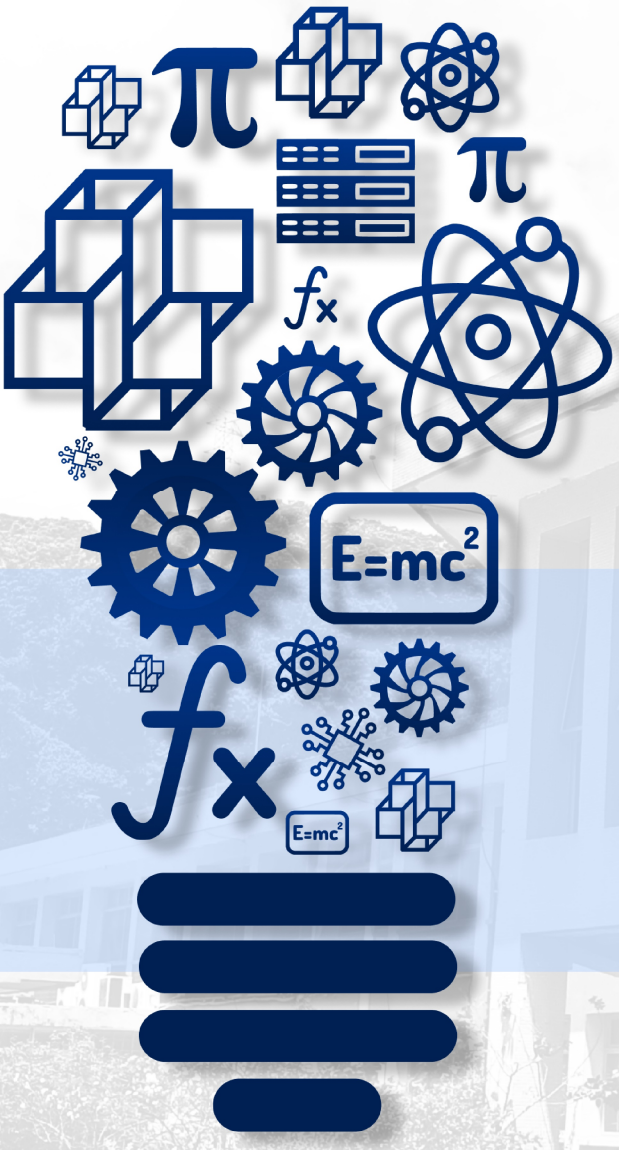
Jornada de Iniciação Científica e Tecnológica

2024



PIBIC · PIBITI

LNCC/MCTI

27 de agosto

CNPq
Conselho Nacional de Desenvolvimento Científico e Tecnológico

LNCC
Laboratório Nacional de Computação Científica

MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO

GOVERNO FEDERAL
BRASIL
UNIÃO E RECONSTRUÇÃO

Jornada de Iniciação Científica e Tecnológica do LNCC

Petrópolis, 27 de agosto de 2024.

Laboratório Nacional de Computação Científica – LNCC

Diretor
Fabio Borges de Oliveira

Coordenação de Gestão e Administração - COGEA
Paulo César de Freitas Honorato

Coordenação de Pós-Graduação e Aperfeiçoamento - COPGA
Antônio André Novotny

Programa Institucional de Bolsas de Iniciação Científica&
Programa Institucional de Bolsas de Iniciação em Desenvolvimento Tecnológico e Inovação
José Karam Filho

Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq

Presidente
Ricardo Magnus Osório Galvão

Coordenadora Geral do PIBIC/PIBITI
Lucimar Batista de Almeida

Jornada de Iniciação Científica e Tecnológica do LNCC

Comissão Interna do PIBIC/PIBITI-LNCC

José Karam Filho
Antônio Tadeu Azevedo Gomes
Eduardo Lucio Mendes Garcia
Fábio Lima Custódio
Jack Baczynski

Avaliadores Externos

Leandro Tavares da Silva – CEFET
Priscila Vanessa Zabala Capriles Goliatt UFJF

Apresentação

O LNCC realiza este ano a XXI 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 2023 a agosto de 2024, o PIBIC e o 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, neste ano presencialmente, os trabalhos são apresentados pelos bolsistas em forma de pôster e avaliados por um comitê científico externo.

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

Prof.ª Priscila Vanessa Zabala Capriles Goliatt - UFJF
Prof. Leandro Tavares da Silva - CEFET

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 de 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 da dedicação de todos os participantes.

Agradecimentos

Agradecemos ao CNPq pelas bolsas concedidas, à Direção e ao SECIN do LNCC pelo apoio, ao Programa de Pós-Graduação do LNCC e à Comissão Interna do PIBIC e PIBITI do 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.

José Karam Filho
Coordenador do PIBIC/PIBITI - LNCC

Índice

Bolsistas PIBITI ativos

Estudo e implementação de Sistema de Banco de Dados para Análise em Saúde Coletiva
Bolsista: Alan de Souza Mello
Orientadores: José Karam Filho e Paulo Cabral Filho

Pesquisa em GPU: Produto Escalar
Bolsista: Alessandra Almeida de Souza Lima
Orientadores: Carla Osthoff Ferreira de Barros e Micaella Coelho

Uso do Padrão de Paralelismo de Linguagem em Esquema de Radiação da Atmosfera
Bolsista: Gabriel Thomaz do Nascimento
Orientadores: Roberto Pinto Souto e Eduardo Lucio Mendes Garcia

Prevenção de Incidentes de Segurança no Ciberespaço, com a utilização de técnicas de rastreamento, através de Fluxos de Pacotes
Bolsista: Gabriel Martins
Orientadores: Marita Maestrelli e Antonio Tadeu Azevedo Gomes

Implementação de Gêmeo Digital para a Indústria de Petróleo
Bolsista: Stefanni Kapps
Orientador: Jauvane Cavalcante de Oliveira

Bolsistas PIBIC ativos

Segmentação Semântica para Geração de Mapas de Cobertura e Uso do Solo para o Território da APA Petrópolis
Bolsista: Alan Daiki Suga
Orientadores: Gilson Antônio Giraldi e Paulo Sérgio Silva Rodrigues

Comparação do Desempenho Computacional de Workflows Científicos de transcriptômica entre Arquiteturas de Memória Distribuída do Supercomputador Santos Dumont
Bolsita: Albert Siqueira Cosme Emidio
Orientadores: Kary Ann del Carmen Ocaña Gautherot, Carla Osthoff Ferreira de Barros

Estudo e implementação de mecanismo de aceleração de desempenho de experimentos científicos e simulações através de Inteligência Artificial
Bolsista: Alexandre de Paiva Almeida
Orientador: Jauvane de Oliveira Cavalcante

Comportamento Estrutural de Dutos Flexíveis no Transporte de Petróleo Offshore
Bolsista: Alexandre Vitor Silva Braga
Orientadores: Eduardo Lucio Mendes Garcia, Elson Magalhães Toledo e Marcos Vinicius Rodrigues

Desenvolvimento de Jogos para o Ensino-Aprendizagem
Bolsista: Amanda Vilas Boas Oliveira
Orientador: Regina Célia Cerqueira de Almeida e Crysttian Arantes Paixão

Análise Computacional da Dinâmica Molecular da Enxima N-acetyl-l-glutamate Kinase associada à super-resistência Bacteriana em Klebsiella Pneumoniae
Bolsista: Ana Júlia Ferreira Fagundes
Orientadores: Laurent Emmanuel Dardenne e Leon Sulfierry Corrêa Costa

Modelagem Computacional para Análise de Dados de Tratamento de Câncer em Saúde Coletiva
Bolsista: Ana Paula de Oliveira Souza
Orientadores: José Karam Filho e Paulo Cabral Filho

Assimilação de Dados no Modelo WRF para Uso Operacional em Previsão de Tempo e Simulação
Bolsista: Gabriel Costa Chaves
Orientadores: Roberto Pinto Souto, Juliana Aparecida Anochi e Helaine Cristina Moraes Furtado

Desenvolvimento de Programa para Simulação Numérica pelo Método dos Elementos Finitos Clássicos
Bolsista: Gabriel Massi de Moura
Orientador: Eduardo Garcia

Análise da aplicação BEAST 1.10 no ambiente nó Sequana do SDumon
Bolsista: Guilherme Freire da Silva Dornelas
Orientadores: Carla Osthoff Ferreira de Barros, Kary Ann del Carmen Ocaña Gautherot e Micaella Coelho Valente de Paula

Metodologia de Auditoria de Código e Planejamento de Otimização aplicada no Núcleo Dinâmico do Modelo MONAN
Bolsista: Isabel de Freitas Barboza
Orientadores: Roberto Pinto Souto e Eduardo Lucio Mendes Garcia

Simulação 3D do Voo Inaugural de Santos Dumont no 14 Bis
Bolsista: Jonatas Halliday Sant Anna do Nascimento
Orientador: Jauvane Cavalcante de Oliveira

Seleção de Conformações Representativas da Proteína Mpro do Vírus SARS-CoV-2
Visando a Descoberta de Novos Tratamentos para a COVID-19
Bolsista: Julia Francisco dos Santos
Orientadores: Laurent Emmanuel Dardenne e Matheus Muller

Desenvolvimento de um conjunto de teste para complexos de inclusão de ciclodextrinas. Otimização e validação por meio do DockThor.
Bolsista: Juliana do Nascimento Amaral
Orientadores: Laurent Emmanuel Dardenne e Isabella Alvim Guedes e Letícia de Assis

Avaliação de métodos de aprendizado de máquina em gateways científicos de bioinformática
Bolsista: Kelen Brito Souza
Orientadores: Kary Ann del Carmen Ocaña Gautherot, Carla Osthoff Ferreira e Micaella Coelho Valente

Pesquisa em GPU: Multiplicação de matrizes densas
Bolsista: Luiza Augusto Tavares
Orientadores: Carla Osthoff Ferreira de Barros e Micaella Coelho Valente

Estudo em GPU: Multiplicação de Matrizes Esparsas x Densas
Bolsista: Mariana Aguiar Ribeiro
Orientadores: Carla Osthoff Ferreira de Barros e Micaella Coelho Valente

Comparação do Desempenho Computacional de Workflows Científicos de Transcriptômica em Arquiteturas HPC
Bolsista: Reiglan Soares Di Lourenço
Orientadores: Kary Ann del Carmen Ocaña Gautherot, Carla Osthoff Ferreira de Barros

Introdução à Programação com Scratch e Tutoriais
Bolsista: Ricardo dos Santos de Lima
Orientador: Regina Célia Cerqueira de Almeida

Modelagem de Vigas de Euler Bernoulli e Timoshenko Equações Diferenciais Parciais
Bolsista: Tarsiane Ribeiro da Costa
Orientador: Jaime Edilberto Munoz Rivera

Inteligência Artifical Aplicada ao Diagnóstico por Imagem
Bolsista: Matheus Molina Alves Lima
Orientador: Bruno Schulze

Relatório de Atividades - Bolsa PIBITI

Dados Gerais

Projeto: Estudo e implementação de Sistema de Banco de Dados para Análise em Saúde Coletiva
Bolsista: Alan de Souza Mello
Orientador: José Karam Filho
Coorientador: Paulo Cabral Filho
Tipo de Bolsa: PIBITI
Período do Relatório: agosto/2023 a julho /2024

Objetivos

Este projeto tem como objetivo ajudar na implementação da tese de doutorado do coorientador, com foco na disseminação de informações em saúde, com base nos sistemas oficiais do SUS, sobre internações por câncer no Brasil desde 2011. O objetivo é estudar e implementar um sistema de banco de dados para análise de saúde coletiva, utilizando ferramentas como MySQL, conversão de formatos, geração de banco de dados, e desenvolvimento de um sistema piloto de monitoramento.

Introdução

A necessidade de uma plataforma computacional flexível orientada a Web, que permita a exportação de dados armazenados no DataSus para outros sistemas de análises, tais como Excel, Tableau ou Power BI, se mostrou evidente. Essa plataforma deverá permitir a criação de um painel de indicadores baseado em dados do SUS, disponibilizando-os via internet para a comunidade acadêmica e equipes de gestão dos governos municipais, estaduais e federal.

Material e Métodos

No período recente do projeto, diversas atividades foram realizadas com foco na melhoria e expansão do sistema de banco de dados e na disseminação das informações de saúde. Primeiramente, desenvolvemos o site "gestao-saude.lncc.br" utilizando as tecnologias React e Next.js. React é uma biblioteca JavaScript para construção de interfaces de usuário, conhecida por sua eficiência e flexibilidade, enquanto Next.js é um framework de desenvolvimento baseado em React que permite a renderização do lado do servidor e a geração de sites estáticos.

Para a construção do site, foi realizado um estudo na documentação oficial de React [1] e Next.js [2] além de cursos em vídeos que foram essenciais para o aprendizado e desenvolvimento das tecnologias[3] [4].

No site estamos com uma parceria com o Instituto de Medicina Social Hesio Cordeiro (IMS) da UERJ e a Faculdade de Educação Tecnológica do Estado do Rio de Janeiro (FAETERJ) de Petrópolis, com o objetivo de demonstrar não apenas este projeto, mas também outras linhas de pesquisa. O uso de React e Next.js proporcionou uma interface dinâmica e responsiva, facilitando a visualização dos dados.

Para organizar e facilitar o desenvolvimento do site, utilizamos o sistema de controle de versão GIT. O GIT é uma ferramenta de versionamento de código que permite rastrear mudanças no código-fonte durante o desenvolvimento do software, facilitando o trabalho em equipe e o gerenciamento de diferentes versões do projeto. Para o aprendizado e uso eficaz do GIT, estudamos a documentação oficial [5] e assistimos a cursos em vídeos específicos sobre a ferramenta [6].

Criamos um repositório remoto no GitHub, que é uma plataforma de hospedagem de código-fonte com controle de versão usando GIT. O GitHub permite que desenvolvedores colaborem em projetos de qualquer lugar do mundo, oferecendo funcionalidades para gerenciamento de projetos e integração contínua. O repositório do nosso projeto pode ser acessado em *https://github.com/alansmello/projetoLNCC*.

Uma parceria com o setor de helpdesk do LNCC foi crucial para a publicação do site. Com a colaboração e o auxílio deste setor, conseguimos garantir que a publicação ocorresse de forma eficaz. Atualmente, o site está disponível em ambiente interno do LNCC, pode ser acessado no *endereço* *http://gestao-saude.lncc.br/* , ainda em fase de desenvolvimento.

Atualizamos o banco de dados com os registros referentes ao ano de 2023, ampliando assim a base de dados disponível para análise. Além disso, a tabela IPCA foi atualizada com índices específicos de saúde e com índices até o ano de 2024, permitindo análises econômicas mais precisas.

Novos campos foram adicionados à tabela principal (INTERNACAO), incluindo VAL_SH_COR_GERAL, VAL_SP_COR_GERAL, VAL_TOT_COR_GERAL, VAL_UTI_COR_GERAL, VAL_SH_COR_SAUDE, VAL_SP_COR_SAUDE, VAL_TOT_COR_SAUDE, e VAL_UTI_COR_SAUDE. Esses campos são decorrentes do cálculo dos valores já existentes na tabela nos campos VAL_SH, VAL_SP, VAL_TOT, e VAL_UTI, aplicando os índices da tabela IPCA. Esses cálculos foram realizados seguindo a documentação de cálculo disponibilizada pelo nosso coorientador Paulo Cabral Filho, disponível no site mencionado e acessível em http://gestao-saude.lncc.br/ipca [7]. Esses campos referem-se a gastos hospitalares, médicos e de UTI.

Iniciamos a fase de análise de dados, realizando diversas consultas ao banco de dados para a geração de gráficos que serão exibidos no site. Estes gráficos oferecem visualizações detalhadas e intuitivas das informações de saúde coletiva, permitindo uma análise mais clara e acessível dos dados. Por exemplo, no site *gestao-saude.lncc.br/oncologia*, é possível visualizar gráficos que mostram a distribuição de internações por câncer ao longo dos anos, comparações de gastos hospitalares e médicos, bem como análises de tendências em tratamentos de UTI. Essas visualizações são fundamentais para identificar padrões e tomar decisões informadas sobre políticas de saúde.

Uma das melhorias significativas neste período de atividades foi a atualização dos scripts de transferência de dados. Anteriormente, utilizávamos 27 scripts distintos

para transferência de dados de cada estado, o que demandava mais tempo e esforço para manutenção. Desenvolvemos um único script otimizado que automatiza a busca e a transferência dos dados de arquivos CSV para o banco de dados, melhorando significativamente o desempenho e a eficiência do processo.

Por fim, estamos no desenvolvimento de uma documentação detalhada do projeto, incluindo instruções claras e abrangentes para orientar eventuais novos bolsistas na criação e atualização do banco de dados, bem como na atualização do site. Esta documentação incluirá informações sobre as tecnologias utilizadas, procedimentos de configuração e execução, além de orientações para a resolução de possíveis problemas.

Resultados e Discussão

Os resultados alcançados durante este período do projeto foram substanciais e significativos. A criação e publicação do site "gestao-saude.lncc.br" proporcionaram uma plataforma robusta para a disseminação de informações de saúde coletiva. O site, desenvolvido com React e Next.js, oferece uma interface intuitiva, facilitando o acesso aos dados e gráficos gerados a partir das análises.

A atualização do banco de dados com os registros de 2023 expandiu a base de dados, permitindo uma análise mais abrangente e atualizada. A inclusão dos novos campos na tabela ‘INTERNACAO’ e a atualização da tabela IPCA com índices específicos de saúde proporcionaram uma maior precisão nas análises econômicas. Esses novos campos, que registram os gastos hospitalares, médicos e de UTI, e suas atualizações conforme os índices IPCA, são essenciais para uma análise detalhada e precisa dos dados.

A otimização dos scripts de transferência de dados resultou em uma melhoria significativa no desempenho do processo. Com a redução de 27 scripts para um único script automatizado, conseguimos aumentar a eficiência e facilitar a manutenção das transferências de dados. Este novo script agiliza a inserção de dados de arquivos CSV

no banco de dados, garantindo que os dados sejam atualizados de uma forma mais rápida e eficiente.

A fase de análise de dados iniciou com sucesso, e as consultas SQL realizadas permitiram a geração de gráficos e indicadores relevantes para a área de saúde coletiva. Estes gráficos são exibidos no site, oferecendo uma visualização clara e acessível das informações de saúde.

A documentação detalhada do projeto garantirá que eventuais novos bolsistas possam continuar o trabalho de forma eficiente, assegurando a continuidade do projeto. Esta documentação é fundamental para a manutenção e atualização contínua do sistema, bem como para a capacitação de novos membros da equipe.

Conclusões

O projeto teve sucesso na atualização e expansão do sistema de banco de dados para análise de saúde coletiva. A criação do site *gestao-saude.lncc.br*, a atualização do banco de dados e a otimização dos scripts de transferência de dados aumentaram a eficiência e a funcionalidade do sistema. As novas atividades realizadas possibilitaram análises mais detalhadas e precisas, contribuindo significativamente para o avanço na disseminação de informações em saúde coletiva. A documentação que está sendo criada vai garantir a continuidade do projeto, facilitando a integração de novos bolsistas e assegurando a manutenção e atualização do sistema.

Referências Bibliográficas

1. React. React documentation. Disponível em: *https://react.dev/learn*;
2. Next.js. Next.js documentation. Disponível em: *https://nextjs.org/docs*;
3. Curso de React e NextJs – CFBCursos - YouTube. Disponível em: *https://www.youtube.com/watch?v=k9IG_tPonwo&list=PLx4x_zx8csUhxQHo-w0xAIMrmt8WgAowL*;
4. Curso de Next.js – Matheus Battisti -Hora de Codar - YouTube. Disponível em: *https://www.youtube.com/watch?v=XHrbg2iYNCg&list=PLnDvRpP8BnezfJcfiClWskFOLODegI_Ft* ;
5. GIT. GIT documentation. Disponível em*: https://git-scm.com/doc*;
6. Curso GIT - Tiago Matos - YouTube. Disponível em: *https://www.youtube.com/watch?v=2c7yWlpWDJM&list=PLcoYAcR89n-gbO7YAVj5S0alABLis_QVU*
7. Cabral Filho, P. Cálculo dos índices IPCA. Disponível em: *http://gestao-saude.lncc.br/ipca*;

PROGRAMA DE BOLSAS PCI/LNCC

Relatório Final de Atividades

TÍTULO DO PROJETO: Pesquisa em GPU: Produto Escalar

Bolsista:Alessandra Almeida de Souza Lima

Orientador: Carla Osthoff Ferreira de Barros e Micaella Coelho

Nível de bolsa: PIBIC

Período de bolsa: Outubro de 2023 a Julho de 2024

1 Objetivo

O objetivo deste projeto foi analisar e comparar o desempenho de duas GPUs, a GeForce GTX 1660 e a NVIDIA V100, durante a execução de operações de produto escalar utilizando a função `cublas<t>dot` da biblioteca cuBLAS. A GTX 1660, eficiente para jogos, possui 1408 núcleos CUDA e 6 GB de memória GDDR5. A análise focou na escalabilidade e eficiência da função em diferentes tamanhos de vetores.

2 Introdução

Este relatório descreve a pesquisa realizada entre Outubro de 2023 a Julho de 2024, cujo objetivo foi explorar a utilização da biblioteca cuBLAS, uma implementação dos BLAS (Basic Linear Algebra Subprograms) na plataforma CUDA™ da NVIDIA. A cuBLAS permite acessar os recursos computacionais da GPU para operações de álgebra linear de alto desempenho.

Durante este projeto, utilizamos a função `cublas<t>dot`, que calcula o produto escalar entre dois vetores. Esta função é importante em várias aplicações científicas e de engenharia, realizando a multiplicação e soma dos elementos correspondentes de dois vetores.

Para testar o desempenho da `cublas<t>dot`, geramos vetores de entrada de tamanhos variados, entre 500 e 500000 elementos para avaliar a escalabilidade e eficiência da função Visamos comparar o desempenho desta função em duas arquiteturas de GPUs distintas: a GeForce GTX 1660 e a NVIDIA V100. A GeForce GTX 1660 é conhecida por sua eficiência energética e adequação para jogos e criação de conteúdo, enquanto a NVIDIA V100,

projetada para data centers e aplicações de inteligência artificial, destaca-se pela sua capacidade de processamento avançada. Neste primeiro momento, será apresentado o desempenho na GeForce GTX 1660.

Os resultados desta pesquisa são importantes para uma compreensão das capacidades e limitações das arquiteturas de GPUs na execução de operações de álgebra linear, auxiliando identificar as vantagens e limitações de cada sistema ao lidar com cálculos intensivos.

3 Métodos

3.1 cuBLAS

A biblioteca cuBLAS (CUDA Basic Linear Algebra Subprograms) é uma biblioteca desenvolvida pela NVIDIA, projetada para executar operações de álgebra linear básica de forma otimizada em GPUs NVIDIA CUDA. Ela oferece uma variedade de funções que são essenciais para aplicações que exigem alto desempenho computacional em operações matriciais e de vetor, como multiplicação de matriz-vetor (gemv), multiplicação de matriz-matriz (gemm) e operações de decomposição (como LU e QR).

Dentro desse conjunto de funções, a `cublasDot` se destaca por sua função específica de calcular o produto escalar entre dois vetores. O produto escalar é uma operação fundamental na álgebra linear, calculando a soma dos produtos dos elementos correspondentes de dois vetores. Este cálculo é usado extensivamente em várias aplicações científicas e numéricas, como métodos de otimização, processamento de sinais, aprendizado de máquina, entre outros.

