [Wes McKinney 2010] para a leitura dos dados.

Para a caracterização dos diferentes modelos de AM (segundo a metodologia descrita na meta 2) foram

⁶<https://archive.ics.uci.edu/ml/datasets/HIGGS>

⁷<https://www.kaggle.com/saurabhshahane/seoul-bike-trip-duration-prediction>

⁸www.python.org/

identificados as principais funções dos algoritmos DT e RF, buscando identificar aquelas que mais impactam no consumo de energia dos algoritmos. Na Figura 4 é apresentado o resultado dos hotspots para a DT com o algoritmo CART e utilizando o conjunto experimental 2⁹. Foram feitas várias execuções do algoritmo CART sendo monitorada cada função do algoritmo através do software pyRAPL. Após analisar o consumo de energia e tempo de cada função, foram identificadas as funções `build_tree`, a qual é responsável por construir a árvore e o `determine_best_split`, o qual é responsável por avaliar qual é o atributo do conjunto de dados que será usado como nó raiz da DT. As funções responsáveis pela escolha dos atributos mais significativos (nós de decisão) faz com que seja necessário testar todos os exemplos e todos os atributos para a construção da DT.

Funções	Energia (Joules)								
	5 atributos			10 atributos			21 atributos		
	Conj 1	Conj 2	Conj 3	Conj 1	Conj 2	Conj 3	Conj 1	Conj 2	Conj 3
Algoritmo de regressão (CART)									
<code>build_tree</code>	27,7	66,1	157,5	67,9	146,4	382,5	247,9	613,0	1512,7
<code>determine_best_split</code>	31	73,7	178,8	77,8	170,5	447,9	295,2	729,9	1863,2

Figura 4: Hotspots DT - CART, baseadas na execução conjunto experimental 2 (Figura 3b).

Com base na compreensão das partes do algoritmo que mais consomem energia, identificadas nos experimentos com hotspots, outro estudo comparou uma DT com o algoritmo de RF usando o conjunto experimental 1 apresentado na Figura 3a. O objetivo é investigar o impacto da variação do número de exemplos e atributos sobre o consumo de energia dos algoritmos DT e do ensemble¹⁰ - RF para as tarefas de classificação e regressão. Os resultados com os 14 conjuntos (Conjunto experimental 1 - Figura 3a) são apresentados na Figura 5. É possível observar que a RF tem um consumo muito maior se comparado a uma DT, por ser um modelo que agrupa vários resultados de árvores. Além disso, um resultado interessante é a influência do aumento do número de atributos no consumo de energia, mesmo o conjunto 6 tendo 5 vezes menos exemplos que o conjunto 7, este consumiu praticamente o mesmo valor de energia, pois tinha 21 atributos, diferente dos 5 atributos apresentados no conjunto 6. Ainda, há uma diferença no consumo de energia para as tarefas de classificação e regressão. Esses resultados se explicam pelo hotspot de energia na função `determine-best-split`, responsável por encontrar o melhor atributo para particionar os nós. Para a definição do nó na tarefa de regressão é realizado o cálculo do Mean Square Error (MSE), enquanto para a classificação o cálculo é o Gini que teve influência diferente no número de operações realizadas pelo algoritmo.

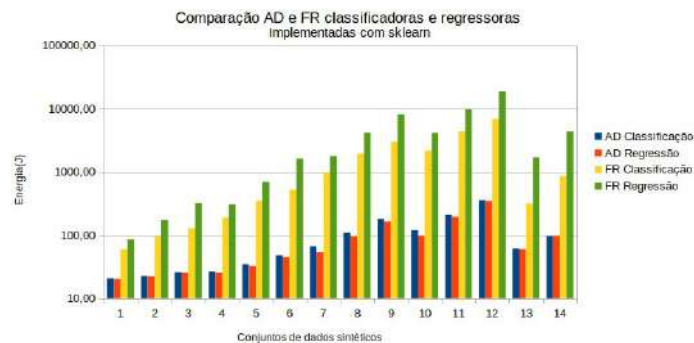


Figura 5: Resultado das execuções DT e RF para as tarefas de classificação e regressão, utilizando os conjunto de dados do conjunto experimental 1 (Figura 3a).