A `cublasDot` é implementada de forma a tirar máximo proveito da arquitetura paralela das GPUs CUDA. Isso significa que, em vez de processar cada elemento sequencialmente, a operação é distribuída entre os muitos núcleos de processamento da GPU, permitindo uma execução muito mais rápida e eficiente do que seria possível em CPUs convencionais para conjuntos de dados grandes.

Portanto, a `cublasDot` não apenas oferece uma maneira eficiente de calcular o produto escalar, mas também demonstra o poder das GPUs CUDA para acelerar operações matemáticas intensivas, proporcionando ganhos significativos de desempenho em

comparação com implementações puramente CPU. Isso faz dela uma escolha para desenvolvedores e pesquisadores que buscam maximizar o desempenho computacional em suas aplicações de álgebra linear.

3.2 Ambiente Computacional

O ambiente utilizado para os experimentos incluiu a GPU da máquina local da sala do CENAPAD, equipada com uma GeForce GTX 1660 baseada na arquitetura Turing, com 6 GB de memória.

4 Resultados

Os resultados das execuções dos vetores na GPU foram coletados e organizados em uma planilha de dados. Com base nesses dados (Tempo de transferência, Kernel e transferência + Kernel), foram gerados gráficos individuais para cada tempo registrado.

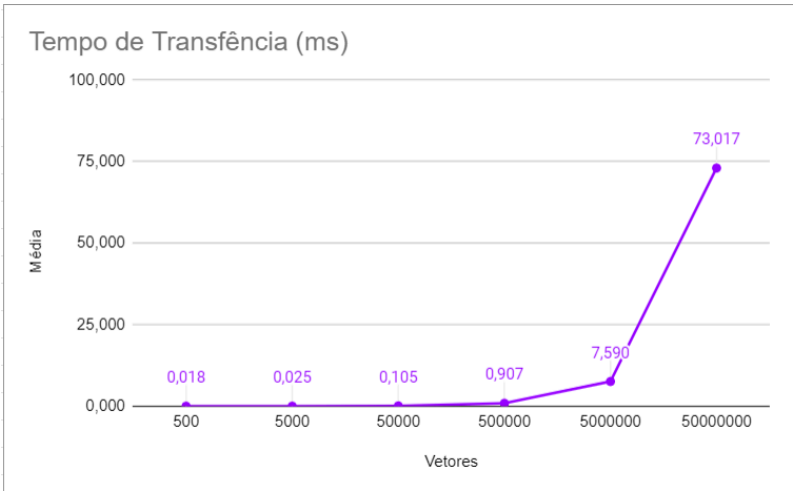


Figura 1 ilustra os tempos de transferência entre CPU e GPU, mutuamente, com as dimensões das matrizes no eixo x e os tempos em milissegundos no eixo y. Pode ser observado que o tempo de transferência aumenta com o crescimento do tamanho dos vetores, indo de 0,018 milissegundos com 500 elementos para 73,017 milissegundos com 50000000 elementos.

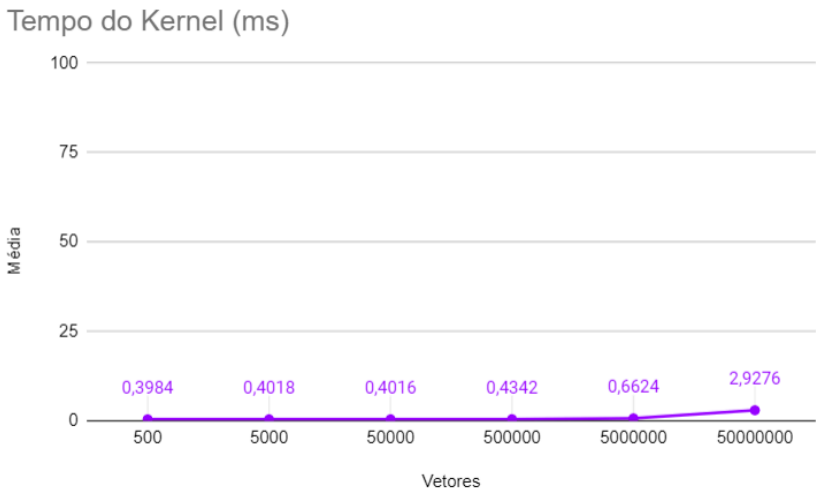


Figura 2 ilustra os tempos de transferência entre CPU e GPU, mutuamente, com as dimensões das matrizes no eixo x e os tempos em milissegundos no eixo y. Pode ser observado que o tempo de transferência aumenta com o crescimento do tamanho dos vetores, indo de 0,3984 milissegundos com 500 elementos para 2,9276 milissegundos com 50000000 elementos.

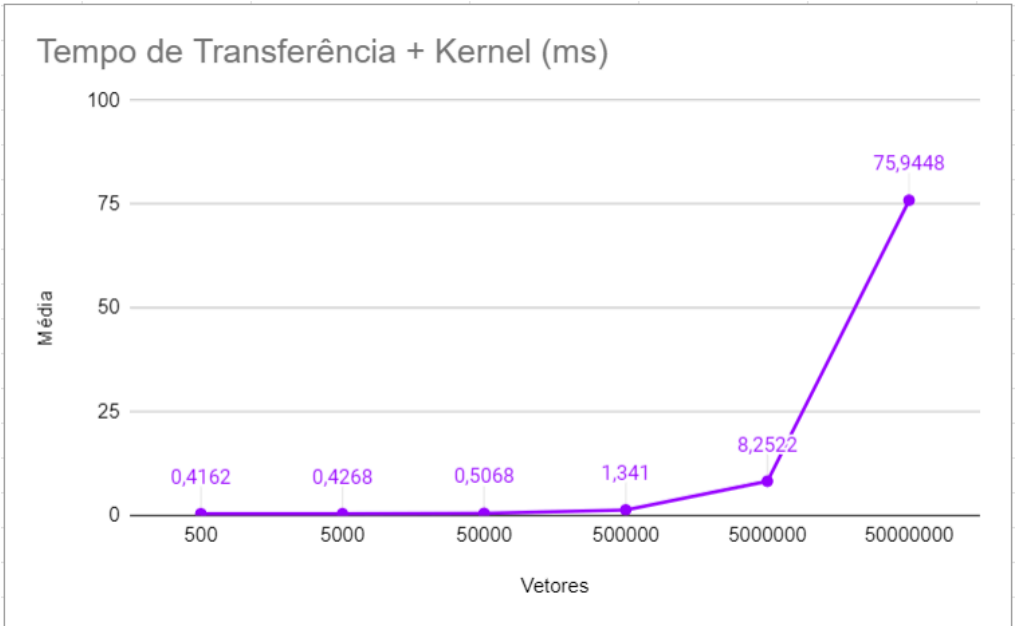


Figura 3 ilustra os tempos de transferência entre CPU e GPU, mutuamente, com as dimensões das matrizes no eixo x e os tempos em milissegundos no eixo y. Pode ser observado que o tempo de transferência aumenta com o crescimento do tamanho dos vetores, indo de 0,4162 milissegundos com 500 elementos para 75,9448 milissegundos com 50000000 elementos.

5 Conclusão

Os experimentos realizados utilizando a biblioteca cuBLAS e a função cublasDot para calcular o produto escalar demonstraram variações no desempenho conforme o tamanho dos vetores. Observou-se que a GeForce GTX 1660 apresentou tempos de transferência inferiores aos tempos de processamento do kernel para vetores de menor tamanho. No entanto, para vetores maiores, variando entre 500.000 e 50.000.000 de elementos, o tempo de transferência superou o tempo de processamento. Como próximos passos, planejamos executar os mesmos experimentos utilizando a GPU V100 do supercomputador SDumont. Isso nos permitirá avaliar o desempenho e a escalabilidade da função nesta arquitetura mais avançada e comparar os resultados obtidos entre as duas plataformas de hardware.

6 Referências Bibliográficas

NVIDIA Developer Documentation. cuBLAS Library Documentation. Disponível em: NVIDIA cuBLAS Documentation.

NVIDIA Developer Blog. Accelerating AI with cuBLAS: NVIDIA's GPU-accelerated Matrix Multiplication Library. Disponível em: NVIDIA cuBLAS Blog.

NVIDIA Developer Forums. Discussions and Resources on cuBLAS. Disponível em: NVIDIA Developer Forums.

NVIDIA CUDA https://docs.nvidia.com/cuda/cublas/index.html#cublas-t-dot

1) Dados Gerais

Título do projeto: Uso do padrão de paralelismo de linguagem em esquema de radiação da atmosfera
Bolsista: Gabriel Thomaz do Nascimento
Orientadores: Roberto Pinto Souto, Eduardo Lucio Mendes Garcia
Tipo de bolsa: PIBITI/CNPq
Período do Relatório: Junho 2023 a Julho de 2024
Início do período de bolsa: 01/06/2023

2) Objetivos

Objetivos abordados durante o período da pesquisa:

- Obter perfil de desempenho da aplicação com ferramentas de avaliação de desempenho
- Obter resultado com paradigma de paralelismo nativo da linguagem Fortran

3)Introdução: O *ECRAD* é um modelo de radiação para utilização em sistemas de previsão do tempo integrados sustentado pelo *ECMWF* (European Centre for Medium-Range Weather Forecasts) que trabalha seus solvers de forma independente e que é pioneiro na representação 3D de efeitos de radiação. Nesta etapa da pesquisa, objetivou-se a iniciação ao uso de paralelismo nativo na linguagem *Fortran* implementado a rotinas dos solvers do *ECRAD*. Visando o ganho de desempenho paralelo em GPU, utilizou-se a estrutura DO CONCURRENT aliada ao compilador *PGI*, da *NVIDIA*. Sob a ótica de ferramentas de análise de desempenho, tal como o *HPCToolkit,* foi possível mensurar a viabilidade do uso de paralelismo expresso por GPU's sem diretivas exteriores. Foram estabelecidas algumas etapas para se chegar ao estado atual:

1. Estabelecer o domínio espacial operacional de cada solver e estudar a saída de seus resultados.
2. Estudar o uso de ferramentas e métricas de avaliação de desempenho.
3. Observar as rotinas com potencial paralelismo.
4. Analisar as configurações referentes à entrada de dados, processamento, estruturas de armazenamento e possíveis abordagens/parametrizações para observar um ganho relevante de desempenho (escalabilidade).
5. Manipular corretamente o uso de ferramentas, rotinas e os parâmetros de compilação referentes.

4) Metodologia

Standard Parallelism e DO CONCURRENT:

As linguagens padrão de programação começaram a adicionar recursos que os compiladores podem usar para programação paralela de GPU e CPU, como por exemplo, o DO CONCURRENT e coarray em *Fortran* [1]. O uso de recursos de linguagem padrão, tais como o DO CONCURRENT, tem como uma de suas vantagens o suporte por um longo prazo da comunidade Fortran. A partir da versão 20.11 do *NVIDIA HPC SDK*, o compilador *NVFORTRAN* incluído acelera automaticamente rotinas contendo DO CONCURRENT, permitindo que você utilize GPUs *NVIDIA* com *Standard Fortran* sem quaisquer extensões, diretivas ou bibliotecas. [2]

Exemplo: Método Jacobi:

O método de Jacobi é um processo iterativo usado para resolver sistemas lineares de equações. Ele opera decompondo a matriz de coeficientes em componentes que permitem a atualização simultânea de todas as incógnitas em cada iteração. A principal característica do método é que ele calcula cada valor de incógnita usando os valores da iteração anterior, o que permite a paralelização das atualizações.

```
! Jacobi iteration example using Do Concurrent in Fortran
! Build with to target NVIDIA GPU
!   nvfortran -stdpar -Minfo=accel -fast f3.f90
! Build with to target host multicore
!   nvfortran -stdpar=multicore -Minfo=accel -fast f3.f90
```

```
!
subroutine smooth( a, b, w0, w1, w2, n, m, niters )
 real, dimension(:,:) :: a,b
 real :: w0, w1, w2; integer :: n, m, niters, i, j, iter
  do iter = 1,niters
   do concurrent(i=2 : n-1, j=2 : m-1)
     a(i,j) = w0 * b(i,j) + &
              w1 * (b(i-1,j) + b(i,j-1) + b(i+1,j) + b(i,j+1)) + &
              w2 * (b(i-1,j-1) + b(i-1,j+1) + b(i+1,j-1) + b(i+1,j+1))
   enddo
   do concurrent(i=2 : n-1, j=2 : m-1)
     b(i,j) = w0 * a(i,j) + &
              w1 * (a(i-1,j) + a(i,j-1) + a(i+1,j) + a(i,j+1)) + &
              w2 * (a(i-1,j-1) + a(i-1,j+1) + a(i+1,j-1) + a(i+1,j+1))
   enddo
  enddo
end subroutine smooth

subroutine smoothhost( a, b, w0, w1, w2, n, m, niters )
 real, dimension(:,:) :: a,b
 real :: w0, w1, w2;  integer :: n, m, niters,i, j, iter
  do iter = 1,niters
   do i = 2,n-1
    do j = 2,m-1
     a(i,j) = w0 * b(i,j) - &
              w1 * (b(i-1,j) + b(i,j-1) + b(i+1,j) + b(i,j+1))! + &
              !w2 * (b(i-1,j-1) + b(i-1,j+1) + b(i+1,j-1) + b(i+1,j+1))
    enddo
   enddo
   do i = 2,n-1
    do j = 2,m-1
     b(i,j) = w0 * a(i,j) - &
              w1 * (a(i-1,j) + a(i,j-1) + a(i+1,j) + a(i,j+1))! + &
              !w2 * (a(i-1,j-1) + a(i-1,j+1) + a(i+1,j-1) + a(i+1,j+1))
    enddo
   enddo
  enddo
end subroutine smoothhost
```

Figura 1 - Rotinas `smooth` e `smoothhost`. A rotina `smooth` trabalha com a versão paralelizada por DO CONCURRENT, enquanto a versão `smoothhost` funciona em sua forma sequencial.

Experimentos envolvendo o método Jacobi com DO CONCURRENT foram realizados na plataforma Jupyter Notebook, nos quais diferentes opções de compilação foram utilizados e sob diferentes perspectivas de wall clock, onde podemos visualizar os resultados em tabela abaixo:

Tabela 1 - Resultados dos experimentos

nome do processo	compilador	opções	tempo de execução (microsegundos)	tempo de execução (segundos)
teste1	gfortran	-ftree-parallelize-loops=4	175 do concurrent, 4320 do classical	0.17969 do concurrent, 4.32031 do classical
teste2	gfortran	sem argumentos	177 do concurrent, 4398 do classical	0.17969 do concurrent, 4.39844 do classical
jacobi_gpu	nvfortran	-stdpar -Minfo=accel -fast	17556 do concurrent, 3072262 do classical	0.01563 do concurrent, 3.07031 do classical
jacobi_multicore	nvfortran	-stdpar=multicore -Minfo=accel -fast	3906113 do concurrent, 3994573 do classical	3.90625 do concurrent, 3.99219 do classical
jacobiNvidia	nvfortran	sem argumentos	4555385 do concurrent, 4446764 do classical	4.55469 do concurrent, 4.44531 do classical

Domínio e Discretização Aplicada

Os esquemas de radiação nos modelos de circulação geral fazem atualmente uma série de simplificações na contabilização das nuvens, sendo uma das mais importantes a negligência da heterogeneidade horizontal no que diz respeito a densidade das nuvens. (Shonk, J. K., & Hogan, R. J. (2008).) [3] Três solvers do *ECRAD* (*McICA, Tripleclouds e Homogeneous*) são baseados em uma formulação clássica 1D de two-stream tanto em ondas curtas quanto em ondas longas, mas com maneiras diferentes de tratar a estrutura da nuvem. O solver *SPARTACUS* (Speedy Algorithm for Radiative Transfer Through Cloud Sides) é um pouco mais complexo, e torna o *ECRAD* o primeiro modelo global capaz de representar os efeitos radiativos das nuvens em 3D. (Hogan, R. J., & Bozzo, A. (2018).) [4]

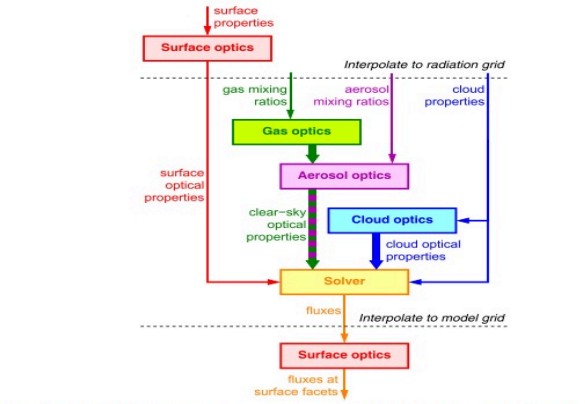


Figura 2 - Estrutura modular e hierárquica do modelo [4]

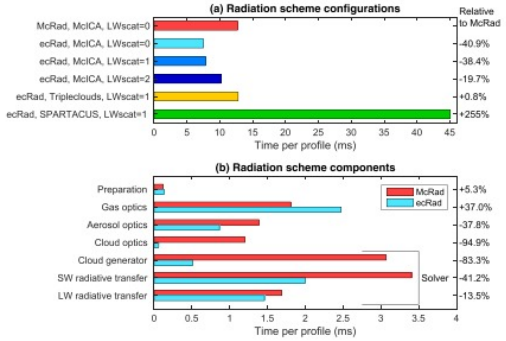


Figura 3 - Componentes e divisão de processos [4]

5) Resultados e discussões

A partir do diretório Practical, podemos obter alguns gráficos relativos à distribuição de efeitos de radiação sob a perspectiva de uma secção através da atmosfera abrangendo o Pólo Norte ao Pólo Sul a uma longitude de 5º ao leste às UTC +12 de 11 de julho de 2019. Os dados são armazenados em 137 níveis, da superfície ao meio da mesosfera (uma pressão de 1 Pa), e foram extraídos a cada 0.5º de intervalo de latitude.

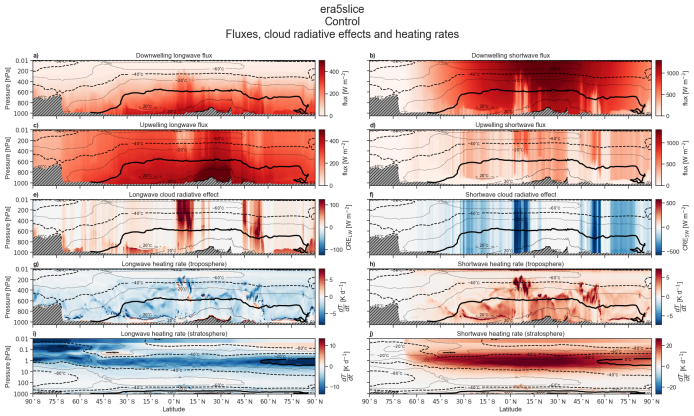


Figura 5 - Gráficos gerados pela saída do ECRAD a partir de um experimento padrão do diretório Practical (ecrad config.nam era5slice.nc control.nc).

Essas saídas são geradas por funções 2D (funções do tipo f(x,y) = z) e descrevem comportamentos como fluxos de onda longa e onda curta, taxas de aquecimento e efeitos de radiação de nuvem.

Paralelização de Rotinas

A partir de um fork do repositório do projeto, estabeleceram-se branches de trabalho com a finalidade de analisar o ganho de desempenho em relação ao código original. Os arquivos podem ser encontrados no *GitHub*: https://github.com/TempoHPC/ecrad. A branch ecrad_nvhpc contém a adaptação de compilação *PGI* para o *standard parallel* otimizado à GPUs *Nvidia* e também o script de submissão de jobs, que contém os módulos de compilação e também os responsáveis por carregar as ferramentas de avaliação de desempenho (*HPCToolkit*) e o *netCDF*, que é o tipo de arquivo responsável pela formatação dos dados. Na atual fase de pesquisa, diretórios de trabalhos locais desenvolveram-se sobre a branch ecrad_nvhpc, onde foram realizados testes com o compilador *PGI* sob os parâmetros noaccel, multicore e stdpar.

Utilizaram-se as ferramentas do *NCO* (netCDF Operators) para o uso do script duplicate_profiles.sh, que duplica um perfil individual (i3rc_mls_cumulus.nc) para cada ângulo de zênite solar. O arquivo se encontra segundo o path: test/i3rc. Um experimento então foi realizado a partir do arquivo gerado de 46 ângulos i3rc_mls_cumulus_46.nc com o arquivo de configuração configI3RC.nam, que usa o solver *SPARTACUS*.

Após visualização com a ferramenta *HPCViewer*, determinou-se a rotina *repeated_square* do arquivo radiation_matrix.F90 como potencial. Uma sequência de 3 loops que chamam a função de forma recursiva foi alterado

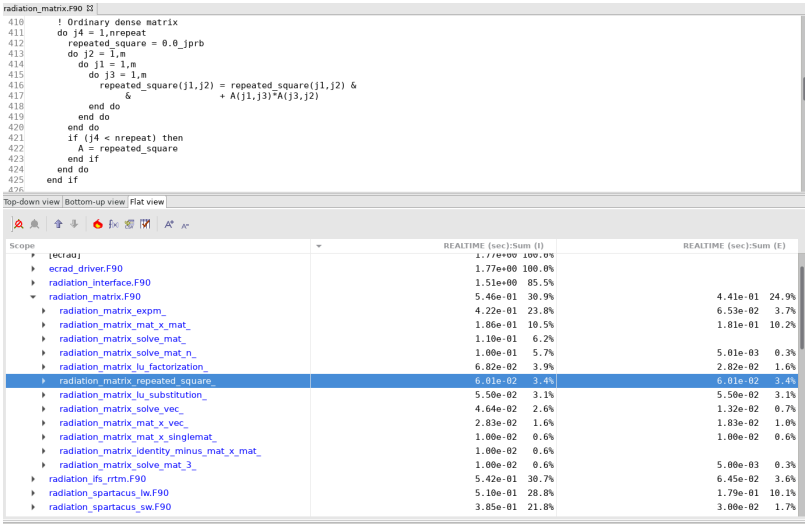


Figura 6 - Visualização do perfil de desempenho com HPCViewer, identificando o laço com potencial para uso de DO CONCURRENT

```
1       ! Ordinary dense matrix
2       do j4=1,nrepeat
3               repeated_square = 0.0_jprb
4               do j2 = 1,m
5                       do j1 = 1,m
6                               do j3 = 1,m
7                                       repeated_square(j1,j2) = repeated_square(j1,j2) &
8                                                               + A(j1,j3) A(j3,j2)
9                               end do
10                      end do
11              end do
12              if (j4 < nrepeat) then
13                      A = repeated_square
14              end if
15      end do
16 end if
```

Figura 7 - Estrutura de loop original

```
1       ! Ordinary dense matrix

2       do j4=1,nrepeat
3               repeated_square = 0.0_jprb
4               do concurrent(j2=1:m, j1=1:m, j3=1:m)
5                               repeated_square(j1,j2) = repeated_square(j1,j2) &
6                                                       + A(j1,j3) A(j3,j2)
7               end do
8               if (j4 < nrepeat) then
9                       A = repeated_square
10              end if
11      end do
12 end if
```

Figura 8 - Estrutura de loop original modificada com DO CONCURRENT

Na saída, alguns resultados foram obtidos através de jobs. Além disso, almejou-se modificar a parametrização de execução, com a finalidade de aumentar a carga computacional e observar com maior clarividência o ganho de desempenho. Através da perspectiva de visualização do HPCToolkit, obteve-se a seguinte saída para o experimento com o *i3rc_mls_cumulus_46.nc*, usando stdpar e nrepeat = 10:

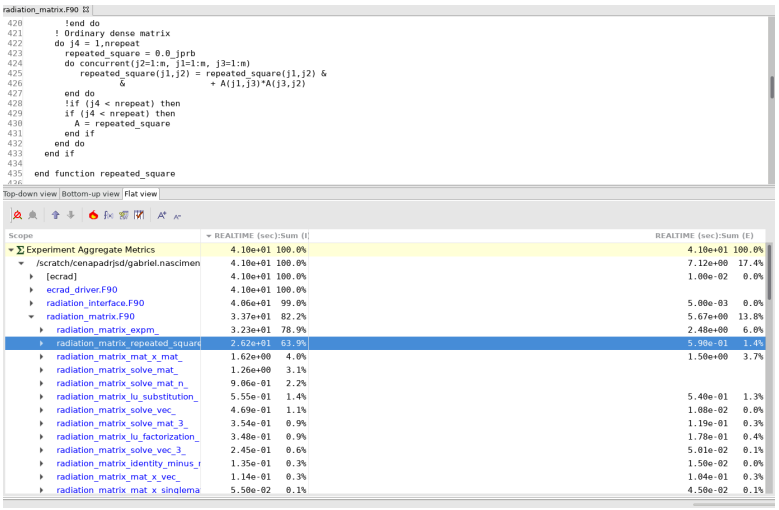


Figura 7 - Visualização do perfil de desempenho com HPCViewer, identificando o laço já alteradocom uso de DO CONCURRENT

Pode-se observar que seguida a alteração do loop, a rotina passou a consumir uma maior quantidade de tempo, acrescido também o seu consumo em termos de porcentagem. Isso se deve em grande parte a custos relacionados a parte de comunicação e transferência de dados para a GPU. Obteve-se êxito em conseguir compilar e executar as modificações realizadas, e os próximos passos estão relacionados a otimizar transferência dados, investigando outras possibilidades para implementação de paralelismo em GPU.