Selecionando a configuração que mais consumiu energia (RF - conjunto 12 - Figura 3a), foi realizado o cálculo de CO_2e (Equação 2). Verificou-se que o consumo gerado por essa RF foi de 0,10736 Kg/ CO_2e para uma única execução de teste. Comparando-se com a emissão de CO_2 de um carro popular motor 1.0 - 8 V, gasolina, isso equivale a emissão de andar 1km com esse carro. Embora esse valor possa parecer baixo, é importante observar que geralmente são realizadas diversas execuções de teste até o ajuste de hiperparâmetros.

Para a meta 3 foi avaliada a influência individual de cada hiperparâmetro no consumo de energia dos algoritmos e no desempenho preditivo (Accuracy e MSE). O objetivo é verificar se é possível manter a precisão e reduzir o consumo de energia. Os resultados para DT classificação e regressão são apresentados nas Figuras 6 e 7, respectivamente. É possível observar que o parâmetro `max_depth` possui maior impacto na precisão e no consumo de energia, para ambas as tarefas de aprendizagem. Para o DT classificadora, com a variação do parâmetro `min_samples_split`, para configurações que têm os mesmos valores que `max_depth` (como 1, 2, 3 e 4; 5, 6, 7 e 8; 9, 10, 11 e 12; e 13, 14, 15 e 16), a precisão não muda e o tempo de execução e energia também não

⁹Por limitações de espaço apenas este resultado está sendo apresentado.

¹⁰São a combinação de diferentes modelos de AM para se obter um único resultado, como as RF.

tem variação significativa. Com a variação do parâmetro `max_depth`, para configurações que têm os mesmos valores de `min_samples_split` (como configurações 1, 5, 9 e 13; 2, 6, 10 e 14; 3, 7, 11 e 15; e 4, 8, 12 e 16), o aumento da precisão é de até 7,5%. No entanto, conforme `max_depth` aumenta de 8 para 66, a energia também aumenta, chegando a 64%.

Para a regressão DT, a variação em `min_samples_split` mostra uma melhoria no MSE variando de 0% (configuração 1-4) a 17% (configuração 13-16) acompanhada por uma redução no consumo de energia de 0% para 7%. É possível notar a influência positiva no MSE, de até 51%, da configuração 1-4 para a configuração 13-16 ajustando o `max_depth`. Porém, para esta melhoria, o aumento do consumo de energia é de até 162%.

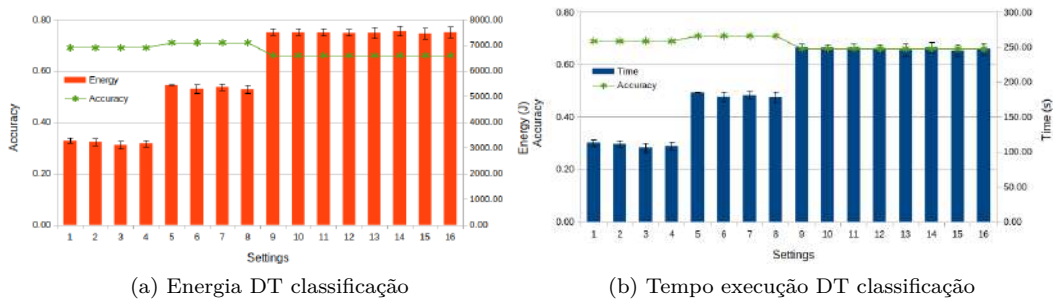


Figura 6: Relação entre a performance preditiva (Accuracy), energia (Energy - Joules) e tempo de execução (time - segundos) da DT classificadora executada com diferentes ajustes de hiperparâmetros (Figura 1a) e utilizando o conjunto experimental 3.