Bibliografia:

1. https://developer.nvidia.com/blog/using-fortran-standard-parallel-programming-for-gpu-acceleration/
2. https://developer.nvidia.com/blog/accelerating-fortran-do-concurrent-with-gpus-and-the-nvidia-hpc-sdk/
3. https://journals.ametsoc.org/view/journals/clim/21/11/2007jcli1940.1.xml?tab_body=pdf
4. https://www.met.reading.ac.uk/~swrhgnrj/publications/ecrad.pdf
5. https://www.met.reading.ac.uk/~swrhgnrj/publications/hogan_shonk_3d.pdf
6. https://pop-coe.eu/node/69
7. https://confluence.ecmwf.int/display/ECRAD/ecRad+practical
8. https://agupubs.onlinelibrary.wiley.com/doi/full/10.1029/2021MS002754
9. https://www.researchgate.net/publication/377060073_Twelve_Times_Faster_yet_Accurate_A_New_State-Of-The-Art_in_Radiation_Schemes_via_Performance_and_Spectral_Optimization
10. https://confluence.ecmwf.int/display/ECRAD?preview=/70945505/114525750/ecrad_documentation.pdf
11. https://confluence.ecmwf.int/display/ECRAD/ECMWF+Radiation+Scheme+Home

Prevenção de Incidentes de Segurança no Ciberespaço, com a utilização de técnicas de rastreamento, através de Fluxos de Pacotes

Aluno: Gabriel Martins - FAETERJ

Orientadora: Marita Maestrelli – Tecnologista Senior

Coorientador: Antonio Tadeu Azevedo Gomes

Objetivos

O objetivo deste trabalho é utilizar as ferramentas Netflow, Nfdump e Nfsen para monitorar, analisar e diagnosticar o tráfego de rede. O foco é obter visibilidade detalhada sobre o tráfego, identificar padrões e anomalias, diagnosticar problemas de desempenho e segurança, e auxiliar no planejamento de capacidade da infraestrutura de rede.

Introdução

Netflow é uma tecnologia desenvolvida pela Cisco para coletar informações detalhadas sobre o tráfego IP nas redes. Ele permite que administradores de rede obtenham dados sobre a origem e o destino do tráfego, a classe de serviço e as causas de congestionamento. Netflow é composto por três componentes principais: cache de fluxo, coletor de fluxo e analisador de dados.

Nfdump é uma ferramenta de linha de comando que processa e exibe dados coletados pelo Netflow, sendo parte do projeto Nfsen. O Nfsen, por sua vez, oferece uma interface gráfica para visualização dos dados do Netflow e possui funcionalidades adicionais como navegação fácil pelos dados, configuração de alertas e criação de plugins para processamento de dados.

Material e Métodos

Material

- Servidor para instalação e execução do Nfdump e Nfsen.

- Ambiente de rede para coleta de dados.

Metodologia

Configuração do Netflow :

- Habilitação do Netflow nas interfaces desejadas.

- Configuração do coletor de fluxo para direcionar os dados ao servidor que executa o Nfdump.

Instalação e Configuração do Nfdump:

- Instalação do Nfdump no servidor.

- Configuração do nfcapd para coletar e armazenar os dados de Netflow.

Instalação e Configuração do Nfsen:

- Instalação do Nfsen no servidor.

- Integração do Nfsen com o Nfdump para visualização gráfica dos dados.

- Configuração de alertas e perfis de monitoramento contínuo.

Coleta e Análise de Dados:

- Coleta de dados de tráfego em diferentes períodos (diário, semanal, mensal).

- Análise dos dados coletados utilizando o Nfdump e a interface gráfica do Nfsen.

- Identificação de padrões, anomalias e possíveis problemas de desempenho ou segurança.

Resultados e Discussão

Resultados

- Visibilidade de Tráfego : A utilização do Netflow permitiu uma visão detalhada do tráfego de rede, identificando padrões de uso e detectando anomalias.

- Diagnóstico de Problemas : Com o Nfdump e o Nfsen, é possível diagnosticar problemas de desempenho e congestionamento em pontos específicos da rede.

- Segurança : A análise dos dados ajuda na detecção de atividades suspeitas e possíveis ataques, permitindo ações preventivas.

- Planejamento de Capacidade : Os dados coletados facilitam o planejamento e dimensionamento da infraestrutura de rede, evitando sobrecargas e otimizando recursos.

Discussão

Os resultados obtidos confirmam a eficácia das ferramentas Netflow, Nfdump e Nfsen na monitorização e análise do tráfego de rede. A integração entre essas ferramentas permitiu uma abordagem abrangente e detalhada, proporcionando informações cruciais para a gestão da rede. A capacidade de configurar alertas e utilizar plugins personalizados aumentou a flexibilidade e adaptabilidade das ferramentas às necessidades específicas da rede monitorada.

Conclusões

O uso combinado de Netflow, Nfdump e Nfsen demonstrou ser uma solução robusta para monitorar, analisar e diagnosticar o tráfego de rede. Os principais benefícios incluem a visibilidade detalhada do tráfego, a capacidade de diagnosticar problemas de desempenho e segurança, e a ajuda no planejamento de capacidade. Essas ferramentas são essenciais para administradores de rede que buscam otimizar a performance e segurança de suas infraestruturas. A configuração adequada e o uso contínuo dessas ferramentas permitem uma gestão mais eficiente e proativa da rede, prevenindo problemas e melhorando a qualidade do serviço.

Planos Futuros

Para aprimorar ainda mais o monitoramento e a gestão da rede, planejo focar no desenvolvimento de plugins personalizados para o Nfsen. Esses plugins serão projetados para fornecer alertas específicos e adaptados às necessidades da rede, melhorando a capacidade de resposta a problemas emergentes.

Adicionalmente, pretendo explorar a integração de técnicas de Inteligência Artificial (IA) com essas ferramentas. Utilizando IA, será possível analisar grandes volumes de dados de forma mais eficiente e detectar padrões complexos que podem indicar problemas de desempenho ou segurança que passariam despercebidos com análises tradicionais. A IA pode também automatizar o processo de resposta a incidentes, sugerindo ou implementando soluções em tempo real.

A combinação de plugins personalizados e IA não só aumentará a eficácia das ferramentas Netflow, Nfdump e Nfsen, mas também proporcionará uma abordagem mais dinâmica e adaptativa para a gestão de redes, elevando o nível de segurança e performance das infraestruturas de TI.

Referências Bibliográficas:

1 – Netflow, protocolo de rede desenvolvido pela Cisco para coletar informações de tráfego IP e monitorar o fluxo de rede.

https://www.cisco.com/c/en/us/products/ios-nx-os-software/ios-netflow/index.html

2 – NFDUMP- ferramentas para captura do NETFLOW

https://sourceforge.net/projects/nfdump

3 – NFSEN – aplicação web para visualização e análise dos fluxos capturados pelo NFDUMP

https://sourceforge.net/projects/nfsen/

Relatório de Atividades

Implementação de Gêmeo Digital para a Indústria de Petróleo

Instituição
Laboratório Nacional de Computação Científica(LNCC)

Bolsista
Stefanni Kapps

Orientador
Jauvane C. de Oliveira

Tipo de Bolsa
PIBIT

Objetivos

O objetivo do projeto consiste em implementar um ambiente 3D que comporá o lado digital de um gêmeo digital para a área de petróleo. A etapa inicial consiste em estudar a engine Unity, a ser utilizada na implementação. O estudo inicial enfoca em compreender os fundamentos essenciais da engine Unity, e da linguagem de programação C#, a ser utilizados no projeto em si.

Introdução

A Unity é uma engine de desenvolvimento muito utilizada na criação de jogos e simulações por conta de sua variabilidade e capacidade de criar experiências imersivas em diversas plataformas. Seu software possui uma interface intuitiva e uma diversidade de recursos, incluindo uma física realista, sistema de áudio e diversas outras funcionalidades interativas.

Material e Métodos

Na etapa inicial deste projeto, as atividades se restringiram ao aprendizado das ferramentas necessárias à implementação desejada. Para esse estudo inicial, foi utilizado o material fornecido pela Unity pela plataforma Unity Learn. O curso "Unity Essentials" foi utilizado como base inicial. Ele abrange desde

os conceitos fundamentais até técnicas mais avançadas de desenvolvimento. O curso possui temas essenciais como criação de ambientes virtuais, programação em C# e implementação de mecânicas de jogo. Dentro dos cursos, o Unity Learn oferece vários projetos práticos que são desenvolvidos para estudo e adaptação durante as aulas.

O curso "Junior Programmer" foca no desenvolvimento de habilidades de programação dentro do ambiente Unity. Possui uma exploração mais profunda em assuntos como lógica de programação, controle de versão, otimização de código e programação orientada a objetos. Isso proporciona uma melhor compreensão e facilita a integração de scripts na plataforma Unity.

Resultados

No exercício da nave que coleta estrelas pela sala, foi desenvolvido um script para controlar o movimento da nave e sua interação com as estrelas. O script foi implementado de maneira a permitir que a nave se mova pelo ambiente, coletando as estrelas à medida que passa por elas.

A implementação das técnicas de programação aprendidas, como controle de movimento, detecção de colisões e coleta de objetos, foi fundamental para que a nave navegue pelo espaço, controlada pelo usuário.



Fig. 1: Visão da câmera

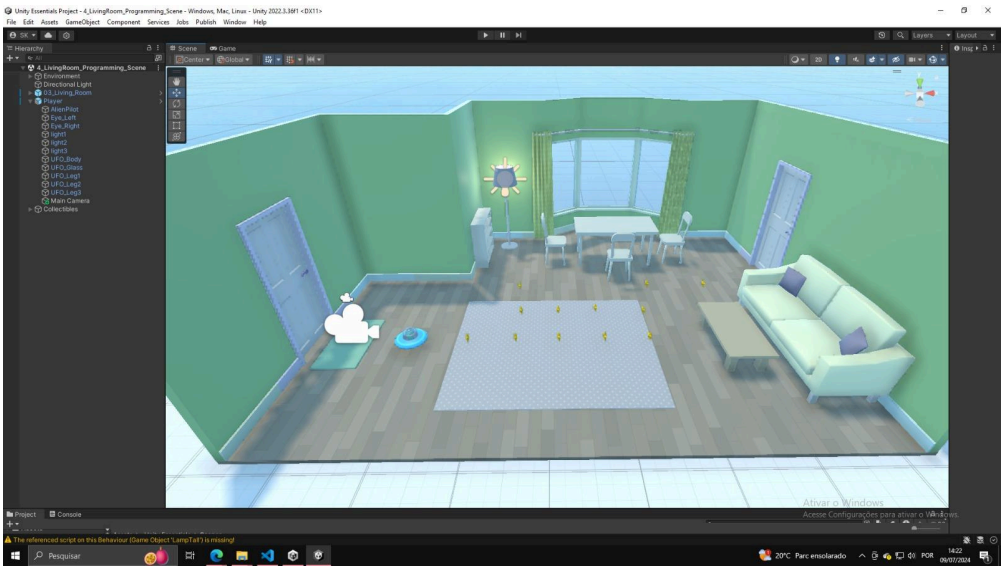


Fig. 2: Visão ampla da sala e dos coletáveis

Conclusões

Durante o estudo, foi possível explorar as capacidades da engine Unity em contextos diversos. Ao realizar os cursos e aplicar os conhecimentos na prática foi essencial para entender o potencial desta plataforma, com suas diversas aplicações e recursos disponíveis. Nas próximas etapas deste projeto, virá a real implementação objeto deste estudo.

Referências bibliográficas

Unity Essentials Pathway. Disponível em: https://learn.unity.com/pathway/unity-essentials

Junior Programmer Pathway. Disponível em: https://learn.unity.com/pathway/junior-programmer

Unity Engine. Disponível em: https://unity.com/pt

Relatório de Atividades

Segmentação Semântica para Geração de Mapas de Cobertura e Uso do Solo para o Território da APA Petrópolis

Bolsista:

Alan Daiki Suga

Período:

01-04-2023 a 30-06-2024

Orientador: Gilson Antonio Giraldi

Co-orientador: Paulo Sergio Silva Rodrigues

Laboratório Nacional de Computação Científica - LNCC/MCTIC

Julho 2024

Resumo

A geração de um mapeamento espacial sobre o uso e a cobertura do solo é fundamental para fornecer subsídios necessários para a tomada de decisões no que diz respeito ao correto manejo dos recursos naturais envolvidos. Cientes dessas questões, os pesquisadores do Instituto Chico Mendes de Conservação da Biodiversidade (ICMBio) e do Laboratório Nacional de Computação Científica (LNCC) vêm desenvolvendo ações integradas para obter um levantamento atualizado das ações humanas referentes à utilização do solo e da distribuição espacial da vegetação da Área de Proteção Ambiental da região da cidade de Petrópolis, a qual compreende uma área de aproximadamente 68.000,00 hectares, no bioma da Mata Atlântica. Neste sentido, este projeto envolve atividades de pesquisa em

segmentação de imagens georreferenciadas extraídas da plataforma Google Earth para segmentação da cobertura e uso do solo da região de interesse. As atividades foram concentradas no estudo da literatura da área, da biblioteca PyTorch para a geração de códigos para o projeto, bem como da rede neural LaNet escolhida para segmentação semântica das imagens.

1. Introdução

Este projeto está inserido no contexto do Acordo de Cooperação número 6/2022, o qual tem por objetivo a cooperação técnica entre o Instituto Chico Mendes de Conservação da Biodiversidade (ICMBio) e o Laboratório Nacional de Computação Científica (LNCC), objetivando a cooperação mútua para que, através de ações integradas na obtenção, processamento, tratamento, classificação, modelagem, sistematização e disponibilização das informações de cobertura e uso do solo, sejam aprimorados os instrumentos de gestão da Área de Proteção Ambiental da região da cidade de Petrópolis (APA-Petrópolis), a qual compreende uma área de aproximadamente 68.000,00 hectares, no bioma da Mata Atlântica, e do território no qual está inserida. Especificamente, a atualização e aprimoramento dos mapas de cobertura e uso do solo da APA Petrópolis é uma demanda importante para o ICMBio porque foi identificado no Mapa de Zoneamento, estabelecido pelo Plano de Manejo da unidade, inúmeras inconsistências na caracterização da cobertura e uso do solo, tendo sido geradas classificações equivocadas, o que permite a utilização de áreas importantes à preservação e protege áreas sem atributos naturais. Estas inconsistências têm gerado dúvidas quanto à aplicação do Plano de Manejo da APA Petrópolis, tendo sido recomendada pelo ICMBio a sua revisão.

Assim, esse projeto envolve ações integradas entre as equipes do ICMBio e LNCC para obter um levantamento atualizado, via imagens aéreas da região, das ações humanas referentes à utilização do solo e da distribuição espacial da vegetação da APA Petrópolis. Nesse contexto, a primeira fase do projeto foi a aquisição de imagens aéreas da região de interesse para seu posterior processamento. Essa etapa foi resolvida com imagens RGB (Red-Green-Blue) extraídas da plataforma Google Earth, por meio da ferramenta ArcGis Pro [1]. Um aluno de mestrado da Universidade Tecnológica Federal do Paraná (UTFPR) trabalhou com essas imagens.

As imagens exportadas têm resolução de 2048 × 2048. Um subconjunto delas foi selecionado manualmente para ser anotado com o objetivo de gerar o padrão-ouro para treinamento de redes neurais que foram utilizadas para a segmentação automática de toda a região da APA Petrópolis, de acordo com as classes de interesse. As imagens selecionadas foram importadas na ferramenta Computer Vision Annotation Tool (CVAT), a qual é uma ferramenta interativa de anotação de vídeo e imagem para visão computacional [2]. As imagens foram então anotadas manualmente por meio da criação de polígonos ou da utilização da ferramenta pincel disponível no CVAT.

Foram testadas pela equipe da UTFPR redes neurais do tipo MPSegnet desenvolvidas pela equipe do projeto [3] bem como a rede U-Net proposta na literatura [4] para a tarefa de segmentação semântica das imagens; ou seja, classificar cada pixel da imagem de entrada de acordo com sua classe na imagem original. No caso, as equipes da UTFPR e do ICMBio descartaram as seguintes classes: Urbano, Floresta, Sombra, Regeneração, Agricultura, Rocha, Solo e Água. Durante o período deste relatório, objetivando melhorar os resultados obtidos apresentados em [5], foram realizadas pesquisas envolvendo:

1. Estudo da biblioteca PyTorch,

2. Treinamento, validação e teste de nova arquitetura de Rede Neural,

3. Ajuste de hiperparâmetros,

4. Aumento de dados.

Cada tema mencionado foi estudado com orientação e auxílio de toda a equipe envolvida. O item (1) foi fundamental para entender a codificação básica de redes neurais. Os itens (2)-(3) foram importantes para compreender o procedimento do treinamento das redes neurais, a importância dos hiperparâmetros e o impacto que a alteração destes pode ter embora se mantenha a arquitetura e o banco de dados.

A troca da arquitetura foi uma das ideias para este projeto, visto que novas redes têm sido publicadas para segmentação de imagens aéreas, usando técnicas mais recentes do que aquelas utilizadas nas redes aplicadas em [5]. Assim, foi escolhida a rede neural LaNet [6], pela sua eficiência relatada na literatura. Foram realizados testes com a rede LaNet, alterando os hiperparâmetros e função de perda, com o objetivo de obter resultados que tornassem as segmentações mais confiáveis para o monitoramento da região em questão.

Outro ponto é o desbalanceamento de classes (item (4)). Por exemplo, em geral, as imagens possuem muito mais regiões com floresta do que com solo exposto. Do ponto de vista do treinamento, isso implica que as redes neurais podem ter dificuldade para distinguir esse último padrão dos demais. Esse tipo de problema foi tratado com a estratégia descrita em [5], formada por aumento de dados e cálculo de pesos para ponderação dos termos da função de perda (loss), no caso, a entropia cruzada ponderada.

Especificamente, foram feitos testes com a LaNet [6], testando como diferentes funções de perda. Os novos resultados mostraram melhora apenas para as classes Urbano, Sombra, Rocha e Solo, com a função de perda entropia cruzada ponderada. Foi testada também a focal loss [7], mas sem sucesso. Mais testes, mudando outros hiperparâmetros do modelo precisam ser realizados para resultados mais conclusivos. Todos os experimentos computacionais foram realizados usando os recursos de GPU (Graphics Processing Unit) do supercomputador Santos Dumont [8].

As próximas seções estão organizadas como segue. Na seção 2 são descritos os principais objetivos do projeto, seguido pela seção 3 onde é explicitada a metodologia de trabalho utilizada. Em seguida, são apresentados os principais resultados (seção 4) e conclusões (seção 5).

2. Objetivos

Considerando o exposto na introdução, o objetivo geral da pesquisa no período considerado foi:

Treinamento de uma nova arquitetura de rede neural para segmentação semântica de imagens

RGB da região da APA Petrópolis. Essa tarefa envolveu os seguintes objetivos específicos:

1. Escolha de nova arquitetura de rede neural considerando a literatura da área

2. Implementar metodologia para aumento de dados direcionada para as classes com menor

número de amostras

3. Ajuste de hiperparâmetros para verificação e comparação dos resultados

3. Metodologia

O bolsista participou de reuniões remotas com a equipe da UTFPR para entender o procedimento para geração do padrão-ouro, o qual passa por duas etapas: extração das imagens aéreas e a anotação manual. Na primeira etapa, as imagens aéreas são extraídas do Google Earth por meio da ferramenta ArcGis Pro [1]. Para isso, uma região do mapa é selecionada e a imagem é exportada em várias partes com tamanhos iguais de 2048 × 2048 pixels em formato RGB (Red-Green-Blue). Dentre as imagens exportadas, algumas delas são selecionadas manualmente para serem anotadas na próxima etapa usando o CVAT. Uma amostra do padrão-ouro pode ser visualizada na Figura 1. Especificamente, a imagem original é mostrada na Figura 1.(a), e os polígonos anotados manualmente contendo as diferentes classes de ocupação do solo, juntamente com regiões de sombra, são visualizados na Figura 1.(b). Finalmente, o padrão-ouro gerado pode ser analisado na Figura 1.(c). Cada cor utilizada corresponde a uma classe de acordo com a legenda da Figura 1. Pode-se notar que na Figura 1 temos uma grande área de agricultura e floresta. As demais classes constituem regiões menores na imagem, o que gera o problema do desbalanceamento comentado anteriormente.



(a) (b) (c)

Figura 1: (a) Imagem original extraída da plataforma Google. (b) Polígonos indicando as diferentes classes observadas. (c) Resultado da segmentação manual realizada onde temos agricultura (em amarelo), floresta em verde, sombra em preto, solo exposto em marrom, área construída em vermelho e piscina em azul.

Para atingir os objetivos da seção 2, a equipe discutiu as possibilidades de melhorar o método de segmentação. Uma das ideias propostas foi a alteração da arquitetura de rede para o LaNet, a qual é uma rede que utiliza esquemas de atenção (Self-Attention). A motivação para essa troca foi a eficiência verificada para esquemas de atenção reportados na literatura [9]. Os testes iniciais usando a LaNet [6] não mostraram resultados melhores do que a arquitetura U-Net utilizada anteriormente. Assim, foram realizados ajustes de hiperparâmetros do treinamento (tamanho de batch, momento, taxa de aprendizagem e weight decay) e mudança para a focal loss, com resultados ainda insatisfatórios. Assim, resolvemos utilizar técnicas de aumento de dados, juntamente com a entropia cruzada ponderada, para mitigar problemas de desbalanceamento de classes, como descrito na próxima seção 4. Todos os experimentos computacionais foram realizados usando os recursos de GPU (Graphics Processing Unit) do supercomputador Santos Dumont [8].

4. Resultados e Discussão

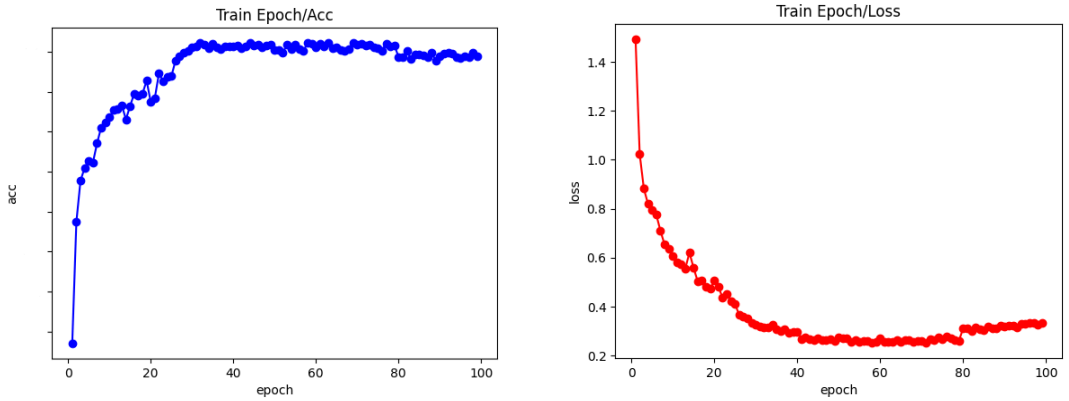
Para efeitos de comparação de resultados, decidimos implementar um protocolo de pré-processamento de dados idêntico ao adotado por [5]. Neste protocolo, primeiramente, o conjunto de imagens originais de tamanho 2048 × 2048 foi dividido em 214 imagens para treinamento e 42 para teste. Recortes (janelas) com tamanho fixo de 256 × 256 pixels são extraídos aleatoriamente do conjunto de treinamento para formar os batches usados durante o treinamento da rede. Essas imagens passam por um processo de aumento de dados usando uma estratégia composta por flip horizontal e flip vertical dos recortes. As intensidades de imagem foram normalizadas dividindo cada canal pela intensidade máxima, que tem valor 255.

Para o treinamento foi usada como função de perda a entropia cruzada ponderada, cujos pesos são calculados usando a técnica explicada no artigo [10]. Foi usado o otimizador SGD (Stochastic Gradient Descent) que utiliza os seguintes parâmetros: taxa de aprendizagem igual a 0.01, momento 0.9, tamanho de batch dado por 64 imagens, weight decay igual a 0.005. Como critério de parada usa-se o número de épocas igual a 100. Como medida de avaliação dos resultados utilizamos matriz de confusão (C) e a acurácia global, calculadas sobre o conjunto de teste. A matriz C para um problema de classificação com N classes, tem seus elementos Ci,jC_{i,j}Ci,j dados pela porcentagem de observações do grupo i que foram classificadas no grupo j, com i,j=1,2,...,Ni, j = 1, 2, ..., Ni,j=1,2,...,N. A acurácia global é dada por:

$$Accuracy = \frac{1}{N} \sum_{i=1}^{N} C_{i,j},$$

onde N é o número de classes.

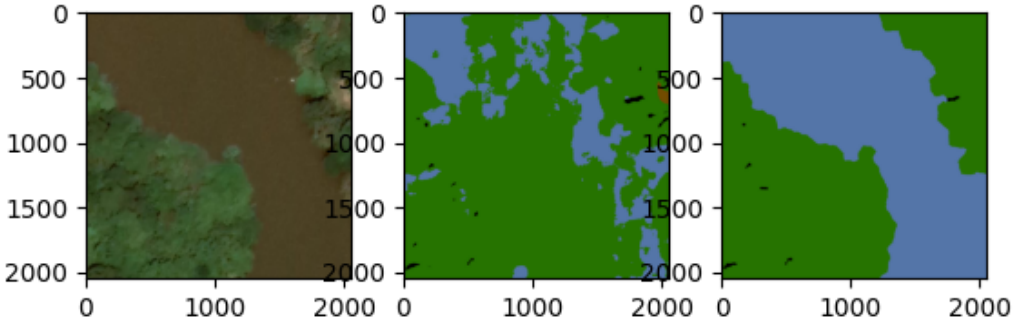
A evolução da acurácia ao longo das épocas de treinamento está apresentada no gráfico da Figura 2.(a). Na Figura 2.(b) é mostrada a evolução da função de perda (loss) ao longo do treinamento. Verificamos que nos dois casos, de forma geral, a eficiência do modelo melhora à medida que avançamos o número de épocas, com um leve decréscimo da acurácia/loss após a época 80.



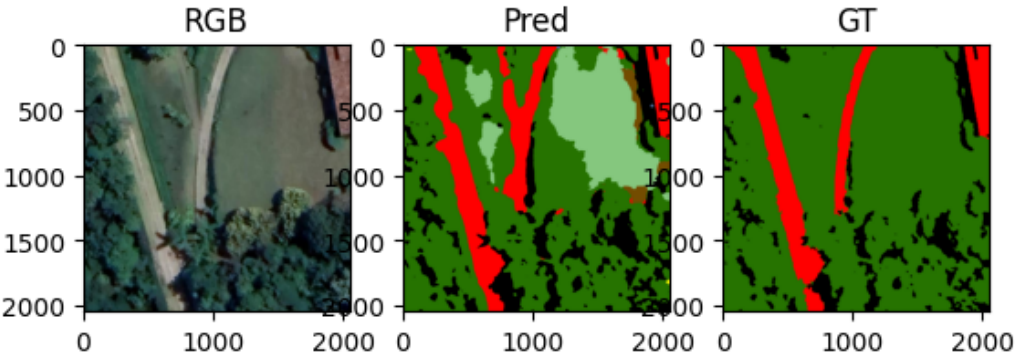
(a) (b)

Figura 2: (a) Mostra a progressão da acurácia ao longo das épocas de treinamento. (b) Mostra a progressão da loss ao longo das épocas de treinamento.

A matriz de confusão, cujo cálculo foi descrito acima, permite uma análise classe-a-classe dos resultados. A Figura 3.(a) mostra a matriz de confusão para a LaNet, a qual pode ser comparada com a matriz de confusão obtida utilizando a rede U-Net (Figura 3.(b)), a qual forneceu os melhores resultados reportados em [5]. É importante observar que a U-Net utilizada em [5] foi treinada usando como função de perda a focal loss. Nos testes que realizamos, os resultados obtidos com essa função de perda foram inferiores aos resultados da LaNet com a entropia cruzada ponderada. Notamos que a mudança da arquitetura gerou resultados que foram satisfatórios em algumas classes, mas em outras não. Especificamente, observamos resultados melhores com a LaNet para as classes Urbano, Sombra, Rocha e Solo, ficando as demais classes com resultados piores em comparação com a U-Net.



(a)



(b)

Figura 4: (a) Exemplo de segmentação com baixa acurácia: Água (em azul).(b) Exemplo de segmentação com alta acurácia: Urbano (em vermelho).

5 Conclusão

O acesso ao supercomputador SDumont possibilitou a realização dos testes, que não seriam viáveis em computadores convencionais. Os resultados obtidos com a rede LaNet mostraram-se promissores, embora não tenham superado completamente os resultados da rede U-Net.

Na continuidade deste trabalho, serão utilizadas estratégias automáticas de otimização de hiperparâmetros, com o objetivo de melhorar os resultados da LaNet e aumentar a confiabilidade da rede para a aplicação de interesse.

Referências

[1] **ArcGisPro**, Software para Geoprocessamento. [Online]. Disponível em: https://www.esri.com/pt-br/arcgis/products/arcgis-pro/overview. Acesso em: 19 jul. 2024.

[2] **CVAT**, Computer Vision Annotation Tool. [Online]. Disponível em: https://www.cvat.ai/. Acesso em: 19 jul. 2024.

[3] A. de Souza Brito, M. B. Vieira, M. L. S. C. de Andrade, R. Q. Feitosa, e G. A. Giraldi, "Combining max-pooling and wavelet pooling strategies for semantic image segmentation," *Expert Systems with Applications*, vol. 183, p. 115403, 2021. [Online]. Disponível em: https://www.sciencedirect.com/science/article/pii/S0957417421008253. Acesso em: 19 jul. 2024.

[4] O. Ronneberger, P. Fischer, e T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, ser. LNCS, vol. 9351. Springer, 2015, pp. 234–241.

[5] F. Bizotto, "Convolutional neural networks for semantic segmentation of aerial images in land cover mapping of environmental protection areas," *Master's thesis*, Universidade Tecnológica Federal do Paraná, Petropolis-RJ, 11 2023, advisor: Gilson A. Giraldi. [Online]. Disponível em: http://repositorio.utfpr.edu.br/jspui/handle/1/33330. Acesso em: 19 jul. 2024.

[6] L. Ding, "LANet: Local Attention Embedding to Improve the Semantic Segmentation of Remote Sensing Images," 2021. [Online]. Disponível em: https://ieeexplore.ieee.org/document/9102424. Acesso em: 19 jul. 2024.

[7] Y. Zhang, R. Togneri, e D. Huang, "A unified loss function to tackle inter-class and intra-class data imbalance in sound event detection," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 996–1000.

[8] **SDumont**, Supercomputador Santos Dumont. [Online]. Disponível em: https://sdumont.lncc.br/. Acesso em: 19 jul. 2024.

[9] M. Hassanin, S. Anwar, I. Radwan, F. S. Khan, e A. Mian, “Visual attention methods in deep learning: An in-depth survey,” *Information Fusion*, vol. 108, p. 102417, 2024.

[10] P. O. Bressan, J. M. Junior, J. A. Correa Martins, M. J. de Melo, D. N. Gonçalves, D. M. Freitas, A. P. Marques Ramos, M. T. Garcia Furuya, L. P. Osco, J. de Andrade Silva, Z. Luo, R. C. Garcia, L. Ma, J. Li, e W. N. Gonçalves, “Semantic segmentation with labeling uncertainty and class imbalance applied to vegetation mapping,” *International Journal of Applied Earth Observation and Geoinformation*, vol. 108, p. 102690, 2022. [Online]. Disponível em: https://www.sciencedirect.com/science/article/pii/S0303243422000162. Acesso em: 19 jul. 2024.

RELATÓRIO DE PROJETO DE INICIAÇÃO CIENTÍFICA

BIOINFORMÁTICA, BANCO DE DADOS E ENGENHARIA DE COMPUTAÇÃO

Título do projeto: Comparação do Desempenho Computacional de *Workflows* Científicos de transcriptômica entre Arquiteturas de Memória Distribuída do Supercomputador Santos Dumont

Nome do bolsista: Albert Siqueira Cosme Emidio

Nome do orientador:

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

D.Sc. Carla Osthoff Ferreira de Barros (Tecnologista Sênior – SEPAD/COTIC//LNCC, Coorientador)

Tipo de bolsa: PIBIC

Período do relatório: 28/06/2023 - 28/06/2024

1. Objetivo

O objetivo deste estudo é realizar uma análise computacional comparativa das arquiteturas Cascade Lake Gold (Sequana) e Ivy Bridge (Nvidia) do supercomputador Santos Dumont (SDumont) a respeito do uso da tecnologia de memória distribuída usando como caso de estudo aplicações de bioinformática, mais especificamente o desempenho do *workflow* de transcriptômica ParslRNA-Seq.

2. Introdução

Workflows científicos de bioinformática são complexos devido à necessidade de adaptação a casos reais em diversas áreas, especialmente quando se lida com dados volumosos, heterogêneos e distribuídos em diversos locais. Esses *workflows* são mais eficazes quando suportados por tecnologias especializadas de Sistemas de Gerenciamento de *Workflows* Científicos (SGWfC) e Computação de Alto Desempenho (CAD). Além desses fatores, para se projetar um *workflow* exige uma compreensão profunda do problema biológico a ser resolvido, já que a natureza dos dados direciona a modelagem e arquitetura computacional mais adequada.

O workflow de transcriptômica ParslRNA-Seq para análise de expressão diferencial dos genes foi utilizado como estudo de caso do projeto para análise de desempenho. Desenvolvido para lidar com um grande volume de dados, ele foi modelado e gerenciado em linguagem de programação python utilizando bibliotecas Parsl em ambiente de Computação de Alto Desempenho (CAD). As atividade do ParslRNA-Seq incluem Bowtie2 para mapear leituras, Sort para ordenar, Split para dividir os arquivos, HTSeq para contar as leituras, Merge para indexar as contagens e Deseq para análise de estatística de EDG. O Bowtie2 é a atividade mais custosa em questão computacional, ela pode ser executada em paralelo, assim como a atividade Sort (SAMTools) permitindo o aumento da eficiência e escalabilidade em ambientes de CAD.

As análises e comparação estão sendo realizadas pelas arquiteturas Cascade lake com sistema de clusters de memórias distribuídas, cada nó com 2 CPUs Intel Xeon Cascade Lake Gold 6252 (48 núcleos) e com 384 GB de RAM. Ivy Bridge também apresenta a tecnologia de memória distribuída, cada nó com 2 CPUs Intel Xeon E5-2695v2 (24 núcleos) e com 64 GB de RAM. As arquiteturas distribuem as tarefas através de uma rede Infiniband EDR.

3. *Background*

O Parsl-RNASeq (Silva et a 2021)[1] é um *workflow* científico de transcriptoma para estudos de expressão diferencial de genes. Ele foi modelado e acoplado no ambiente do supercomputador SDumont. A Figura 1 apresenta o modelo conceitual do ParslRNA-Seq. As análises de EDG são de extrema importância na análise de dados transcriptômicos, permitindo a identificação do gene com expressão significativamente diferentes entre as condições experimentais. Essa abordagem é muito usada para descobertas de novos transcritos e isoformas de genes, proporcionando estudos sobre esses processos biológicos complexos, como diferenciação celular, resposta a estímulos e desenvolvimento de doenças.

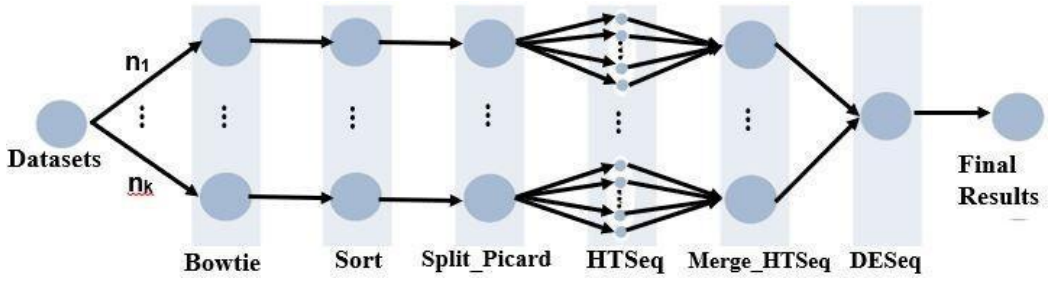


Figura 1. Modelagem Conceitual do *Workflow* Científico ParslRNA-Seq(6 atividades).

Workflow ParslRNA-Seq é composto por seis atividades e utiliza como entrada o genoma de referência do *Mus musculus*, arquivos GTF contendo metadados genômicos e sequenciamentos em formato FASTQ. Um arquivo CSV relaciona os arquivos FASTQ às condições experimentais, sendo três de controle e três da condição Wnt (relacionada à via de sinalização de transcrição Wnt).

O ParslRNA-Seq é implementado utilizando o Parsl, uma ferramenta em Python desenvolvida para executar workflows. O Pars[2] permite a integração e comandos externos diretamente no código Python por meio de decoradores (Apps), isso facilita a sincronização das atividades. O *workflow* utiliza dois modelos de execução: ThreadPoolExecutor e HTEx. O ThreadPoolExecutor[3] é um modelo que suporta *multithreading* em recursos locais gerenciando em *pool* de *threads* para executar as atividades de forma corrente. Já o modelo HTEx permite a execução em múltiplos nós computacionais simultaneamente.

4. Metodologia

4.1 Dados do experimento

Os dados pertencem a um experimento real de RNA-Seq, extraídos do repositório público *Gene Expression Omnibus*[4] (GEO) e divididos em: (1) grupo de controle: SRR5445794, SRR5445795, SRR5445796 e (2) grupo de condições das vias Wnt: SRR5445797, SRR5445798, SRR5445799. O organismo é *Mus musculus* e o GEO.ID é GSE97763 (Plataforma Illumina HiSeq 2000 - *Mus musculus*).

[1] https://doi.org/10.5753/bresci.2021.15789
[2] https://parsl-project.org/
[3] https://docs.python.org/3/library/concurrent.futures.html.
[4] https://www.ncbi.nlm.nih.gov/geo/

As leituras de sequência foram alinhadas ao genoma de referência do rato (UCSC versão mm9) com Bowtie2. Para cada gene, o número de leituras mapeadas foi contado com Hhtseqcount. DESeq2 analisou a EDG a partir das matrizes das contagens do alinhamento e do mapeamento das sequências frente ao genoma de referência. Essas matrizes (arquivo GTF) contém o número de leituras que foram alinhadas de forma única (colunas) com os exames de cada gene nas amostras (colunas). Os seis arquivos de entrada totalizam 13GB e geram arquivo de saída de aproximadamente 74 GB.

4.2 Ambiente Computacional

O SDumont[5] 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. Todos os *software*, algoritmos, dependências de bioinformática (Bowtie, Samtools, Picard, HTSeq e DESeq2) e os componentes do Parsl foram alocados e instalados no ambiente do SDumont.

No presente experimento foi usado o nó 2 CPUs Intel Xeon Cascade Lake Gold 6252 (48 núcleos) e com 384 GB de RAM. As execuções do experimento original nas publicações de Silva et a 2021[13] foram realizadas em nós computacional de duas CPUs Intel Xeon E5-2695v2 Ivy Bridge, 24 núcleos (12 por CPU) de 64 GB de memória RAM. Todos os demais parâmetros do experimento se mantiveram iguais ao experimento original.

4.3 Análise do uso da cache nos processadores

Devido à natureza não inclusiva da LLC (cache de último nível), a ausência de uma linha de cache na LLC não significa que ela não esteja nos caches privados dos núcleos. Por isso, a arquitetura usa um filtro de *snoop* para buscar no cache L2 ou cache de nível médio (MLC) dos núcleos quando a LLC não contém os dados necessários. Antes da família de CPUs Intel Xeon Cascade Lake, a LLC compartilhada resolvia essa tarefa. A mudança para uma LLC menor e um cache L2 maior reduz a latência nas operações de leitura/escrita.

Nas arquiteturas anteriores ao Cascade Lake, todas as linhas no cache L2 de cada núcleo eram armazenadas na LLC inclusiva e compartilhada. No Cascade Lake, cada núcleo tem um cache L2 privado de 1 MB, e todos os núcleos compartilham uma LLC maior que pode atingir 38,5 MB. Nem todas as linhas no cache L2 de cada núcleo estão na LLC não inclusiva e compartilhada. Um cache L2 maior aumenta a taxa de acertos do cache L2, reduzindo a latência de memória e a demanda na LLC.

4.4 Tempo total de execução do *workflow* utilizando 1 nó (Ivy Bridge) X 1 nó (Cascade Lake)

A reexecução do *workflow* de Silva et al. (2021)[6] foi realizada em um único nó com 2 CPUs Intel Xeon Ivy Bridge de 2.4GHz, 64 GB de RAM e 24 núcleos. Na presente pesquisa, o experimento foi

[5] https://sdumont.lncc.br/
[6] https://doi.org/10.5753/bresci.2021.15789

conduzido utilizando 2 CPUs Intel Xeon Cascade Lake Gold 6252 com 48 núcleos e 384 GB de RAM. Foram testadas configurações de 1, 2, 4, 6, 12, 18 e 24 threads, semelhantes ao experimento original, para avaliar o desempenho computacional no Cascade Lake. O modelo ThreadPoolExecutor do Parsl foi utilizado, permitindo fazer *multithreading* em recursos locais.

4.5 Tempo total de execução do *workflow* utilizando 6 nós (Ivy Bridge) X 6 nós (Cascade Lake)

No experimento, usamos 6 nós do Intel Xeon de 2,4 GHz, cada um com 64 GB de RAM e 24 núcleos, executando 1 arquivo por nó através do modelo HTEx do Parsl que permite a execução do *workflow* em múltiplos nós computacionais simultaneamente. A mesma configuração foi realizada com o Intel Xeon Cascade Lake Gold 6252, cada um com 384 GB de RAM e 48 núcleos. A escolha de 6 nós dos processadores foi feita para validar a capacidade da tecnologia de memória distribuída em uma arquitetura mais antiga (Ivy Bridge) e com uma mais nova (Cascade Lake), permitindo uma comparação justa do desempenho com diferentes quantidades de threads: 1, 2, 4, 6, 12, 18, 24, 48. Este arranjo nos permitiu avaliar como cada sistema gerencia a carga de trabalho em cenários variados, desde um baixo número de *threads* até a capacidade máxima dos núcleos disponíveis.

5. Resultados e Discussão

5.1 Resultados de desempenho do *workflow* 1 nó

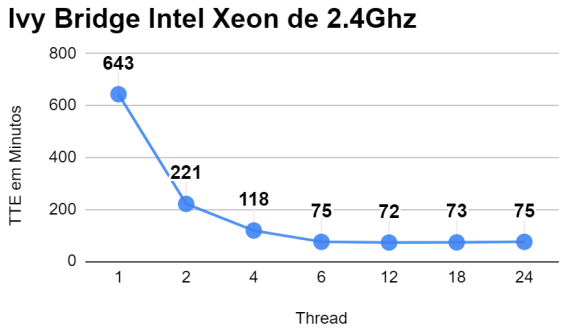


Figura 2 TTE do *workflow* em um nó Nvidia

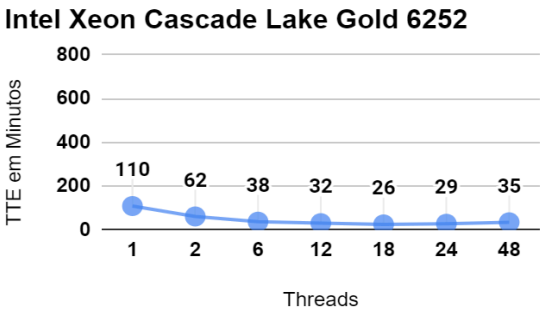


Figura 3 TTE *workflow* em um nó Cascade Lake

O gráfico 1 exibe o tempo total de execução do *workflow* em um nó Ivy Bridge com 24 threads (Silva et al., 2021), enquanto o Gráfico 2 mostra a execução no Cascade Lake de 48 *threads*. A comparação inicial é feita com 1 e 24 *threads* em ambos os casos. Observamos que o Cascade Lake superou oIvy Bridge, com uma melhoria de 83% na comparação com apenas 1 *thread* e uma redução de 61% no tempo com 24 *threads*. Além disso, o Cascade Lake alcançou seu melhor desempenho utilizando 18 *threads*, com um tempo total de execução de 26 minutos. Após esse ponto, houve um aumento no tempo devido à falta de trabalho disponível para os *threads* adicionais.