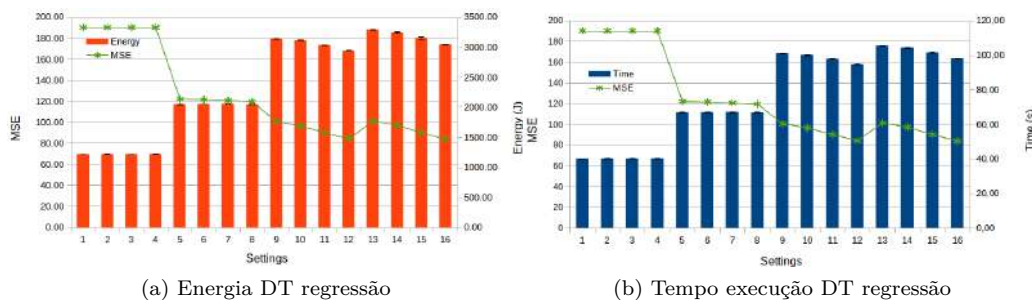


Figura 7: Relação entre a performance preditiva (MSE), energia (Energy - Joules) e tempo de execução (time - segundos) da DT regressão executada com diferentes ajustes de hiperparâmetros (Figura 1b) e utilizando o conjunto experimental 3.

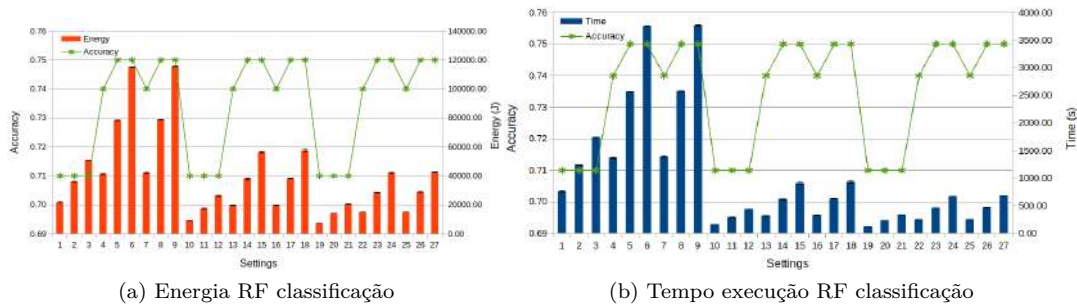
Os resultados para RF são mostrados nas Figuras 8 e 9. `Max_depth` e `n_estimators` são os principais responsáveis pelo aumento do consumo de energia, em proporções muito maiores do que o ganho obtido no desempenho preditivo. O consumo de energia é quase 3 vezes maior quando `n_estimators` e `max_depth` aumenta. É possível notar o crescimento do consumo de energia a cada três configurações (1,2,3 - 4,5,6 e assim por diante), conforme aumenta `n_estimator`. Além disso, para ambas as tarefas, o crescimento em energia é mitigado apenas pela variação em `n_jobs`. É possível ver o impacto positivo ao longo das execuções quando há um maior número de processos executando paralelamente $n_jobs = -1$ ¹¹.

3.1 Trabalhos Publicados

Ao longo deste período foram realizadas as seguintes publicações:

- [Bernardo et al. 2020a] De Paula, F.B., Ferro, M., Vieira, V., Silva, G. D., and Schulze, B. (2020). Em Busca de uma Inteligência Artificial Ecologicamente Viável: Um estudo de caso do Consumo Energético de Algoritmos de Árvore de Decisão. XXI Simpósio em Sistemas Computacionais de Alto Desempenho (WSCAD) 2020.

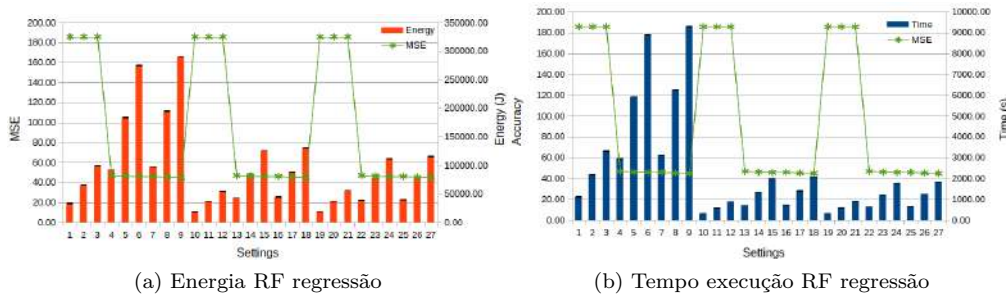
¹¹Para `n_jobs` Default = None, apenas 1 núcleo é usado, já -7, metade dos processadores e -1, todos os processadores.



(a) Energia RF classificação

(b) Tempo execução RF classificação

Figura 8: Relação entre a performance preditiva (Accuracy), energia (Energy - Joules) e tempo de execução (time - segundos) da RF classificadora executada com diferentes ajustes de hiperparâmetros (Figura 2a) e utilizando o conjunto experimental 3.



(a) Energia RF regressão

(b) Tempo execução RF regressão

Figura 9: Relação entre a performance preditiva (MSE), energia (Energy - Joules) e tempo de execução (time - segundos) da RF regressão executada com diferentes ajustes de hiperparâmetros (Figura 2b) e utilizando o conjunto experimental 3.

- [Bernardo et al. 2020b] De Paula, Silva, G., Ferro, M., Vieira, V. and Schulze, B. (2020), Technical report, LNCC. Análise de algoritmos de árvores de decisão e floresta randômica.
- Ferro, M., Silva, G. D., De Paula, F.B., Vieira, V., and Schulze, B. (2021). Towards a Sustainable Artificial Intelligence: A Case Study of Energy Efficiency in Decision Tree Algorithms. Special Issue in Concurrency and Computation Practice and Experience (submetido em Maio 2021).

4 Conclusões e trabalhos Futuros

A identificação dos hotspots permitiu identificar quais partes dos código, e os parâmetros atrelados, tem maior impacto no consumo de energia, sendo um indicativo de quais hiperparâmetros devem ser melhor configurados. Além disso, foi possível observar que não só o aumento do número de exemplos tem impacto no consumo de energia, mas também a quantidade de atributos e o seu tipo. Analisando o ajuste dos hiperparâmetros foi possível perceber que em alguns casos é possível manter o desempenho (ou perdas pequenas), reduzindo significativamente o consumo de energia.

Como trabalhos futuros, será realizada a execução de experimentos com *low-precision* (computação aproximada), para estudar como a variação da precisão impacta o consumo de energia e precisão preditiva durante as fases de treinamento e inferência dos algoritmos de AM. Serão transformados os conjuntos de dados da base de pontos flutuantes que ele estiver (por padrão grande maioria se encontra em 64 bits) para os tamanhos 32 bits e 16 bits. Além disso, partir do conhecimento e resultados obtidos nesses experimentos iniciais, propor uma metodologia sistemática para avaliação de desempenho e consumo de energia de algoritmos de AM, apontando os caminhos para o treinamento e o uso de AM de maneira mais ecológica e sustentável. Por exemplo, que tipo de arquitetura usar para cada algoritmo e tarefa a ser executada, como preparar o conjunto de dados, como definir os hiperparâmetros, quando e se usar menor precisão.