5.2 Resultados desempenho do *workflow* em múltiplos nós

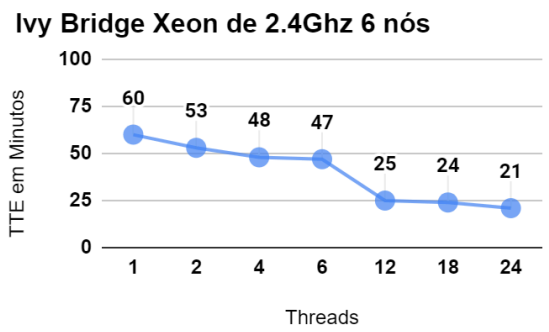


Figura 4 TTE do *workflow* em 6 nós Ivy Bridge

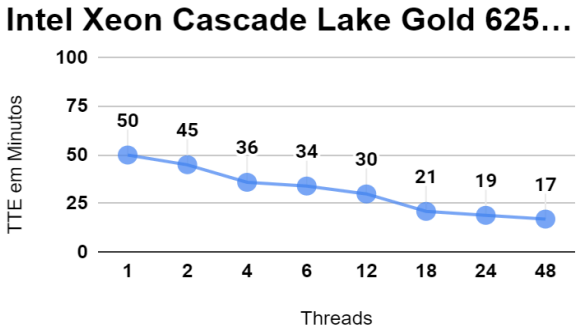


Figura 5 TTE *workflow* em 6 nós Cascade Lake

Os resultados evidenciam a superioridade do Cascade Lake Figura 5, em relação ao Ivy Bridge Figura 4, nas diferentes configurações de threads avaliadas. Com apenas 1 *thread*, Cascade Lake concluiu o workflow em 50 minutos, comparado aos 60 minutos do Ivy Bridge. A diferença se acentua com 24 *threads*, onde o Cascade Lake executou o *workflow* em 19 minutos, contra 21 minutos do Ivy Bridge. O Cascade Lake alcançou seu melhor desempenho com 48 *threads*, completando o processo em 17 minutos.

Conclusões

Os resultados obtidos nos experimentos destacam claramente a superioridade do Cascade Lake em relação ao Ivy Bridge para a execução do *workflow* de transcriptômica. O Cascade Lake apresentou uma notável redução no tempo de execução comparado ao Ivy Bridge, especialmente evidente ao usar 18 threads, onde alcançou seu melhor desempenho. Portanto, o Cascade Lake, como uma arquitetura mais recente e poderosa, se sobressai.

O emprego de memória distribuída nas arquiteturas testadas ressalta a importância do paralelismo para a execução otimizada deste *workflow*. Ao adotar um nó para cada arquivo de entrada, o workflow ganha em desempenho e redução de tempo.

As variações de desempenho observadas entre os dois sistemas reforçam a necessidade de escolher a arquitetura de hardware adequada para aplicações que exigem alto desempenho e intenso paralelismo. Considerando isso, os próximos passos incluem a continuidade da otimização do workflow de transcriptômica para diferentes ambientes computacionais, visando melhorar ainda mais a eficiência e a escalabilidade do processamento de dados genômicos.

Esses esforços beneficiarão não só pesquisadores e cientistas na análise de grandes conjuntos de dados biológicos, mas também contribuirão para o avanço da bioinformática em instituições como o LNCC e outras colaboradoras deste projeto.

Referências bibliográficas

[1] V. Marx, “Biology: The big challenges of big data,” Nature, vol. 498, no. 7453, pp. 255–260, Jun. 2013.

[2] J. Freire, D. Koop, and L. Moreau, Eds., Provenance and Annotation of Data and Processes, vol. 5272. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[3] M. Mattoso et al., “Towards supporting the life cycle of large scale scientific experiments,” International Journal of Business Process Integration and Management, vol. 5, no. 1, pp. 79–92, 2010.

[4] T. J. Hastie, R. J. Tibshirani, and J. H. Friedman, The elements of statistical learning: data mining, inference, and prediction, 2. ed., At 7. printing. New York, NY: Springer, 2013.

[5] G. Da San Martino and A. Sperduti, “Mining Structured Data,” IEEE Computational Intelligence Magazine, vol. 5, no. 1, pp. 42–49, Feb. 2010.

[6] L. Hirschman, J. C. Park, J. Tsujii, L. Wong, and C. H. Wu, “Accomplishments and challenges in literature data mining for biology,” Bioinformatics, vol. 18, no. 12, pp. 1553–1561, Dec. 2002.

[7] Ocaña, K. A. C. S.; Galheigo, M.; Osthoff, C.; Gadelha, L. M. R.; Porto, F.; Gomes, A. T. A.; Oliveira, D.; Vasconcelos, A. T. BioinfoPortal: A scientific gateway for integrating bioinformatics applications on the Brazilian national high-performance computing network. Future Generation Computer Systems, v. 107, p. 192-214, 2020.

[8] M. Wilde, M. Hategan, J. M. Wozniak, B. Clifford, D. S. Katz, and I. Foster, “Swift: A language for distributed parallel scripting,” Parallel Computing, no. 37(9), pp. 633–652, 2011.

[9] L. M. R. Gadelha, M. Wilde, M. Mattoso, and I. Foster, “MTCProv: a practical provenance query framework for many-task scientific computing,” Distrib Parallel Databases, vol. 30, no. 5–6, pp. 351–370, 2012.

[10] Cruz, L., Coelho, M., Terra, R., Carvalho, D., Gadelha, L., Osthoff, C., & Ocaña, K. 2021 jul 18. Workflows Científicos de RNA-Seq em Ambientes Distribuídos de Alto Desempenho: Otimização de Desempenho e Análises de Dados de Expressão Diferencial de Genes. Anais do Brazilian e-Science Workshop (BreSci). [https://doi.org/10.5753/bresci.2021.15789.]

[11] Taipe, S.H.H. *et al.* (2023). An Evaluation of Direct and Indirect Memory Accesses in Fluid Flow Simulator. In: Gervasi, O., *et al.* Computational Science and Its Applications – ICCSA 2023. ICCSA 2023. Lecture Notes in Computer Science, vol 13956 . Springer, Cham. https://doi.org/10.1007/978-3-031-36805-9_3

RELATÓRIO DE ATIVIDADES

1) Dados gerais

Título do projeto:
Estudo e implementação de mecanismo de aceleração de desempenho de experimentos científicos e simulações através de Inteligência Artificial

Nome do bolsista:
Alexandre de Paiva Almeida

Nome do orientador:
Jauvane de Oliveira Cavalvante, PhD

Tipo de bolsa e período do relatório:
PIBITI – De 30 MAIO 2023 até 30 SET 2024

2) Objetivos

Este projeto tem por objetivo o estudo, desenvolvimento e utilização da técnica de engenharia de software, ciência de dados e machine learining para identificar padrões de bugs de software relacionados à performance e utilizar esses resultados para acelerar a execução de experimentos científicos e simulações em vários tipos de projetos de desenvolvimento, inclusive os relacionados à natureza militar e de bioinformática.

3) Introdução

A aceleração da execução de experimentos científicos e simulações desempenha um papel fundamental no avanço da pesquisa em diversas áreas, incluindo a natureza militar e a bioinformática. No entanto, essas atividades computacionais muitas vezes enfrentam desafios relacionados à performance, resultando em processamentos lentos e ineficientes. A busca por soluções eficazes e economicamente viáveis para melhorar a velocidade de execução desses experimentos tem se mostrado uma tarefa complexa.

Neste contexto, este projeto de pesquisa tem como objetivo principal o estudo, desenvolvimento e utilização de técnicas de engenharia de software, ciência de dados e soluções envolvendo machine learning para identificar padrões de bugs de software que afetam o desempenho e melhorar sua performance.

A identificação desses padrões permite uma abordagem mais eficiente na resolução dos gargalos de performance, possibilitando a escolha adequada dos recursos e estratégias necessários para otimizar o processamento e a simulação de experimentos, bem como o treinamento de uma LLM para classificação de experimentos até então desconhecidos.

Nesse contexto, a técnica de engenharia de software e ciência de dados combina princípios da engenharia de software com análise de dados e aprendizado de máquina. Essa abordagem multidisciplinar busca resolver problemas complexos de desenvolvimento e otimização de software, garantindo qualidade e eficiência. Ela utiliza ferramentas da engenharia de software para escalabilidade e técnicas de ciência de dados para extrair insights de grandes volumes de dados. Essa combinação é fundamental para identificar padrões, detectar bugs e melhorar a performance dos sistemas de computação.

A pesquisa proposta aborda a utilização da técnica de Mineração de Repositórios de Software (MSR) e Aprendizado de Máquina (IA) como meio de identificar os padrões de bugs relacionados à performance. Ao compreender a causa raiz dos problemas de performance, será possível direcionar os esforços de forma mais precisa e evitar soluções genéricas e onerosas, que podem não trazer os resultados desejados.

Além disso, este projeto também busca a formação de recursos humanos nas áreas de Mineração de Repositórios de Software, Aprendizado de Máquina e análise de gargalos de performance em simulações, softwares para ciência de dados e experimentos científicos.

Pretende-se desenvolver uma metodologia e automação de scripts para a descoberta dos gargalos de performance, visando reduzir o tempo de processamento/simulação em pelo menos 10 vezes.

Espera-se que os resultados obtidos nessa pesquisa contribuam para a otimização da execução de experimentos científicos e simulações, possibilitando avanços significativos em diversas áreas do conhecimento. Os resultados também serão documentados em um relatório científico no formato de artigo acadêmico, com potencial para submissão em periódicos indexados, considerando a qualidade do trabalho e os resultados alcançados.

4) Material e Métodos ou Metodologia

Simulações, softwares para ciência de dados e experimentos científicos in sílico tem um processamento computacional muito lento, dadas as complexidades do mundo real, cuja verossimilhança é buscada ao máximo em tais sistemas. Para resolver estes problemas, os cientistas e centros de pesquisa lançam mão de recursos como supercomputadores, processamento via placas gráficas de alta performance (GPU), paralelismo de software, aumento de escala com computação em nuvem e otimizações computacionais nos programas. Porém, nem sempre tais soluções são eficientes ou justificam seu emprego, já que podem ser caras tanto do ponto de vista da compra dos recursos propriamente dito, quanto do ponto de vista da quantidade de horas necessárias para ajustar os experimentos e simulações a tais ambientes.

Diante deste dilema, uma das estratégias para empregar os recursos de forma eficiente e responsável é o estudo da causa raiz do gargalo de performance, o que pode ser feito através da implementação de probes e profiling ou através da

identificação de padrões de bugs relacionados à performance. A pesquisa relacionada a este projeto tem por finalidade a utilização da técnica de Mineração de Repositórios de Software (MSR) e Aprendizado de Máquina (IA) para identificar padrões de bugs relacionados à performance. Uma vez identificado o padrão, o cientista pode atacar de forma mais eficiente os gargalos relacionados ao seu contexto, empregando o recurso mais apropriado para solucionar o problema da performance em vez de uma solução “força bruta” e que nem sempre trará o resultado desejado ou terá um custo muito alto para um benefício baixo.

5) Resultados e Discussão

Nesse período de trabalho os resultados foram:
1. Formação de recursos humanos em mineração de repositórios de software (MSR), Aprendizado de Máquina (IA) e análise de gargalos de performance em simulações, softwares para ciência de dados e experimentos científicos.
2. Levantamento de projetos e funções para que as técnicas desenvolvidas, como o método de serialização e interpretador, pudessem ser aplicados com o objetivo de reduzir o tempo de processamento/simulação;
3. Definição de metodologia e automação de scripts para descoberta de gargalos de performance
4. Análise dos resultados do tempo de execução dos experimentos
5. Estudo do código de cada experimento para levantamento de padrões a serem tomados como parâmetros para o modelo de máquina
6. Escrita de relatório de atividades

Os resultados apurados a partir da análise inicial mostraram relevante a automatização do procedimento para garantir que os experimentos sejam executados exatamente da mesma maneira e sob as mesmas condições para que a comparação seja precisa e justa.

Para isso, foi desenvolvido um notebook Colab que comparará todas as versões do projeto com todos os métodos de serialização implementados (pickle, picklejson, simplejson e hdf5) bem como padrões dos códigos. Os padrões observados como mais relevantes poderão agora ser considerados para o treinamento de uma LLM capaz de classificar experimentos de maneira mais ágil.

6) Conclusões

O projeto de pesquisa se mostra cada vez mais relevante no cenário atual e o progresso nesse mês de trabalho mostra que é possível fazer as análises esperadas

7) Referências bibliográficas

[1] SELAKOVIC, Marija; PRADEL, Michael. Performance issues and optimizations in
javascript: an empirical study. In: Proceedings of the 38th International Conference
on Software Engineering. 2016. p. 61-72.
[2] HU, Jingmei et al. Improving data scientist efficiency with provenance. In: 2020
IEEE/ACM 42nd International Conference on Software Engineering (ICSE). IEEE,
2020. p. 1086-1097.
[3] PIMENTEL, João Felipe et al. A survey on collecting, managing, and analyzing
provenance from scripts. ACM Computing Surveys (CSUR), v. 52, n. 3, p. 1-38, 2019.
[4] MORENO, Alexander; BALCH, Tucker. Speeding up large-scale financial
recomputation with memoization. In: 2014 Seventh Workshop on High Performance
Computational Finance. IEEE, 2014. p. 17-22.
[5] GUO, Philip J.; ENGLER, Dawson. Using automatic persistent memoization to
facilitate data analysis scripting. In: Proceedings of the 2011 International Symposium
on Software Testing and Analysis. 2011. p. 287-297.

1) Dados Gerais:

- Alexandre Vitor Silva Braga (alexandre.braga@estudante.ufjf.br)

  Estudante graduação, Engenharia Computacional, Universidade Federal de Juiz de Fora
- Elson M. Toledo (emtc@lncc.br)

  Departamento de Mecânica Aplicada Computacional, Universidade Federal de Juiz de Fora
- Marcos V. Rodrigues (marcos.rodrigues@engenharia.ufjf.br)

  Departamento de Engenharia de Produção, Universidade Federal de Juiz de Fora
- Eduardo L. M. Garcia (bidu@lncc.br)

  Departamento de Modelagem Computacional, Laboratório Nacional de Computação Científica

2) Objetivos:

Neste trabalho estudamos os diferentes modos de ruptura de um tubo flexível quando submetidos a cargas offshore, incluindo as fases de fabricação, instalação ou operação, visando aprofundar o conhecimento do seu comportamento. O foco do trabalho está na camada de armadura de tração, o grande desafio é obter o correto entendimento de seu comportamento estrutural considerando a propriedade de deslizamento da armadura sobre outras camadas quando submetida a cargas de flexão. A busca pela correta modelagem e análise numérica desta camada foi o tema e objetivo principal deste estudo.

3) Introdução:

Dentre os desafios da indústria offshore brasileira está encontrar soluções para o escoamento de petróleo e gás que atendam tecnicamente e sejam comercialmente viáveis. Especialmente no caso da região do Pré-Sal, os ganhos em termos de projeto, otimização, desenvolvimento de novas soluções e melhor conhecimento da fase operacional resultam em impactos altamente positivos para as indústrias deste setor O transporte de petróleo e gás natural, provenientes de grandes profundidades offshore, normalmente é realizada por meio de dutos, chamados de risers e linhas de fluxo, que conectam os poços à plataforma flutuante. Esses dutos podem ser feitos de aço (dutos rígidos) ou camadas intercaladas de fios de aço e camadas de plástico (tubos flexíveis). Em tubos flexíveis, cada camada individual tem uma finalidade específica (como suportar pressão interna e cargas de tração), além de resultar na flexibilidade e capacidade dos dutos de absorver movimentos e vibrações de meio ao qual estão submetidos. Este estudo visou aprofundar o conhecimento dos diferentes modos de falha da camada de reforço de aço elástico, também conhecida como camada de armadura de risers flexíveis. Um comportamento que tem sido fundamental para esses dutos flexíveis instalados em grandes profundidades. Alguns estudos metodológicos foram revisados para entender seu comportamento estrutural no decorrer deste trabalho, dentre eles destacamos Mendes de Souza

[1], Leroy e Estrier [2] e Toledo et al. [3]. Este trabalho ao final tentou reproduzir a metodologia do último mencionado e gerar novos resultados.

4) Metodologia:

Inicialmente foi feita uma revisão literária de trabalhos relacionados por parte do aluno, incluindo principalmente os trabalhos de Mendes de Souza [1], Leroy e Estrier [2] e Toledo et al. [3]. Durante e após a etapa de revisão literária foram gerados resumos contendo os estudos do aluno. Em seguida, buscou-se replicar a metodologia aplicada por Toledo et al. [3] e gerar novos resultados a partir desta, portanto focando o estudo sobre sua modelagem.

Hipótese do Modelo

A hipótese principal para esse estudo definiu que risers flexíveis com armaduras de tração de movimento livre, sujeitos a cargas de flexão, irão alterar a configuração de sua posição final entre uma loxodrômica e uma geodésica, sem variação em seu comprimento. Se as armaduras fossem fundidas ao tubo, não haveria deslocamentos entre elas, e o atrito tenderia a um valor máximo, tornando a geometria final em conformidade com uma loxodrômica. Por outro lado, se as armaduras fossem completamente livres, assumindo a ausência de atrito entre elas, sua configuração final tenderia a percorrer uma geodésica. Ambas curvas pertencem a uma mesma família e podem ser definidas por um parâmetro ***Z***, o qual depende do ângulo de assentamento das armaduras, e sendo este definido como $0 \leq Z \leq \overline{Z}$. Onde:

$Z = 0$ na loxodrômica

$Z = \overline{Z} = -\left(2 + \frac{1}{\tan^2 \alpha}\right)$ na geodésica

Definimos também para este modelo que as armaduras atuam como curvas helicoidais sobre o eixo do tubo cilíndrico, sendo ***alfa*** o ângulo de assentamento entre tubo e armaduras, ***L*** o comprimento do tubo de um ponto ***M*** em relação à origem que será deslocado, e ***s*** o comprimento da armadura helicoidal do ponto ***M***, como ilustram as Figura 1 e 2 a seguir:

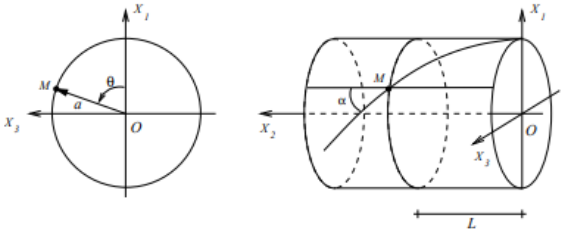


Figura 1. Coordenadas Helicoidais

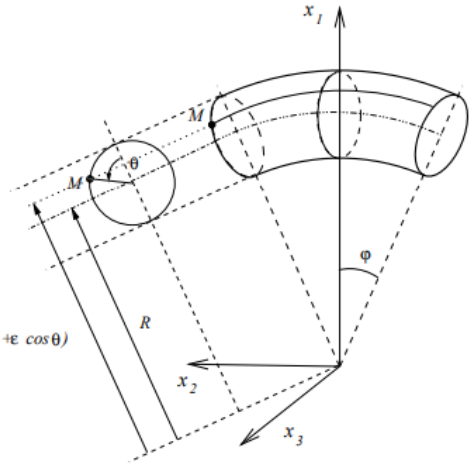


Figura 2. Coordenadas Torus

$$g = \frac{d\phi}{d\theta}$$

Utilizando relações conhecidas do triedro de Frenet-Serret, além de relações definidas por Feret et al. [4], encontrou-se equações generalizadas para a curvatura transversal, normal e para a torção para um ponto ***M*** encontrando-se sobre uma curva ***S(t)***.

$$\begin{cases} C_n = \dfrac{d\vec{t}}{ds} \cdot \vec{N} = \left[a + Rvg^2 \cos\theta\right] \left(\dfrac{d\theta}{ds}\right)^2 \\ C_b = \dfrac{dt}{ds} \cdot \vec{B} = \left[\sin\theta(2a^2 g + R^2 v^2 g^3) - aRv\dfrac{dg}{d\theta}\right] \left(\dfrac{d\theta}{ds}\right)^3 \\ \dfrac{1}{\tau} = -\dfrac{d\vec{N}}{ds} \cdot \vec{B} = Rg\left(\dfrac{d\theta}{ds}\right)^2 \end{cases}$$

Significando que, possuindo o valor do parâmetro ***Z*** que define as curvas entre loxodrômica e geodésica, além da relação angular ***g*** é possível calcular os deslocamentos da posição de equilíbrio de uma curva qualquer com atrito.

Equações Governantes

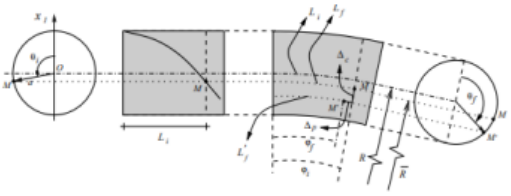
De acordo com a hipótese, dado um parâmetro ***Z*** e relação angular ***g*** é possível obter os deslocamentos de qualquer curva, porém um problema que surge é como encontrar esse parâmetro.