Referências

- [Avelar et al. 2012] Avelar, V., Azevedo, D., French, A., and Power, E. N. (2012). Pue: a comprehensive examination of the metric. 2
- [Bernardo et al. 2020a] Bernardo, F., Ferro, M., Vieira, V., Silva, G., and Schulze, B. (2020a). Em busca de uma inteligência artificial ecologicamente viável: Um estudo de caso do consumo energético de algoritmos de árvore de decisão. In *Anais do XXI Simpósio em Sistemas Computacionais de Alto Desempenho*, pages 179–190. SBC. 5
- [Bernardo et al. 2020b] Bernardo, F., vieira, V., Silva, G., Ferro, M., and Schulze, B. (2020b). Análise de algoritmos de árvores de decisão e floresta randômica. Technical report, Laboratório Nacional de Computação Científica. 6
- [Bifet et al. 2010] Bifet, A., Holmes, G., Kirkby, R., and Pfahringer, B. (2010). Moa: Massive online analysis <http://sourceforge.net/projects/moa-datastream>. *Journal of Machine Learning Research (JMLR)*. 3
- [Breiman 2001] Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32. 2, 3
- [Breiman et al. 1984] Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984). *Classification and regression trees*. CRC press. 2
- [Gron 2017] Gron, A. (2017). *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O’Reilly Media, Inc., 1st edition. 1
- [Kaggle 2020] Kaggle (2020). State of data science and machine learning 2020. Technical report. 2
- [Klôh et al. 2019] Klôh, V., Gritz, M., Schulze, B., and Ferro, M. (2019). Towards an autonomous framework for hpc optimization: Using machine learning for energy and performance modeling. In *Anais do XX Simpósio em Sistemas Computacionais de Alto Desempenho*, pages 438–445, Porto Alegre, RS, Brasil. SBC. 2
- [Miranda 2012] Miranda, M. M. d. (2012). *Fator de emissão de gases de efeito estufa da geração de energia elétrica no Brasil: implicações da aplicação da Avaliação do Ciclo de Vida*. PhD thesis, Universidade de São Paulo. 2
- [Mitchell 1997] Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill. 2
- [Pedregosa et al. 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830. (Decision Tree module scikit-learn 1.10.6). 3
- [Quinlan 1986] Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1(1):81–106. 2
- [Schwartz et al. 2020] Schwartz, R., Dodge, J., Smith, N. A., and Etzioni, O. (2020). Green ai. *Commun. ACM*, 63(12):54–63. 1
- [Strubell et al. 2019] Strubell, E., Ganesh, A., and McCallum, A. (2019). Energy and policy considerations for deep learning in nlp. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650. 2
- [Villani et al. 2018] Villani, C., Bonnet, Y., Rondepierre, B., et al. (2018). *For a meaningful artificial intelligence: towards a French and European strategy*. Conseil national du numérique. 1
- [Wes McKinney 2010] Wes McKinney (2010). Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56 – 61. 3

RELATÓRIO DE TRABALHO DO BOLSISTA

Método dos Volumes Finitos aplicados para a solução de problemas de mudança de fase

Ophir Valiante Duarte Neto (bolsista)

Elson Magalhães Toledo (orientador)

Maicon William Niebus Odone (coorientador)

PIBIC – 01/08/2020 a 01/08/2021

Objetivos

- Prosseguimento dos estudos de diferentes modelos e métodos numéricos relacionados aos processos de mudança de fase com vistas a simulação computacional de problemas de cunho real.
- Exploração e estudo do código do OpenFOAM.
- Estudo dos fundamentos da Mecânica do Contínuo e suas aplicações em modelos computacionais de problemas de engenharia.
- Estudo do escoamento de fluidos viscoelásticos e do módulo viscoelasticFluidFoam, com finalidade de entender a natureza termo-elasto-plástica e viscoelástica de um material em processo de fundição e o surgimento de tensões residuais durante sua solidificação.
- Desenvolvimento de novas funcionalidades (como o cálculo de tensões residuais) para o módulo do OpenFOAM que trata problemas de mudança de fase.