Sabe-se que a diferença de curvaturas nas direções transversal (***B***) e normal (***N***) pode ser dada pela diferença entre esses valores na geodésica e loxodrômica em função de ***Z***, o que resulta na seguinte equação envolvendo ***DeltaC_n*** e ***DeltaC_b***:

$$\begin{cases} \Delta C_N = C_{nG} - C_{nL} = \frac{\epsilon \cos\alpha}{a} \cos\alpha \cos\theta \left(\cos 2\alpha - 2z \operatorname{sen}^2 \alpha\right) \\ \Delta C_B = C_{bL} = \frac{\epsilon \cos\alpha}{a} \{1 + \sin^2\alpha\} \sin\theta - \frac{\sin^2\alpha}{a^2} \Delta_b \end{cases}$$

Utilizando as relações geométricas que definem os comprimentos ***L*** e ***s***, é possível encontrar os deslizamentos paralelo e circunferencial das armaduras como ilustrado pela figura 3. Aplicando uma decomposição nas direções tangencial ***Delta_t*** e transversal ***Delta_b*** como na figura 4, resulta na seguinte equação em função de ***Z***.

$$\begin{cases} \Delta_t = -a\varepsilon \frac{\cos^2\alpha}{\sin\alpha} \sin\theta \\ \Delta_b = a\varepsilon z \cos\alpha \sin\theta \end{cases}$$



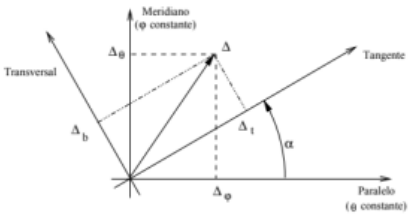


Figura 3. Deslocamentos originados da deformação sobre o cilindro

Figura 4. Decomposição nas direções tangencial e transversal

A partir desse ponto, obtém-se trivialmente outras equações como de deslocamentos relativos:

$$\begin{cases} \delta_t^E = \Delta_t^E - (\Delta_t \cos(2\alpha) - \Delta_b \sin(2\alpha)) \\ \delta_b^E = \Delta_b^E - (\Delta_t \sin(2\alpha) + \Delta_b \cos(2\alpha)) \end{cases}$$

A equação de equilíbrio de forças:

$$\begin{cases} \dfrac{dF_t}{dS} + q_t - Q_t^e \cos 2\alpha + Q_b^e \sin 2\alpha = 0 \\ F_t \Delta C_b + q_b - Q_t^e \sin 2\alpha - Q_b^e \cos 2\alpha = 0 \end{cases}$$

E as equações diferenciais:

$$\begin{cases} A_1 \dfrac{\partial \Delta_b}{\partial \varepsilon} + A_2 \dfrac{\partial \Delta_b^E}{\partial \varepsilon} = Q_b^E \dfrac{\partial \Delta_t^E}{\partial \varepsilon} + A_3 \dfrac{\partial \Delta_t}{\partial \varepsilon} \\ \dfrac{\partial \Delta_b}{\partial \varepsilon} = A_4 \dfrac{\partial \Delta_t}{\partial \varepsilon} \end{cases}$$

Método de Solução

O problema final foi resolvida utilizando um método de integração de Runge-Kutta de 4º Ordem. Sendo os dados as forças de atrito entre o tubo e armadura interna ***q*** e armadura interna e externa ***Q***, além da força axial ***F_t***, obtém-se primeiramente ***Delta_t*** e ***Delta_b***. Em seguida tendo-os, são obtidos os deslocamentos relativos, o parâmetro **Z** que define curvaturas e por fim as variações de curvatura ***DeltaC_n*** e ***DeltaC_b***. Após todos esses passos uma tensão máxima pode ser calculada para um ponto situado em um ângulo ***theta***, através da fórmula dada por:

$$\begin{cases} \sigma_t(\theta) = \dfrac{F_p}{A} + E\dfrac{e}{2}\Delta C_n(\theta) + E\dfrac{h}{2}\Delta C_b(\theta) + \dfrac{q_t a}{A \sin\alpha}\left(\dfrac{\pi}{2} - \theta\right) \\ \sigma_t^E(\theta) = \dfrac{F_p}{A^E} + E^E\dfrac{e^E}{2}\Delta C_n^E(\theta) + E^E\dfrac{h^E}{2}\Delta C_b^E(\theta) + \dfrac{Q_t^E a}{A^E \sin\alpha}\left(\dfrac{\pi}{2} - \theta\right) \end{cases}$$

Sendo ***E*** módulo de Young, ***A*** área da seção, ***e*** espessura e ***h*** largura.

5) Resultados e Discussão:

No total 5 casos de resultados foram gerados e separados em 2 grupos para análise. O primeiro grupo, ilustrado pela figura 6 abaixo, contém os casos 1, 2 e 3, no qual variou-se as forças de atrito e manteve-se uma força axial constante, similar aos resultados obtidos por Toledo et al. [3]. Para o Caso 1 utilizou-se ***Q*** = 325N/m, para o caso 2 tivemos ***Q*** = 650N/m, e para o caso 3 optou-se por ***Q*** =1300N/m . Para todos os 3 casos adotou-se ***q***=2Q e ***F_t***=3600N.

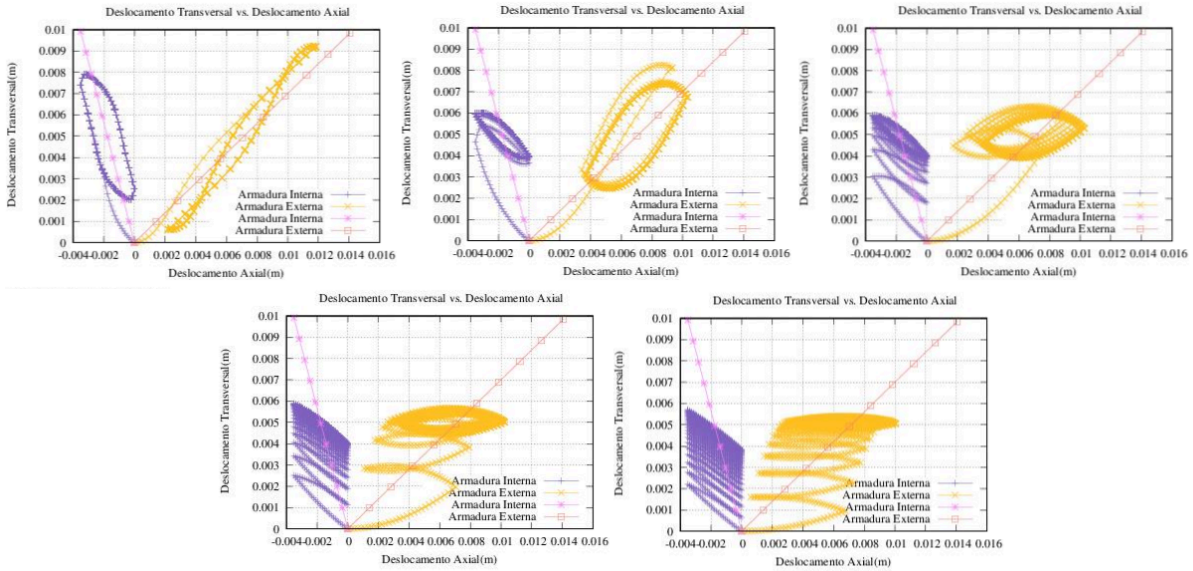


Figura 6. Resultados para o Caso 1 (topo esquerda), 2 (topo meio), 3 (topo direita), 4 (inferior esquerda), 5 (inferior direita)

O segundo grupo engloba os casos 3, 4 e 5, no qual variou-se as manteve-se as forças de atrito constantes e variou-se a força axial, de forma a gerar novos resultados. Para o Caso 3 utilizou-se ***F_t***=3600N, para o caso 4 optou-se ***F_t***=1800N, e para o caso 5 foi escolhido ***F_t***=900N. Para todos os 3 casos adotou-se ***Q***=1300N/m e ***q***=2Q.

Vale ressaltar que os resultados gerados em roxo e amarelo, oscilam como esperado sobre o menor caminho que existiria na ausência de atrito, definido pela geodésica e representado pelas semirretas em rosa.

6) Conclusões:

Nota-se que o modelo atual produziu os resultados esperados de acordo com o estudado, onde para um ciclos iniciais a armadura desliza ao longo do tubo e estabiliza-se movendo ao redor de uma mesma região ao longo do tempo. Trabalhos futuros com dados comparativos devem ser realizados utilizando dados de outros estudos ou ainda apresentando novas situações como a do ponto de touchdown, em que o riser se encontra ao solo. Continuações desse estudo podem permitir o desenvolvimento de novas técnicas além de contribuir para este desafio.

7) Referências Bibliográficas:

[1] Mendes de J. Souza. Análise numérica de risers flexíveis. Master 's thesis, Instituto de Pesquisas Tecnológicas do Estado de São Paulo, São Paulo, Brasil, 1999.

[2] J.-M. Leroy and P. Estrier. Calculation of stresses and slips in helical layers of dynamically bent flexible pipes. Oil Gas Science and Technology – Rev. IFP Copyright Editions Technip Oil Gas Science and Technology – Rev. ´ IFP, vol. 56, pp. 545–554, 2001.

[3] E. M. Toledo, L. P. S. Barra, C. V. Abreu, R. C. P. Leal-Toledo, J. R. M. Sousa, and L. M. B. Troina. A semi-analytical approach for stress analysis of riser tendons with layer interaction. In XXV CILAMCE - Iberian Latin American Congress on Computational Methods in Engineering, Recife, Brazil. Anais do XXV CILAMCE, 2004.

[4] J. Feret, J.-M. Leroy, and P. Estrier. Calculation of stresses and slips in flexible armour layers with layers interaction. Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering - OMAE, vol. 5, pp. 469–474, 1995.

Projeto de Iniciação Científica:

Desenvolvimento de Jogos para o Ensino-Aprendizagem

Bolsista: Amanda Vilas Boas Oliveira
Orientadora: Regina Célia Cerqueira de Almeida
Co-orientador: Crysttian Arantes Paixão, Universidade Federal da Bahia - UFBA
Edital / Programa: 2023/PIBIC

Palavras-chave: programação, Scratch, ensino, metodologias de ensino, tutoriais

1 Objetivos

O objetivo desse projeto é o desenvolvimento de tutoriais e jogos para o ensino-aprendizagem de alunos, tanto do ensino fundamental quanto do ensino médio, por meio da plataforma Scratch. Assim, pretende-se oferecer tutoriais com instruções detalhadas, de forma estruturada e didática, acerca dos conteúdos sobre programação. Os tutoriais abordam os temas em níveis crescentes de complexidade, para que os estudantes desenvolvam noções básicas de programação utilizando o software Scratch[1]. Os primeiros tutoriais apresentando conceitos de programação, utilizando abordagem lúdica com conteúdos práticos. Os tutoriais mais avançados descrevem a criação de diferentes jogos didáticos.

2 Introdução

A chegada da internet provocou mudanças desde fins dos anos 1990 em vários campos e ressignificou muitas palavras, como interação e colaboração, para além da gramática das redes digitais. A nova forma de distribuição de saberes influencia a própria dinâmica social, trazendo a reflexão dos seus efeitos tanto da perspectiva da comunicação, quanto da cultura, da economia, da política e principalmente da educação [1]. Consoante a isso, as possibilidades e oportunidades sobre as novas formas de se inteirar e colaborar sobre os novos saberes cresceram de forma exponencial, sobretudo acerca da computação. Entretanto, mesmo com a evolução dessa área e das tecnologias que se desenvolveram e estão cada vez mais presentes no cotidiano das pessoas, há uma ausência de contato com conhecimentos básicos de informática, especialmente de programação, por parte de uma grande parcela dos estudantes durante sua formação inicial. As noções de programação permitem que o usuário desenvolva habilidades em estruturar soluções para diferentes problemas e que

[1] https://scratch.mit.edu/

podem ser aplicadas a circunstâncias do dia a dia.

Essa falta de contato persiste por diversos fatores. Um deles é o desinteresse por parte dos estudantes acerca dos temas ligados a computação, em parte motivado pela existência de estereótipos e informações conturbadas sobre a área, uma vez que esta não faz parte da grade curricular da maioria das escolas. Além disso, algumas instituições de ensino nem mesmo possuem laboratórios de informática disponíveis para os seus estudantes, o que os priva desse tipo de educação que estimula a melhoria do seu raciocínio lógico. Desse modo, essa defasagem se estende até mesmo aos alunos universitários, os quais apresentam dificuldades em seguir ordens sucessivas para executar a solução de um problema, habilidade que pode ser desenvolvida com noções de programação.

Além disso, a disseminação de novos saberes por meio de jogos é também uma ferramenta poderosa. Segundo *Piaget*, o jogo tem o poder de desenvolver percepções, inteligência e instintos sociais das crianças, além de ser um meio eficaz para sua aprendizagem, transformando a iniciação de uma área de conhecimento considerada maçante em um uma atividade lúdica [2]. Desse modo, desenvolvemos um conjunto de tutoriais para dar acesso a esses estudantes abordando noções básicas de informática e programação através de jogos. A execução desses tutoriais consiste em utilizar a plataforma Scratch, que é um software de programação gráfico que foi desenvolvido pelo MIT (*Massachussets Institute of Tecnology*).

No Scratch é possível escrever uma sequência de instruções que dirão ao computador o que fazer, permitindo criar histórias e animações por meio de uma linguagem de programação. Esse software possui uma interface gráfica em blocos que são montados como Lego. É usado em muitas escolas de países desenvolvidos na grade de ensino, apresentando para as crianças a computação desde cedo [3]. Assim, enquanto a maioria das linguagens de programação é baseada em texto, o usuário é obrigado a aprender a linguagem e as regras de sintaxe. Com uma nova abordagem, o Scratch é uma linguagem de programação visual, o que torna o aprendizado mais fácil e interativo, uma vez que usam os atores para obedecerem aos conjuntos de instruções dadas por meio de blocos.

Portanto, os tutoriais, que estão em desenvolvimento, são para o ensino-aprendizagem dos alunos do ensino fundamental e médio, bem como orientação dos professores envolvidos. Para isso, foi feita uma pesquisa bibliográfica para seleção das temáticas dos jogos e seus diferentes graus de complexidade, bem como o desenvolvimento dos jogos de forma didática na plataforma Scratch. Assim, os tutoriais permitem o acesso a noções básicas de informática, de lógica de programação e de desenvolvimento de jogos

educacionais, capacitando os envolvidos e permitindo que os estudantes tenham a oportunidade de ocupar futuramente instituições de ensino superior ou técnico já com essas competências.

3 Materiais e Métodos

Para elaboração dos tutoriais foi desenvolvido um template para criação de uma identidade visual do material. Sendo assim, o template inclui elementos visuais que são característicos do Scratch, bem como a paleta de cores, trazendo as cores azul, roxo, laranja e verde, e imagens do ator Scratch Cat, mascote oficial do software. Além disso, cada tutorial possui uma numeração, um título condizente com o conteúdo abordado nele, uma explicação do conceito envolvido, apresentação dos blocos que serão trabalhados, uma atividade e um exercício em forma de desafio.

Consoante a isso, a estrutura dos tutoriais foi planejada para facilitar o aprendizado do aluno e criar familiaridade com o material de acordo com sua estrutura bem definida. Na Figura 01 é exibido a primeira parte do Tutorial 1 que apresenta a Interface do Scratch, que é o ambiente no qual é realizado a criação dos projetos, bem como o template adotado. De forma mais detalhada, todos os tutoriais possuem um pequeno texto padrão de boas-vindas que também explica brevemente o seu objetivo. Seguido a isso, há uma pergunta padrão formulada como “Afinal, o que é...?”, a fim de explicar e introduzir o conceito central do tutorial de maneira didática, garantindo que diferentes públicos com diferentes graus de conhecimento sobre programação possam compreender.



Figura 1 – Capa do Tutorial 1.

Desse modo, no decorrer do tutorial há uma explicação detalhada sobre a função de cada bloco que será trabalhado, com a descrição da atividade proposta e passos claros, ilustrados por imagens do software Scratch, facilitando a reprodução em caso de dúvidas. Essas imagens passam por uma simples edição no aplicativo *Paint*, onde recebem sinalizadores na cor vermelha que guiarão o leitor do tutorial. Por fim, a proposta do desafio é projetada para que o leitor possa realizá-lo de forma independente, então muitos desafios são simples adições ou pequenas modificações nos projetos já desenvolvidos.

4 Resultados e Discussão

Os tutoriais foram desenvolvidos seguindo os passos básicos para aprendizagem de lógica de programação junto aos elementos do Scratch, com foco em desenvolvimento de jogos. Portanto, foi estabelecida uma ordem lógica para os assuntos tratados, com um nível ascendente de complexidade, divididos em 30 tutoriais que podem ser setorizados em 3 grupos. Desse modo, o primeiro grupo possui um pequeno grau de complexidade, sendo os tutoriais: Apresentando a Interface do Scratch; Criando uma variável para armazenar valores; Tipos de variáveis; Realizando contas com as variáveis; Operadores aritméticos; Operadores relacionais; Estruturas condicionais; Criando um vetor; Operadores lógicos e Laços de repetição. Ao terminar o primeiro bloco, o leitor possuirá uma noção básica de programação.

Já o segundo grupo possui um grau intermediário de complexidade, pois aqui estão inseridos mais elementos do Scratch que envolvem animações, sendo os tutoriais: Combinando estruturas condicionais e de repetição; Usando os atores; Trabalhando com sons; Posição dos atores; Deslocando um ator; Criando uma animação; Modificando cenários; Criando botões; Template para a criação de jogos e Criando um jogo de tabuleiro. Dessa forma, ao terminar o segundo bloco, o leitor terá uma noção sobre o desenvolvimento de animações geradas a partir dos conceitos envolvendo programação.

Por fim, o terceiro grupo possui um grau mais elevado de dificuldade, pois são compostos por jogos simples, sendo os tutoriais: Jogo Par ou Ímpar; Jogo de Perguntas; Jogo do Labirinto; Jogo de Memória; Jogo do Gato e Rato; Jogo da Cobrinha; Jogo do PacMan; Jogo de Lançamento (Angry Birds); Jogo Corrida e Torre de Hanói. Assim, ao finalizar o terceiro e último bloco o leitor terá uma noção de desenvolvimento de jogos simples no Scratch e a lógica de programação para isso.

No entanto, é necessário reconhecer algumas pequenas limitações do Scratch. Uma delas é por ser uma linguagem de programação visual, o entendimento de sintaxe e estruturas textuais se torna limitado quando comparada com as demais linguagens de programação. Além disso, por ser baseado em blocos visuais e ser intuitivo para iniciantes, pode não preparar completamente os alunos para transição para as linguagens baseadas em texto. Portanto, esse software pode não ser adequado para o desenvolvimento de projetos com maiores complexidades e que demandam maior controle sobre o código, se tornando útil para introdução à computação e construção de uma base sólida para evolução nesta área.

5 Conclusões

Portanto, os tutoriais para ensino de programação com o Scratch se mostram uma ferramenta eficaz para o ensino-aprendizagem dos alunos devido à sua abordagem didática. Dessa forma, destaca-se a importância do contínuo investimento em infraestrutura tecnológica nas escolas e na formação de profissionais capacitadas para o ensino de programação, o que irá impactar positivamente no interesse dos alunos pela área e reduzirá a variação do nível de familiaridade destes com a informática.

Além disso, a interface intuitiva e os recursos visuais do Scratch possibilitam a compreensão dos alunos sobre conceitos fundamentais de programação. O aprendizado através dos tutoriais e os desafios propostos neles estimulam a criatividade, o pensamento crítico dos estudantes e o desenvolvimento de habilidades de gestão de problemas com o pensamento lógico sobre a implementação de algoritmos.

6 Referências

[1] SILVEIRA, Sérgio Amadeu. Democracia e Códigos Invisíveis: Como os algoritmos estão modulando comportamentos e escolhas políticas. São Paulo: editora edições Sesc SP, 2019 .

[2] PIAGET, J. Psicologia e Pedagogia. 7ª impressão. Rio de Janeiro: Editora Forense Universitária LTDA, 1985.

[3] VARELA, H. (2017). Scratch: Um jeito divertido de aprender programação. Brasil: Casa do Código.

[4] VALENTE, José Armando et al. O computador na sociedade do conhecimento. Campinas: Unicamp/NIED, p. 11-18, 1999.

[5] DA SILVA, Angela Carrancho. Educação e tecnologia: entre o discurso e a prática. Revista Ensaio: Avaliação e Políticas Públicas em Educação, v. 19, n. 72, p. 527-554, 2011.

[6] NUNES, Felipe Becker et al. Um estudo de caso sobre a importância do uso de objetos de aprendizagem no ensino fundamental como apoio pedagógico. In: Anais do Workshop de Informática na Escola. 2014. p. 542.

[7] LOPES, José Junio et al. A introdução da informática no ambiente escolar. Clube do professor, v. 23, 2004.

[8] SEVERO, C. E. P. (2021). Jogos com Scratch: em projetos práticos com linguagem de blocos. Brasil: Casa do Código.

Salvador, 12 de julho de 2024.

Amanda Vilas Boas Oliveira

Estudante

Orientador (a)

GRUPO DE MODELAGEM MOLECULAR DE SISTEMAS BIOLÓGICOS
INICIAÇÃO CIENTÍFICA

Ana Júlia Ferreira Fagundes

ANÁLISE COMPUTACIONAL DA DINÂMICA MOLECULAR DA ENZIMA N-ACETYL-L-GLUTAMATE KINASE ASSOCIADA À SUPER-RESISTÊNCIA BACTERIANA EM *KLEBSIELLA PNEUMONIAE*

Relatório referente à pesquisa sobre a N-acetil glutamato kinase de *Klebsiella pneumoniae* associado à super-resistência e a análise dos resultados da dinâmica molecular.

Orientador: Laurent Emmanuel Dardenne
Co-orientador: Leon Sulfierry Corrêa Costa
Tipo da bolsa: PIBIC/CNPq
Período do relatório: out. 2023 - jul. 2024

Petrópolis
2024

Introdução

A resistência bacteriana é um problema global e preocupante, referindo-se à capacidade de uma bactéria resistir aos efeitos dos antimicrobianos aos quais inicialmente era sensível. Este perfil é adquirido por meio de mutações genéticas e disseminado via plasmídeos e transferência horizontal de genes, além disso, também está diretamente associado ao uso excessivo e inadequado de antibióticos (SHON; BAJWA; RUSSO, 2013). Bactérias Gram-negativas, tendem a apresentar resistência a diversas classes de antibióticos. O surgimento de cepas com esse perfil de super-resistência é considerado uma ameaça urgente à saúde humana, tornando a busca e estudo de novos alvos moleculares para antimicrobianos necessária na medicina moderna (HOLT et al., 2015).

Klebsiella pneumoniae é uma bactéria Gram-negativa encontrada no trato gastrointestinal e na nasofaringe humana, além de superfícies de mucosa de animais e no ambiente (WANG et al., 2020). Embora comensal no intestino, é um patógeno oportunista significativo, especialmente em pacientes hospitalizados e imunocomprometidos, podendo causar infecções graves com altas taxas de mortalidade. Esse índice elevado de mortalidade está relacionado a fatores de virulência e resistência antimicrobiana ampla, incluindo resistência a carbapenêmicos, antibióticos de "último recurso" (NORDMANN; POIREL, 2014). A *K. pneumoniae* possui o ciclo de produção da L-arginina, crucial para a síntese de proteínas e armazenamento de energia (RAMÓN-MAIQUES et al., 2002). A N-acetilglutamato kinase (NAGK), uma enzima do ciclo, é essencial para a fosforilação do N-acetilglutamato (NAG) e, por ser exclusiva de bactérias e plantas, representa um alvo promissor para a inibição seletiva da biossíntese de L-arginina (ALONSO; RUBIO, 1989). O estudo computacional da NAGK é, portanto, fundamental para entender sua dinâmica e estrutura em nível atômico.

Utilizou-se no presente trabalho a dinâmica molecular para modelar o comportamento desta biomolécula. Para a quantificação dos dados, foram empregadas análises de RMSD (Root Mean Square Deviation) para medir a estabilidade estrutural, PCA (Principal Component Analysis) para identificar os modos principais de movimento, e FEL (Free Energy Landscape) para mapear as conformações energeticamente favoráveis. O presente trabalho visa contribuir para o entendimento molecular detalhado deste alvo promissor para o desenvolvimento de novos tratamentos associados à super-resistência bacteriana.

Objetivos

Objetivo geral

- Aprofundar o entendimento sobre a resistência manifestada por *Klebsiella pneumoniae*, centrando-se na dinâmica molecular da enzima N-acetyl-L-glutamate kinase, analisar e identificar fenômenos e motivos importantes a partir dos resultados obtidos.

Objetivo específico

- Revisão da literatura a respeito do tema proposto.
- Definição da estrutura tridimensional da enzima N-acetyl-L-glutamate kinase utilizando dados disponíveis no Protein Data Bank e através de técnicas de modelagem comparativa.
- Parametrização e preparação adequada da estrutura.
- Condução das dinâmicas moleculares da enzima sob condições similares a seu ambiente intracelular, buscando identificar conformações e interações representativas.
- Realizar análises qualitativas e quantitativas acerca dos resultados das dinâmicas moleculares, identificando estruturas e motivos importantes para o funcionamento da enzima.