Introdução

Os processos de mudança de fase líquido-sólido e sólido-líquido possuem aplicações importantes na engenharia. Os exemplos mais clássicos são os diversos tipos de processos de fundição, onde peças são fabricadas através da introdução de material líquido em um molde, que então solidifica-se no formato desejado. Outro exemplo mais moderno são os chamados *phase-change materials* (PCM), que podem ser utilizados no lugar de isolantes térmicos em locais com grande variação de temperatura (como em satélites ou em construções civis).

Estes processos normalmente envolvem fenômenos físicos relativamente complexos como, por exemplo: escoamento da fase líquida por convecção natural, surgimento de tensões térmicas residuais na parte sólida, formação de porosidade no interior do material solidificado, descolamento do material sólido das paredes do molde, fase líquida com comportamento viscoso e elástico, etc. Portanto, sua modelagem matemática é notavelmente complexa e os poucos modelos computacionais disponíveis para o estudo de problemas desta natureza são, geralmente, limitados. Por conta disto, o estudo, desenvolvimento e otimização de tecnologias baseadas na mudança de fase líquido-sólido e sólido-líquido (como processos de fundição) depende bastante de experimentos e de modelos empíricos. Isto cria dificuldades principalmente na otimização destas tecnologias, das quais dependem aplicações críticas como a fabricação de materiais energéticos, como combustíveis sólidos de foguetes, granadas de tanques ou ainda de peças metálicas com aplicações de engenharia. A falta de qualidade nestes processos implica em falhas catastróficas.

A pesquisa aqui apresentada dedicou-se ao estudo de modelos físicos e numéricos para a simulação de problemas de mudança de fase. Na busca de melhor compreender melhor a física destes problemas, bem como o desenvolvimento de novos modelos que possam contemplar mais precisamente os fenômenos que neles ocorrem, foram realizados estudos em diversos tópicos da mecânica do contínuo, contemplando desde os fundamentos à casos mais complexos.

Materiais, Métodos e Metodologia

Como base, utilizou-se o *software* OpenFOAM. Esta escolha se deu pelas seguintes características: gratuito, de código aberto e sua base de código contempla uma ampla gama de problemas da mecânica do contínuo. Ele já possui, nativamente, módulos que permitem a modelagem de escoamentos com mudança de fase através do método da entalpia-porosidade: o *solver buoyantPimpleFoam* e o módulo *solidificationMeltingSource*. Também possui um *solver*, desenvolvido por terceiros, que trata do escoamento de fluidos viscoelásticos: o *viscoelasticFluidFoam*. Outros *solvers*, como o *solidDisplacementFoam*, também foram estudados.

O Método dos Volumes Finitos pode ser entendido como um caso particular da formulação de elementos finitos (onde a função peso é simplesmente igual a 1). É bastante utilizado em problemas de dinâmica dos fluidos computacional (CFD), podendo ser aplicado na solução de outras equações diferenciais. Problemas de escoamento com mudança de fase normalmente são tratados como um caso particular de CFD, sendo comum o emprego desta metodologia em sua modelagem. O OpenFOAM trabalha com discretização espacial exclusivamente em volumes finitos do tipo *cell-centered*.

O estudo de problemas de dinâmica dos fluidos computacional e do funcionamento do *buoyantPimpleFoam* e do *solidificationMeltingSource* foi realizado no período de agosto de 2020 a agosto de 2021. Neste período o foco foi ampliar o conhecimento sobre as possibilidades que estes módulos apresentam, implementando neles modelos mais sofisticados. Focou-se, inicialmente, no cálculo das tensões térmicas residuais. Por esta razão, foi estudada a modelagem de problemas de tensões em sólidos empregando a metodologia dos volumes finitos.

Alguns materiais sob processo de fundição são viscoelásticos. Este tipo de modelagem, porém, não é comum, devido à dificuldade do conhecimento dessas propriedades físicas e da complexidade deste modelo. A fim de desenvolver um modelo mais completo de análise de tensões residuais, foram feitos estudos na mecânica de sólidos viscoelásticos e no escoamento de fluidos viscoelásticos. Nesta vertente, foi estudado o *solver viscoelasticFluidFoam*.