Metodologia

A revisão da literatura e a coleta de informações foram realizadas principalmente nas plataformas de busca PubMed e Google Scholar. Foram utilizadas palavras-chave relevantes ao tema proposto para garantir uma pesquisa direcionada e abrangente.

A estrutura tridimensional (3D) da N-acetilglutamato kinase (NAGK) de *Klebsiella pneumoniae* foi gerada a partir de sua sequência de aminoácidos obtida no NCBI, utilizando a técnica de modelagem comparativa na plataforma Swiss-Model (SCHWEDE, 2003), com base na estrutura homóloga da NAGK de *Escherichia coli* (PDB ID: 1GS5). Após a modelagem, o software Propka, via PDB2PQR (OLSSON et al., 2011), foi usado para adicionar átomos de hidrogênio e determinar os estados de protonação dos resíduos, e o arquivo resultante foi parametrizado usando o tLeap, um módulo do pacote Amber (CASE et al., 2005).

Para a simulação de dinâmica molecular, foi utilizada a proteína com o campo de força ff19SB e água com o campo de força OPC3, conhecida por sua precisão em propriedades termodinâmicas. A proteína foi imersa em uma caixa octaédrica de água explícita, neutralizada com 6 íons Na+ e ajustada a uma concentração iônica de 150 mM. A simulação foi realizada no software Amber, começando com a minimização de energia para remover tensões estruturais. O sistema foi então relaxado em quatro etapas ao longo de 1 nanossegundo, envolvendo aquecimento gradual e remoção progressiva de restrições. Após um equilíbrio de 5 nanossegundos, a produção da simulação durou 100 nanossegundos, totalizando 1 microssegundo de dinâmica molecular em 10 réplicas.

No conjunto total de trajetórias, foram realizadas análises detalhadas de RMSD, PCA e FEL. A análise de RMSD foi utilizada para medir as variações na estrutura da proteína ao longo do tempo, comparando a conformação da proteína em diferentes instantes com uma estrutura de referência. A PCA foi empregada para reduzir a dimensionalidade dos dados e identificar os modos de movimento predominantes da proteína, revelando padrões importantes de dinâmica conformacional (SILVA et al., 2022). A análise de FEL foi conduzida para mapear o perfil energético da proteína, identificando estados conformacionais estáveis e as barreiras energéticas entre eles (FRAUENFELDER; SLIGAR; WOLYNES, 1991). Os links para os algoritmos utilizados estão disponíveis no Github (GitHub_RMSD_PCA, GitHub_FEL).

Resultados e Discussão

A análise do desvio quadrático médio (RMSD) do backbone da proteína NAGK forneceu insights significativos sobre sua estabilidade estrutural ao longo da simulação (Figura 1-A). A média geral do RMSD é de 1.7470, com um desvio padrão de 0.4261, indicando variação considerável entre as diferentes conformações. Regiões de alta estabilidade têm uma média de RMSD de 1.3500 e desvio padrão de 0.1379, sugerindo uma conformação mais constante. Em contraste, regiões de baixa estabilidade mostram uma média de RMSD de 2.0081 e desvio padrão de 0.3975, indicando flutuações estruturais significativas. A região entre 200.000 e 225.000 frames, identificada como um mínimo energético global, apresenta uma média de RMSD de 1.2504 e desvio padrão de 0.0871, destacando sua estabilidade excepcional.

A análise de PCA revelou que o PC1 explica 64,71% da variância, o PC2 explica 24,74% e o PC3 explica 10,55%. O PC1 é influenciado negativamente pelo movimento da estrutura proteica identificado como Amplitude1 (primeiro componente principal), enquanto o PC2 e o PC3 são influenciados positivamente pelos movimentos identificados como Amplitude2 e Amplitude3, respectivamente (Figura 1-C). Na região de mínimo energético global, observou-se alterações conformacionais significativas nas fitas β3, β4 e hélice G (Figura 1-B.1 e B.2), com Amplitude2 contribuindo majoritariamente para a variância no PC1, seguida por uma influência oposta de Amplitude3. A representação da FEL (Figura 1-B), gerado a partir das duas primeiras componentes principais, exibe toda a paisagem energética ao longo da trajetória e, consequentemente, a região onde as conformações mais estáveis e energeticamente favoráveis da NAGK são predominantemente encontradas (Figura 1-B.2). Esta visualização permite identificar estados conformacionais relevantes e os possíveis caminhos de transição entre eles, oferecendo insights cruciais sobre a dinâmica e funcionalidade da biomolécula.

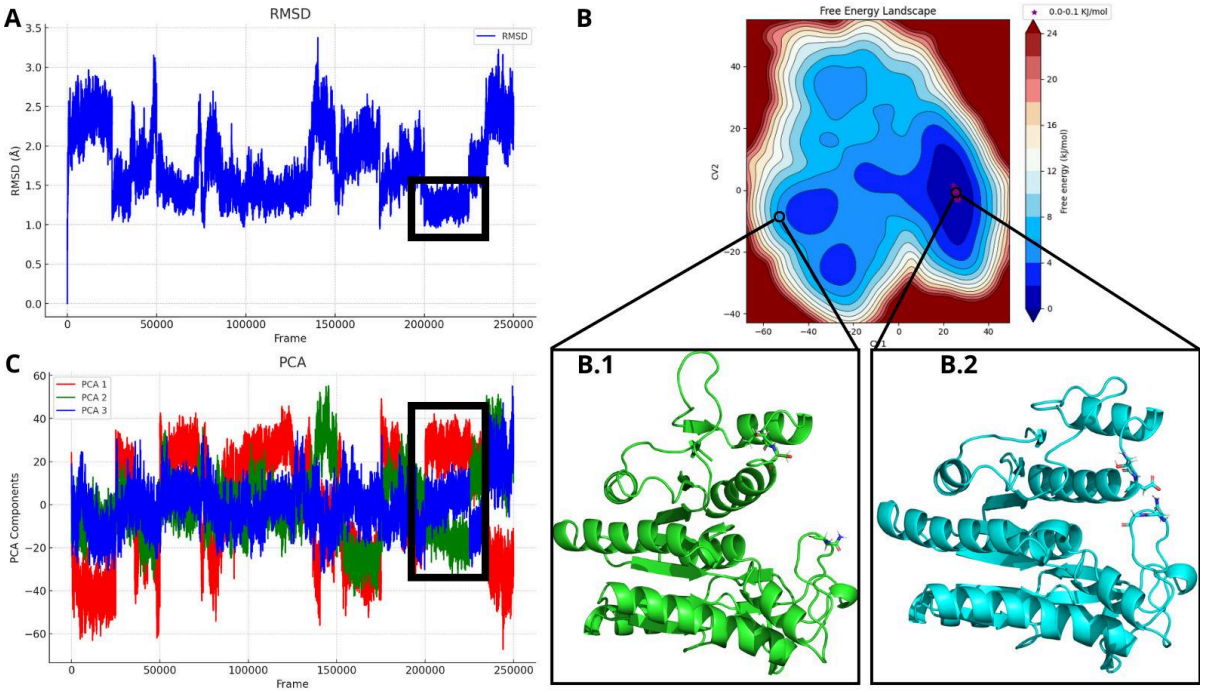


Figura 1 - A) RMSD; B) FEL, onde B.1 corresponde à abertura da pinça e B.2 corresponde à região de mínimo energético, onde a pinça encontra-se fechada; C) PCA. As áreas destacadas correspondem à região de mínimo energético.

Essas análises integradas de RMSD, PCA e FEL destacam a estabilidade estrutural e conformacional da proteína NAGK, especialmente na região entre 200.000 e 225.000 frames, sugerindo que essa conformação é biologicamente relevante e essencial para a funcionalidade da enzima. A estabilidade observada nesta região, correlacionada com um mínimo global energético, indica que as conformações nessa faixa são as mais prováveis de serem encontradas na natureza, desempenhando um papel crucial na atividade enzimática.

Conclusão

O estudo detalhado da dinâmica molecular da enzima N-acetil-L-glutamato kinase (NAGK) de *K. pneumoniae* revelou mudanças conformacionais e interações cruciais para seu funcionamento. Utilizando dados do NCBI e PDB, a estrutura tridimensional da enzima foi modelada e submetida a simulações computacionais com o software Amber, fornecendo insights valiosos sobre as interações atômicas e conformação da enzima. As análises de RMSD, PCA e FEL indicaram que as alterações no grampo formado pelas fitas β3, β4 e hélice G são fundamentais para a abertura e fechamento do sítio catalítico. A região entre 200.000 e 225.000 frames, identificada como um mínimo energético global, apresenta uma conformação estável e energeticamente favorável, essencial para a atividade enzimática.

Esses resultados destacam a importância de resíduos como Met214, Asp212 e Lys217 na hélice G, além de Arg66, Asp212, Asn63 no grampo para a estabilidade energética da NAGK. A análise integrada dos dados fornece uma compreensão mais profunda dos movimentos e do comportamento da enzima em condições naturais, abrindo novas perspectivas para o desenvolvimento de inibidores seletivos e novos fármacos contra a resistência da *K. pneumoniae*, especialmente em infecções nosocomiais visto este estudo ser pioneiro ao aplicar dinâmica molecular ao respectivo alvo.

6. Referências

ALONSO, E.; RUBIO, V. Participation of ornithine aminotransferase in the synthesis and catabolism of ornithine in mice. Studies using gabaculine and arginine deprivation. **Biochemical Journal**, v. 259, n. 1, p. 131–138, 1 abr. 1989. Disponível em:<https://doi.org/10.1042/bj2590131>. Acesso em: 5 jul 2024

CASE, D. A. et al. The Amber biomolecular simulation programs. **Journal of Computational Chemistry**, v. 26, n. 16, p. 1668–1688, 30 dez. 2005. Disponível em:<https://doi.org/10.1002/jcc.20290>. Acesso em: 5 jul 2024

FRAUENFELDER, H.; SLIGAR, S. G.; WOLYNES, P. G. The Energy Landscapes and Motions of Proteins. **Science**, v. 254, n. 5038, p. 1598–1603, 13 dez. 1991. Disponível em:<https://doi.org/10.1126/science.1749933>. Acesso em: 5 jul 2024

HOLT, K. E. et al. Genomic analysis of diversity, population structure, virulence, and antimicrobial resistance in *Klebsiella pneumoniae* , an urgent threat to public health. **Proceedings of the National Academy of Sciences**, v. 112, n. 27, 7 jul. 2015. Disponível em:<https://doi.org/10.1073/pnas.1501049112>. Acesso em: 5 jul 2024

NORDMANN, P.; POIREL, L. The difficult-to-control spread of carbapenemase producers among Enterobacteriaceae worldwide. **Clinical Microbiology and Infection**, v. 20, n. 9, p. 821–830, set. 2014. Disponível em:<https://doi.org/10.1111/1469-0691.12719>. Acesso em: 5 jul 2024

OLSSON, M. H. M. et al. PROPKA3: Consistent Treatment of Internal and Surface Residues in Empirical p K_a Predictions. **Journal of Chemical Theory and Computation**, v. 7, n. 2, p. 525–537, 8 fev. 2011. Disponível em:<https://doi.org/10.1021/ct100578z>. Acesso em: 5 jul 2024

RAMÓN-MAIQUES, S. et al. Structure of Acetylglutamate Kinase, a Key Enzyme for Arginine Biosynthesis and a Prototype for the Amino Acid Kinase Enzyme Family, during Catalysis. **Structure**, v. 10, n. 3, p. 329–342, mar. 2002. Disponível em:<https://doi.org/10.1016/S0969-2126(02)00721-9>. Acesso em: 5 jul 2024

SCHWEDE, T. SWISS-MODEL: an automated protein homology-modeling server. **Nucleic Acids Research**, v. 31, n. 13, p. 3381–3385, 1 jul. 2003. Disponível em:<https://doi.org/10.1093/nar/gkg520>. Acesso em: 5 jul 2024

SHON, A. S.; BAJWA, R. P. S.; RUSSO, T. A. Hypervirulent (hypermucoviscous) *Klebsiella pneumoniae*. **Virulence**, v. 4, n. 2, p. 107–118, 15 fev. 2013. Disponível em:<https://doi.org/10.4161/viru.22718>. Acesso em: 5 jul 2024

SILVA, J. C. A. et al. Investigation of the structural dynamics of a knotted protein and its unknotted analog using molecular dynamics. **Journal of Molecular Modeling**, v. 28, n. 4, p. 108, 31 abr. 2022. Disponível em:<https://doi.org/10.1007/s00894-022-05094-y>. Acesso em: 5 jul 2024

WANG, G. et al. The Characteristic of Virulence, Biofilm and Antibiotic Resistance of Klebsiella pneumoniae. **International Journal of Environmental Research and Public Health**, v. 17, n. 17, p. 6278, 28 ago. 2020. Disponível em:<https://doi.org/10.3390/ijerph17176278>. Acesso em: 5 jul 2024

Relatório de Atividades

1) Dados Gerais

Projeto: Modelagem Computacional para Análise de dados de Tratamento de Câncer em saúde Coletiva

Nome do bolsista: Ana Paula de Oliveira Souza
Orientadores: José Karam Filho e Paulo Cabral Filho
Tipo de bolsa: Pibic
Período do relatório: agosto/2023 a julho /2024

2) Objetivos

Criação do site "gestao-saude.lncc.br" utilizando React e Next.js, para demonstração deste projeto, em parceria com as instituições IMS (Instituto de Medicina Social Hesio Cordeiro) da UERJ e FAETERJ (Faculdade de Educação Tecnológica do Estado do Rio de Janeiro) Petrópolis, para amostra de diversas linhas de pesquisa.

Atualização do banco de dados com os registros referentes ao ano de 2023.

Atualização da tabela baseada no IPCA com índices específicos de saúde e atualização dos índices até 2024.

Criação de novos campos na tabela principal (INTERNACAO), além dos quatro existentes anteriormente, utilizando as atualizações da tabela IPCA.

Início da fase de análise de dados, com diversas consultas ao banco para geração de gráficos para o site.

Atualização dos scripts de transferência de dados referentes aos registros que anteriormente eram 27 para cada estado, tornando-se apenas que busca nos arquivos e transfere de um em um.

Criação de documentação do projeto para facilitar a continuação e novas atualizações do site.

3) Introdução

O Projeto tem como principal objetivo a criação de um banco de dados automatizado (chamado DataSusCancer) e a geração de um sistema de gestão de saúde digital. O desenvolvimento deste projeto está sendo realizado em parceria com o Instituto de Medicina Social Hesio Cordeiro (IMS) da UERJ e a Faculdade de Educação

Tecnológica do Estado do Rio de Janeiro (FAETERJ) Petrópolis. Este trabalho inclui a criação de um site , "gestao-saude.lncc.br", utilizando as tecnologias React e Next.js.

Para garantir a organização e o versionamento do código, foi criado um repositório remoto utilizando GIT. A publicação do site, por enquanto, está restrita ao ambiente interno do Laboratório Nacional de Computação Científica (LNCC), devido ao fato de estar em fase de desenvolvimento.

Uma parte significativa do projeto envolveu a atualização do banco de dados com os registros referentes ao ano de 2023 e a atualização da tabela IPCA com índices específicos de saúde até 2024. Além disso, foram criados novos campos na tabela principal (INTERNACAO), com base nas atualizações da tabela IPCA.

A fase de análise de dados também foi iniciada, realizando diversas consultas ao banco para gerar gráficos que serão exibidos no site. Para otimizar o processo de transferência de dados, foi desenvolvido um script unificado que substituiu os 27 scripts anteriormente utilizados para cada estado. Este script unificado busca nos arquivos e transfere os dados de forma automatizada.

Por fim, foi criada uma documentação detalhada do projeto, destinada a futuros bolsistas, para que possam seguir e implementar o processo de criação e atualização do banco de dados, bem como a manutenção e atualização do site, facilitando a continuidade do projeto e também assegurando a sustentabilidade e a escalabilidade da solução desenvolvida.

4) Material e Métodos

React e Next.js: Utilizados para o desenvolvimento do site "gestao-saude.lncc.br".
GIT: utilizado para o controle de versionamento do código e colaboração no desenvolvimento.

MySQL: utilizado para a criação e gerenciamento do banco de dados.
Bash Scripts: utilizados para automatizar a criação de tabelas e a transferência de dados.

FileZilla: utilizado para a transferência de arquivos.
Máquina Virtual Pantanal: Ambiente de desenvolvimento e testes.

Criação do Site: desenvolvimento do site "gestao-saude.lncc.br" usando React e Next.js.

Integração com as instituições IMS (UERJ) e FAETERJ (Petrópolis) para exibir diversas linhas de pesquisa.

Gerenciamento de Versionamento: criação de um repositório remoto no GIT para facilitar o controle de versão e a colaboração entre desenvolvedores.

Publicação do Site: colaboração com o setor de helpdesk do LNCC para a publicação do site em ambiente interno, garantindo que todas as funcionalidades fossem testadas .

Inserção de novos registros referentes ao ano de 2023 no banco de dados MySQL.

Atualização da tabela IPCA com índices específicos de saúde e atualização dos índices até 2024.

Criação de novos campos na tabela principal: adição de novos campos na tabela principal (INTERNACAO) baseando-se nas atualizações da tabela IPCA.

Análise de Dados: realização de diversas consultas ao banco de dados para a geração de gráficos e análises que serão exibidos no site.

Automatização da transferência de dados: desenvolvimento de um script bash unificado para a transferência de dados de arquivos CSV de cada estado, substituindo os 27 scripts individuais anteriormente utilizados.

Documentação do Projeto: criação de documentação detalhada para o projeto, permitindo continuidade e atualizações futuras.

5) Resultados e Discussão

Criação do Site "gestao-saude.lncc.br": desenvolvido com React e Next.js, o site serve como plataforma de demonstração para várias linhas de pesquisa em parceria com IMS (UERJ) e FAETERJ (Petrópolis).

Implementação eficiente de GIT, facilitando a colaboração e mantendo um histórico claro de alterações.

Publicado internamente no LNCC para testes e feedback contínuo antes do lançamento público.

Atualização com registros de 2023 e índices de saúde até 2024 na tabela IPCA.

Adição de novos campos na tabela INTERNACAO, permitindo análises mais detalhadas.

Consultas ao banco de dados resultaram em gráficos informativos, exibidos no site, abrangendo internações por ano, mês, região, tipo de unidade gestora, raça, cor, gênero, e complexidade.

Novo script bash unificado para transferência de dados de arquivos CSV, substituindo 27 scripts individuais.

Documentação detalhada criada para facilitar a continuidade do projeto por novos bolsistas.

A eficiência do desenvolvimento do site foi possível pela escolha do React e Next.js, colaborando com IMS e FAETERJ para aumentar o alcance e a relevância do projeto. Para o gerenciamento de versionamento, o GIT essencial tanto na organização quanto na colaboração, mantendo um histórico claro de mudanças. Testes internos do site garantiram uma plataforma robusta e confiável antes do lançamento público. A atualização do banco de dados e da tabela IPCA garantiu dados atualizados e relevantes, essenciais para análises e decisões precisas. A unificação de scripts aumentou a eficiência e reduziu erros na transferência de dados.
A documentação realizada possibilitará a continuidade de novos desenvolvimentos.

6) Conclusões

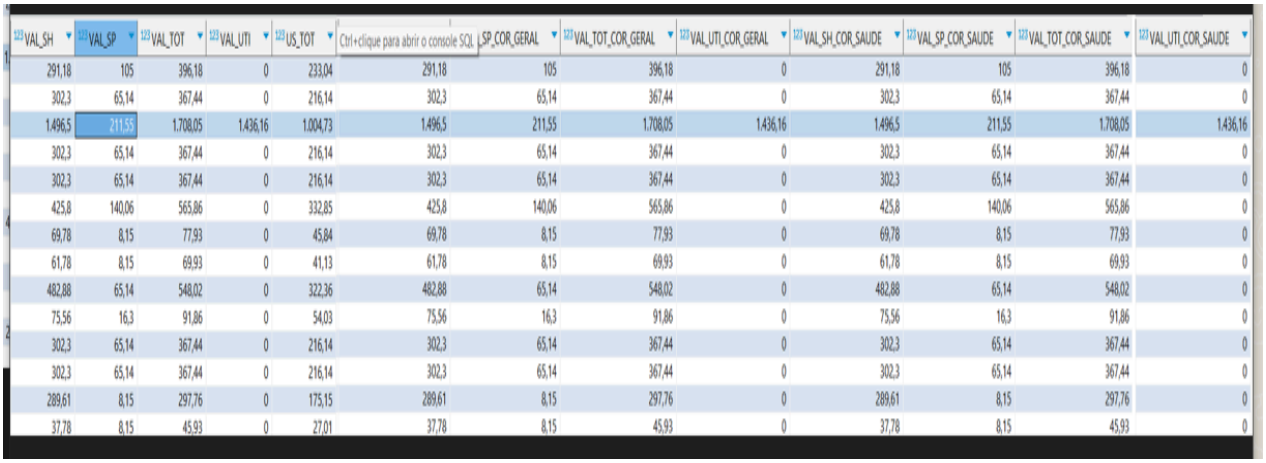
No decorrer do projeto DataSusCancer, diversas metas foram atingidas, contribuindo significativamente para o avanço das pesquisas em saúde pública e gestão de dados. As principais conclusões obtidas são:
A criação do site "gestao-saude.lncc.br" utilizando React e Next.js foi bem-sucedida, proporcionando uma plataforma eficiente e intuitiva para pesquisas em saúde.
A utilização do GIT facilitou o controle de versão e a colaboração entre os desenvolvedores, garantindo a integridade do código e a eficiência do processo de desenvolvimento.
A publicação do site em ambiente interno permitiu testes extensivos e feedback contínuo, garantindo uma plataforma robusta e livre de erros antes do lançamento público.
A atualização do banco de dados com registros de 2023 e a inclusão de novos campos na tabela principal (INTERNACAO) ampliaram a capacidade de análise e precisão dos dados, essenciais para a tomada de decisões.
A unificação dos scripts de transferência de dados melhorou a eficiência e reduziu a possibilidade de erros, simplificando significativamente o processo de atualização dos registros.
A criação de uma documentação detalhada do projeto assegura a continuidade e a facilidade para novas atualizações e desenvolvimentos, garantindo a sustentabilidade do projeto a longo prazo.
A geração de gráficos e tabelas a partir de consultas ao banco de dados permite uma visualização clara e informativa dos dados, facilitando análises detalhadas e decisões baseadas em evidências.
O projeto alcançou seus objetivos principais, estabelecendo uma base sólida para futuras expansões e melhorias. A colaboração entre as instituições envolvidas (LNCC, IMS/UERJ, e FAETERJ) foi fundamental para o sucesso do projeto, destacando a importância da cooperação interdisciplinar em pesquisas de saúde pública.