Resultados e Discussões

O estudo do *solver viscoelasticFluidFoam* se deu com base em experimentos de *benchmark* presentes na literatura e no entendimento de sua modelagem matemática e numérica. Os resultados destes estudos atestam a capacidade do *solver* na resolução deste tipo de problema e foram apresentados na Jornada de Iniciação Científica do LNCC.

Os objetivos propostos foram de implementar um modelo de análise de tensões residuais nos módulos do OpenFOAM que tratam do escoamento de mudança de fase. Esta tarefa mostrou-se bem exigente, motivo pelo qual o foco da pesquisa durante este período se deu no estudo dos fundamentos da mecânica do contínuo e da arquitetura do OpenFOAM. Citam-se as principais atividades realizadas desde agosto de 2020:

- 1 Estudo e benchmarking do *solver viscoelasticFluidFoam*.
- 2 Apresentação na JICT de setembro de 2020.
- 3 Apresentação na JICT de fevereiro de 2020.
- 4 Participação nas Web Oficinas de Engenharia e Modelagem Matemática, no total de 20 aulas, com duração entre 2 a 3 horas por aulas.
- 5 Participação nos programas Trilhas do Conhecimento I e II, ministrados pela Wikki Brasil.

- 6 Participação no curso “Introdução à CFD usando OpenFOAM”, ministrado pela Escola Piloto Virtual, com carga horária de 5 horas.
- 7 Matrícula na disciplina de Mecânica do Contínuo, do Programa de Pós Graduação em Modelagem Computacional da UFJF, com carga horária de 45 horas.(em andamento)
- 8 Preparação de um curso de Modelagem Computacional junto ao GET Mecânica, a ser apresentado na XLIV Semana da Engenharia da UFJF.

Conclusões

Todo o trabalho realizado desde abril de 2019 até o presente momento resultou em um notável aprendizado sobre diferentes tópicos da mecânica do contínuo. Ressalta-se a natureza interdisciplinar das pesquisas realizadas, bem como da complexidade dos modelos físicos estudados. Foi consequência, também, a presença em eventos científicos, como a Jornada de Iniciação Científica do LNCC, onde foi uma das pesquisas destaque de 2020, e o XL CILAMCE (onde tivemos um artigo aceito).

A pesquisa com problemas de mudança de fase e o estudo do OpenFOAM mostraram-se um tema de estudos bem completo. Foi possível acumular uma literatura bastante interessante em diversos temas, consolidando conhecimentos que encontravam-se dispersos e pouco detalhados, como a análise de tensão em sólidos através do método dos volumes finitos e as metodologias para simular um problema de mudança de fase. Também foi possível obter-se uma experiência razoável no uso do OpenFOAM, um *software* ainda relativamente pouco documentado mas que apresenta grande potencial na pesquisa e desenvolvimento nacional: o fato de ser gratuito e de código aberto torna-o transparente e colaborativo por natureza, algo especialmente estratégico em um país como o Brasil.

Os resultados obtidos até aqui foram bons e bastante promissores. Tratou-se de um trabalho desafiador para aluno de graduação, exigindo não só bom entendimento em modelagem computacional, como também de programação em C++ e da arquitetura do código do OpenFOAM. A complexidade do que se almeja e o rápido progresso realizado até então resultou em um período de maior aprofundamento teórico, resultando em menos resultados de experimentos numéricos. Acredita-se que a continuação com dos estudos em modelagem computacional e no funcionamento do OpenFOAM, torne possível atingir objetivos de longo prazo propostos.

Referências

- Bergman, T. L., Incropera, F. P., DeWitt, D. P., & Lavine, A. S. (2008). *Fundamentos de Transferência de Calor e de Massa* (6 ed.). LTC.
- Brent, A. D., Voller, V. R., & Reid, K. T. (1988). Enthalpy-porosity technique for modeling convection-diffusion phase change: application to the melting of a pure metal. *Numerical Heat Transfer, Part A Applications*, 13, 297–318.
- Brooks, A. N., & Hughes, T. J. (1982). Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer methods in applied mechanics and engineering*, 32, 199–259.
- Gau, C., & Viskanta, R. (1986). Melting and solidification of a pure metal on a vertical wall. *Journal of Heat Transfer*, 108, 174–181.
- Ghia, U. K., Ghia, K. N., & Shin, C. T. (1982). High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *Journal of computational physics*, 48, 387–411.

- Holzmann, T. (2016). Mathematics, numerics, derivations and OpenFOAM®. *Loeben, Germany: Holzmann CFD*.
- Jasak, H., & Weller, H. G. (2000). Application of the finite volume method and unstructured meshes to linear elasticity. *International journal for numerical methods in engineering*, 48, 267–287.
- Maric, T., Hopken, J., & Mooney, K. (2014). *The OpenFOAM Technology Primer*. Sourceflux.
- Odone, M. W. (2014). *Estudos numéricos na solução de problemas de mudança de fase*. Tese de Mestrado, Universidade Federal de Juiz de Fora - Programa de Pós-Graduação em Modelagem Computacional, Juiz de Fora, Brasil.
- Odone, M. W., Barra, L. P., Sutter, L. F., & Toledo, E. M. (2015). Uma Implementação do Método dos Volume Finitos Para Resolução de Problemas de Mudança de Fase. Rio de Janeiro: XXXVI CILAMCE.
- Odone, M. W., Neto, O. V., Toledo, E. M., Barra, L. P., & Soares, A. C. (2019). A Study of Phase Change Cast Modelling Using Open-FOAM. Natal: XL CILAMCE.
- Odone, M. W., Toledo, E. M., Barra, L. P., & Da Silva, C. E. (2013). Um Estudo Numérico Sobre o Método da Entalpia-Porosidade na Solução de Problemas de Mudança de Fase. Pirenópolis: XXXIV CILAMCE.
- Rady, M. A., & Mohanty, A. K. (1996). Natural convection during melting and solidification of pure metals in a cavity. *Numerical Heat Transfer, Part A Applications*, 29, 49–63.
- Soares, A. C. (2017). *Modelagem numérica da solidificação em moldes: aplicação a granadas de grande calibre*. Tese de Mestrado, Universidade Federal do Rio de Janeiro - Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia, Rio de Janeiro, Brasil.
- Teskeredžić, A., Demirdžić, I., & Muzaferija, S. (2002). Numerical method for heat transfer, fluid flow, and stress analysis in phase-change problems. *Numerical Heat Transfer: Part B: Fundamentals*, 42, 437–459.
- Velasco, A. R. (7 de 2017). Boundary Conditions for Convective Heat Transfer in OpenFOAM. Fonte: https://foamingtime2.files.wordpress.com/2017/07/convection-bc_1.pdf
- Versteeg, W. (2007). *An introduction to computational fluid dynamics: the finite volume method*. Pearson education.
- Voller, V. R., Brent, A. D., & Prakash, C. (1989). The modelling of heat, mass and solute transport in solidification systems. *International Journal of Heat and Mass Transfer*, 32, 1719–1731.
- Wolff, F., & Viskanta, R. (1988). Solidification of a pure metal at a vertical wall in the presence of liquid superheat. *International journal of heat and mass transfer*, 31, 1735–1744.
- Favero, J. L., Secchi, A. R., Cardozo, N. S. M., & Jasak, H. (2010). Viscoelastic flow analysis using the software OpenFOAM and differential constitutive equations. *Journal of non-newtonian fluid mechanics*, 165(23-24), 1625-1636.
- Favero, Jovani Juiz. *Simulação de escoamentos viscoelásticos: desenvolvimento de uma metodologia de análise utilizando o software OpenFOAM e equações constitutivas diferenciais*. Dissertação de Mestrado, UFRGS. 2009.