7) Referências Bibliográficas

1. React. React documentation. Disponível em: *https://react.dev/learn*;
2. Next.js. Next.js documentation. Disponível em: *https://nextjs.org/docs*;

3. Curso de React e NextJs – CFBCursos - YouTube. Disponível em: *https://www.youtube.com/watch?v=k9IG_tPonwo&list=PLx4x_zx8csUhxQHo-w0xAIMrmt8WgAowL*;
4. Curso de Next.js – Matheus Battisti -Hora de Codar - YouTube. Disponível em: *https://www.youtube.com/watch?v=XHrbg2iYNCg&list=PLnDvRpP8BnezfJcfiCIWskFOLODeqI_Ft* ;
5. GIT. GIT documentation. Disponível em*: https://git-scm.com/doc*;
6. Curso GIT - Tiago Matos - YouTube. Disponível em: *https://www.youtube.com/watch?v=2c7yWlpWDJM&list=PLcoYAcR89n-qbO7YAVj5S0alABLis_QVU*
7. Cabral Filho, P. Cálculo dos índices IPCA. Disponível em: *http://gestao-saude.lncc.br/ipca*;

Anexos : Tabelas e Gráficos



VAL_SH	VAL_SP	VAL_TOT	VAL_UTI	US_TOT	SP_COR_GERAL	VAL_TOT_COR_GERAL	VAL_UTI_COR_GERAL	VAL_SH_COR_SAUDE	VAL_SP_COR_SAUDE	VAL_TOT_COR_SAUDE	VAL_UTI_COR_SAUDE
291,18	105	396,18	0	233,04	291,18	105	396,18	0	291,18	105	396,18	0
302,3	65,14	367,44	0	216,14	302,3	65,14	367,44	0	302,3	65,14	367,44	0
1.496,5	211,55	1.708,05	1.436,16	1.004,73	1.496,5	211,55	1.708,05	1.436,16	1.496,5	211,55	1.708,05	1.436,16
302,3	65,14	367,44	0	216,14	302,3	65,14	367,44	0	302,3	65,14	367,44	0
302,3	65,14	367,44	0	216,14	302,3	65,14	367,44	0	302,3	65,14	367,44	0
425,8	140,06	565,86	0	332,85	425,8	140,06	565,86	0	425,8	140,06	565,86	0
69,78	8,15	77,93	0	45,84	69,78	8,15	77,93	0	69,78	8,15	77,93	0
61,78	8,15	69,93	0	41,13	61,78	8,15	69,93	0	61,78	8,15	69,93	0
482,88	65,14	548,02	0	322,36	482,88	65,14	548,02	0	482,88	65,14	548,02	0
75,56	16,3	91,86	0	54,03	75,56	16,3	91,86	0	75,56	16,3	91,86	0
302,3	65,14	367,44	0	216,14	302,3	65,14	367,44	0	302,3	65,14	367,44	0
302,3	65,14	367,44	0	216,14	302,3	65,14	367,44	0	302,3	65,14	367,44	0
289,61	8,15	297,76	0	175,15	289,61	8,15	297,76	0	289,61	8,15	297,76	0
37,78	8,15	45,93	0	27,01	37,78	8,15	45,93	0	37,78	8,15	45,93	0

Figura 1. Campos Atualizados



Figura 2. Primeira página do Site desenvolvido

PROGRAMA DE BOLSAS PIBIC/LNCC

RELATÓRIO DE ATIVIDADES

1) Dados Gerais

Título do projeto: Assimilação de dados no modelo WRF para uso operacional em previsão de tempo e simulação
Bolsista: Gabriel Costa Chaves
Orientadores: Roberto Pinto Souto (LNCC); Juliana Aparecida Anochi (INPE), Helaine Cristina Moraes Furtado (UFOPA)
Tipo de bolsa: PIBIC/CNPq
Período do Relatório: Julho de 2023 a Julho de 2024.

2) Objetivos

Objetivos abordados durante o período da pesquisa:

- Implementar o WRF no Santos Dumont e no sistema computacional híbrido do Instituto Nacional de Pesquisas Espaciais (INPE):
  - WRF já instalado no SDumont.
- Ativar o módulo de assimilação de dados variacional do WRF (método 3DVar):
  - Em andamento para a preparação das rodadas do WRF para o domínio da região de interesse.

3) Introdução

Muitos problemas em geociências requerem a estimação ao longo do tempo do estado do sistema a partir de uma sequência de medidas ruidosas. Esses sistemas podem ser descritos por meio de modelos matemáticos, isto é, por um conjunto de equações diferenciais. No entanto, descrever fenômenos físicos por meio da modelagem matemática, é uma atividade passível de erro, uma vez que qualquer modelo é sempre uma aproximação imperfeita da realidade. Uma alternativa, para melhorar a descrição desses modelos é adicionar observações ao sistema. Essas observações consistem de dados medidos obtidos a partir de experimentos. A combinação dessas fontes de informação pode ser realizada de modo eficaz fazendo uso das ferramentas de assimilação de dados.

A descrição da maioria dos fenômenos físicos por meio de equações diferenciais envolve erros de modelagem. Para sistemas operacionais de previsão, uma estratégia para mitigar os erros de modelagem é adicionar alguma informação real do sistema físico ao modelo

matemático. Esta informação adicional consiste de observações (valores medidos) do fenômeno que se deseja modelar. No entanto, os dados observados devem ser inseridos apropriadamente, para evitar uma degradação no desempenho da previsão – "choque dos dados". Técnicas de assimilação de dados são ferramentas que combinam de modo eficaz observações com dados de modelos físico-matemáticos para a determinação da melhor condição inicial (análise), que é usado para executar o modelo de previsão. Esse processo é fundamental na prática operacional da previsão numérica do tempo (PNT).

O presente projeto tem por meta implantar um novo sistema de assimilação de dados baseado em redes neurais artificiais e árvores de decisão (algoritmos de aprendizagem de máquina) no modelo WRF (*Weather Research and Forecasting*) em meso-escala para a região Amazônica para a previsão numérica do tempo e a simulação de eventos ambientais na região Oeste do Pará. Este projeto propõe a ativação do método de assimilação de dados existente no sistema WRF e a implementação do método de assimilação de dados por aprendizagem de máquina.

Nesta primeira etapa do projeto, foram testados *scripts* de execução de uma instalação do WRF v4.2 no supercomputador Santos Dumont. É mostrado neste relatório,

4) Metodologia

Para cumprimento das atividades nos dois primeiros meses da pesquisa, fez-se necessário ter o entendimento da estruturação do ambiente no supercomputador Santos Dumont, para a execução do modelo WRF.

- Modelo WRF v4.2 já previamente compilado no supercomputador Santos Dumont
- Aquisição de dados de condição inicial (GFS) para o modelo WRF
- Preparação das rodadas de 48h de integração do WRF, para todos os dias de determinado ano.

Na continuação da pesquisa, serão realizadas tarefas a fim de obter redes neurais que irão emular a assimilação de dados, para reduzir o custo computacional da previsão, sem prejuízo da qualidade da análise. A seguir, as etapas da metodologia no prosseguimento da pesquisa:

- Aquisição de dados provenientes de estações meteorológicas do Banco de Dados Meteorológicos (BDMET) do Instituto Nacional de Meteorologia (INMET);
- Selecionar as variáveis: precipitação, temperatura máxima e mínima, evaporação, umidade relativa e temperatura média no período em que o modelo for executado;
- Coletar dados ambientais obtidos por equipamentos em solo, sondagens por balões, instrumentos de satélite (*Total Ozone Mapping Spectrometer* (TOMS), *Ozone Monitoring Instrument* (OMI), *Microwave Limb Sounder* (MLS)), dados de reanálise do *European Centre for Medium-Range Weather Forecast* (ECMWF – ERA - *Interim daily*) entre outros serão utilizados a fim de validar as simulações realizadas pelo modelo WRF;
- Analisar e interpretar dados meteorológicos;
- Produzir rotinas computacionais para seleção automática das séries temporais a serem visualizadas.
- Implementar uma rede neural artificial como uma técnica de assimilação de dados aplicada ao modelo WRF.

5) Resultados e Discussão

O desenvolvimento da pesquisa requer o uso de um modelo de previsão numérica do tempo e um modelo de assimilação de dados; o modelo predeterminado para a pesquisa é o WRF, da NCAR. O modelo de assimilação de dados é o WRFDA, também desenvolvido pela NCAR.

De modo geral, o modelo usado é um sistema chamado ARW (*Advanced Research WRF*), que engloba outros subprogramas para a execução de diferentes tipos de pesquisas envolvendo modelagem e previsão numérica do clima. O tipo de pesquisa específica deste projeto é a assimilação de dados, quanto à previsão numérica do tempo.

Como expresso anteriormente, o objetivo da pesquisa é avaliar a eficiência e a eficácia da assimilação de dados por meio de redes neurais artificiais, em detrimento à técnica empregada pelo WRFDA. Nesse sentido, é necessário que se realizem experimentos primeiramente com o modelo de assimilação de dados padrão, o WRFDA; depois, tendo os resultados catalogados, como condições de controle, realizam-se experimentos com o modelo de assimilação de dados por rede neural artificial.

5.1) Fluxo de trabalho do ARW

Como supracitado, o ARW é um sistema de programas que podem ser usados conjuntamente para diferentes propósitos de pesquisa em modelagem climática. Quando se trata do seu funcionamento geral, pode-se dizer que o uso da suíte ARW envolve, primeiramente, a alimentação da pesquisa com dados externos, dados que são simulados ou não, e que alimentam o processo de modelagem e previsão climática pelo ARW; além disso, a suíte envolve o pré processamento dos dados iniciais citados anteriormente, por meio do WPS; em seguida, a etapa de previsão e assimilação de dados é tratada pelo WRF, que envolve a interação entre o WRF em si e o WRFDA; derradeiramente, os resultados podem ser, graficamente ou não, visualizados por programas com finalidades de análise, ou seja, uma etapa de pós processamento e visualização de dados.

5.2.1) Dados externos

Os dados externos a serem introduzidos, como alimentação inicial do ARW, são dados GFS, que são dados climáticos simulados, e dados de observação, que são dados climáticos reais. Ambos os dados devem ser combinados e pré-processados, especialmente pelo sistema de assimilação de dados, WRFDA.

5.2.2) WPS

O WPS é o sistema de pré processamento de dados do ARW, e é composto, basicamente, por três subprogramas, a saber, o ungrib, que processa os dados de entrada no formato GRIB2; o geogrid, que define os domínios de simulação; e o metgrid, que realiza a interpolação de dados terrestres.

5.2.3) WRFDA

O WRFDA é o modelo de assimilação de dados, que implementa técnicas variacionais tridimensionais, como 3D-VAR e 4D-VAR. Esse modelo pode ser usado para inserir dados observacionais nos dados de análise interpolados pelo WPS. Também pode ser usado para atualizar as condições iniciais do modelo WRF, quando este é usado em modo cíclico, i.e. em minimização iterativa de erro.

5.2.4) WRF

Este é o principal componente do ARW, que consiste no próprio modelo de previsão, e implementa uma série de funcionalidades matemáticas do âmbito da modelagem climática.

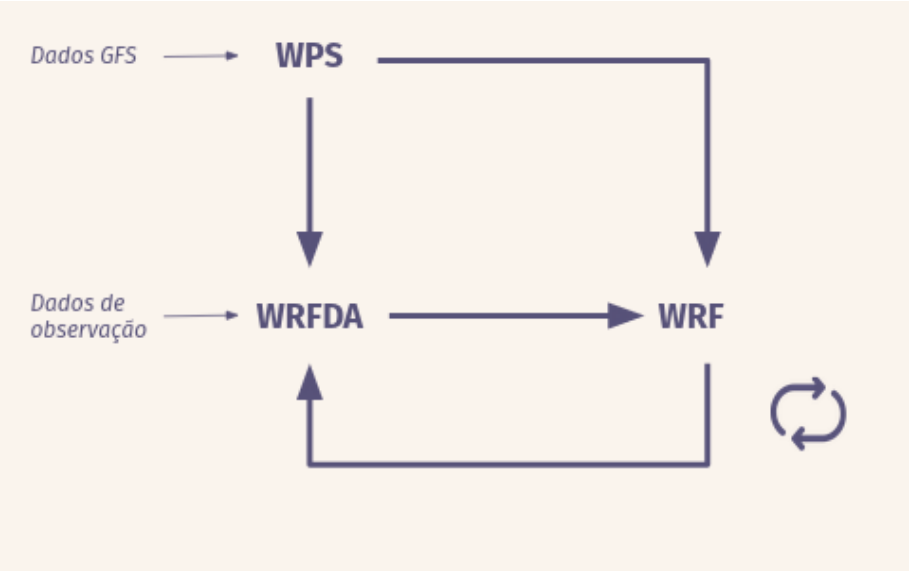


Figura 1: Funcionamento básico do WRF, com minimização iterativa de erro.

5.2.5) Ferramentas de visualização de dados

Para a visualização dos dados gerados pelo ARW, pode-se usar programas como o GRADS e Panoply, e linguagens de programação, como Julia, R e Haskell.

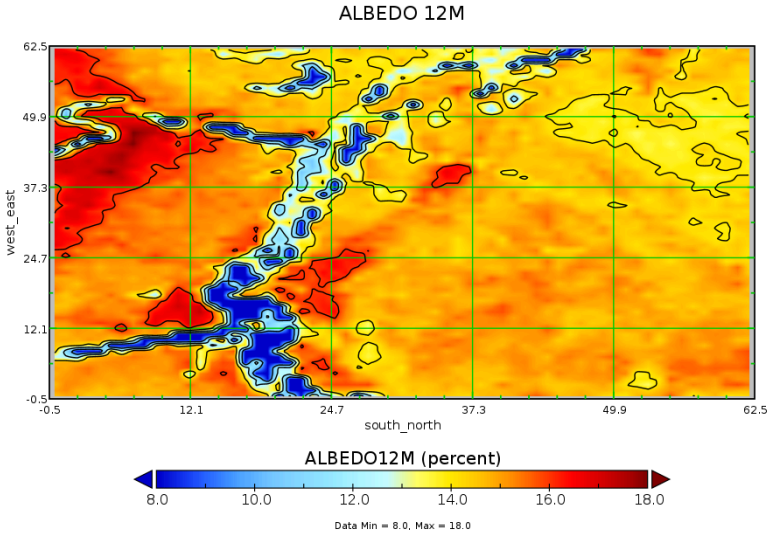


Figura 2: Exemplo de representação gráfica dos dados de albedo de uma rodada de previsão do ano 2018.

5.3) Ambiente de trabalho

A execução dos experimentos e, consequentemente, da pesquisa, requer ao menos uma estação de trabalho suficientemente poderosa. O projeto dispõe de três supercomputadores para as experimentações, a saber, o supercomputador Santos Dumont, do LNCC; o

supercomputador LacHibrido, do INPE; e o supercomputador Tapajós, da UFOPA. O bolsista dispõe de acesso ao supercomputador SDumont e ao supercomputador Tapajós.

O modelo WRF e o modelo WRFDA estavam, de antemão, prontos no supercomputador SDumont, mas não no Tapajós. Durante vários meses, o bolsista desenvolveu tarefas no supercomputador SDumont, com o modelo presente. A partir de dezembro de 2023, no entanto, ele passou a compilar o modelo WRFDA e o WRF no supercomputador de sua universidade.

5.4 Prática de execução de rodadas

Durante algumas semanas de setembro, foram executadas algumas rodadas usando os dados descarregados anteriormente, de 2018, com objetivo de sanar algumas falhas de execução relacionadas ao número de pontos de grade das previsões, e ao tempo de condição de contorno.

5.5 Visualização gráfica das previsões em tempo 0 do modelo WRF

Os orientadores desejaram visualizar graficamente os arquivos de saída das previsões executadas com os dados descarregados anteriormente, o que foi feito usando uma ferramenta chamada Panoply. O objetivo foi verificar se os dados abrangiam toda a área de previsão almejada.

5.6 Execução de rodadas de previsão para os meses de junho, julho e agosto

Os orientadores encarregaram os bolsistas Gabriel Chaves e Karliene Pinto de executarem rodadas para os meses de junho, julho e agosto do ano de 2018. Ambos o fizeram em alguns dias, inicialmente de modo mais manual e depois com o auxílio de uma rotina escrita em Bash.

Essa tarefa possibilitou aprendizado de ambos os bolsistas, com relação ao gerenciamento das tarefas de rodadas em filas distintas, e automação. Houve interação entre ambos, para auxílio.

Houve problemas nas execuções de algumas rodadas, pelo foi necessário várias tentativas. Gabriel escreveu rotinas para automatizar a execução das rodadas e limpeza das falhas, e modificou algumas rotinas existentes para seu melhor manuseio no seu ambiente de trabalho.

5.7 Preparo do sistema de assimilação de dados WRFDA

O bolsista teve acesso ao supercomputador Tapajós, de sua universidade, e aproveitou o período de dezembro, enquanto eram preparadas as rotinas de uso do WRFDA pelo seu orientador, para ler a documentação do modelo e executar sua compilação e seu uso com casos de teste disponibilizados pelo servidor do modelo.

5.8 Instalação do ARW e WRFDA no supercomputador Tapajós

O bolsista ficou responsável pela compilação da suíte ARW, que é composta pelo WPS e WRF, bem como do WRFDA, todos no supercomputador Tapajós, em cujo ambiente o bolsista ficou responsável por realizar as rodadas de previsão numérica do tempo e as rodadas de assimilação de dados.

O processo de compilação envolveu um estudo do guia de instalação do ARW e WRFDA, e demandou instruções dos orientadores para a resolução de erros no processo. O processo abrangeu as seguintes etapas:

1. Compilação e teste do módulo WPS;
2. Compilação e teste do módulo WRF;

3. Compilação e teste do módulo WRFDA.

Foi elaborado pelo bolsista um relatório, como um guia, do processo de instalação do ARW e do WRFDA no supercomputador Tapajós, bem como um outro relatório apenas para expor o processo de teste dos módulos, ou seja, suas execuções.

Resultados obtidos e esperados

- Foi realizada com êxito uma rodada de teste do WRF no supercomputador Santos Dumont.
- Estes primeiros passos foram importantes para a familiarização com o ambiente computacional, com o modo de submissão de jobs, e também com o modelo WRF.
- Os passos seguintes na pesquisa consistem em executar rodadas paralelas do WRF em umas das regiões de interesse do projeto, na região norte do país, abrangendo o estado do Pará.

Espera-se ainda, no decorrer do andamento do trabalho, chegar-se aos seguintes resultados finais:

- Novo método de assimilação de dados para o modelo WRF, baseado em aprendizado de máquina, para a região amazônica;
- Os algoritmos de aprendizado de máquina propostos devem garantir a qualidade da análise e apresentar grande ganho no desempenho computacional;
- Uso da metodologia de assimilação por aprendizado de máquina em previsão operacional para a região de interesse.

6) Conclusões

Neste período, teve-se contato com o ambiente virtual do supercomputador Santos Dumont e realizou a execução do processamento de rodadas com o modelo WRF para o período de estudo com recortes de dados do domínio de interesse (região Oeste do Pará) presentes na máquina. O cumprimento desta etapa foi importante como aprendizado para a preparação das rodadas do modelo aos domínios de interesse neste projeto, tais como, a região norte do país, mais especificamente o estado do Pará. Esta etapa de pré-processamento dos dados são cruciais para a aplicação da metodologia de Redes Neurais Artificiais como uma técnica de assimilação de dados para se obter previsões com o Modelo WRF com melhor precisão.

7) Referências Bibliográficas

[1] E. Kalnay, Atmospheric modeling, data assimilation, and predictability, 2nd ed. New York: Cambridge university press, 2003.

[2] Ismail-Zadeh, A. and Tackley, P. J. (2010). Computational Methods for Geodynamics. Cambridge University Press, United States of America, first edition.

[3] Skamarock, W. C., Klemp, J. B., J., D., Gill, D. O., Liu, Z., Berner, J., Wang, W., Powers, J. G., Duda, M. G., Barker, D. M., and Huang, X.-Y. (2019). A description of the advanced research wrf version 4. Technical report, NCAR Technical Notes -NCAR/TN-556+STR, Colorado. USA. DOI: 10.5065/1dfh-6p97.

Desenvolvimento de Programa para Simulação Numérica pelo Método dos Elementos Finitos Clássicos

Bolsista: Gabriel Massi de Moura
Orientador: Eduardo Garcia
Instituição: Laboratório Nacional de Computação Científica

Modalidade da Bolsa: Iniciação Científica
Vigência: 01/12/2023 a 31/08/2024

Petrópolis, 09 de julho de 2024.

Objetivos:

Este projeto tem como principal objetivo a participação do bolsista no desenvolvimento e manutenção de um simulador numérico com método dos elementos finitos. Para isso, reconheceu-se a necessidade de dominar o uso de Sistemas de Controle de Versão, com foco no Git, para trabalhar no simulador através de um repositório. Também foi exigido a introdução do bolsista à análise de elementos finitos, à matemática envolvida e às técnicas de programação necessárias para a implementação do simulador.

Palavras-chave: Método dos elementos finitos, desenvolvimento de software, sistemas de controle de versão, Git.

1. INTRODUÇÃO

O método dos elementos finitos é um método numérico extremamente útil para modelagem computacional de sistemas físicos. Ele consiste em dividir o objeto estudado em várias partes, os chamados elementos finitos, para identificar suas propriedades em cada elemento e então conseguir prever possíveis alterações.

O método começa identificando as equações diferenciais parciais que descrevem o problema analisado, estas devem ser colocadas em sua forma integral, ou “forma fraca”. Como equações integrais são problemas contínuos, para obter um resultado numérico é necessário discretizar as equações. Isto é feito ao dividir o domínio em pequenas partes, os elementos. Cada elemento é formado por uma certa quantidade de nós, pontos onde é possível escrever as mesmas equações, porém na sua forma discreta, através de somatórios. Os elementos podem então ser descritos como o somatório de seus nós, e assim é possível transformar o problema em uma forma matricial, na qual este pode ser resolvido por um solver adequado.

Para entender o método dos elementos finitos e suas aplicações, foram realizados exercícios de programação, com o objetivo de criar um modelo simplificado de uma malha de elementos finitos.

Os VCS (Version Control System) são ferramentas cujo principal objetivo é organizar o fluxo de atualizações em um projeto de software. Isso é feito através da criação de uma estrutura de arquivos em que as modificações dos arquivos são referenciadas de acordo com a ordem em que ocorreram. Os VCS são utilizados amplamente ao redor do mundo para desenvolvimento de software,

e conhecendo suas vantagens, decidimos utilizar o sistema Git para o desenvolvimento do simulador. Para isto, realizamos um estudo de seu funcionamento.

Ao longo do estudo do Git, redigimos um documento a fim de reunir o essencial para o entendimento dos Sistemas de Controle de Versão, além de algumas ferramentas que permitiriam o melhor uso possível do Git para o desenvolvimento do simulador.

2. METODOLOGIA

Devido ao interesse do projeto em reunir informação pertinente e confiável a respeito dos Sistemas de Versionamento, redigimos um documento sintético. Para isso, ao longo dos estudos, discutimos e testamos a aplicabilidade de cada conceito e ferramenta vistos de acordo com o tipo de aplicação que iremos desenvolver (simulador numérico). Entre os assuntos estudados podemos destacar a história e evolução dos Sistemas de Versionamento, o funcionamento e uso do Git, o uso do GitHub, e a implementação de Workflows no desenvolvimento de software com Git.

No que diz respeito ao estudo do método dos elementos finitos, mostrou-se necessário um entendimento da modelagem deste tipo de problema. Por conta disso, foram propostos pelo orientador alguns exercícios de programação, nos quais a modelagem, mesmo que de forma simplificada, se assemelha à dos elementos finitos.

Nesses exercícios, representamos a malha como um array de pontos flutuantes de duas dimensões, nodes[numNP][dim], onde *numNP* representa o total de nós que compõem a malha, e *dim* define as suas dimensões espaciais.

```
int dim = 2;     // dim: numero de dimensões espaciais

int numelX = 8, numelY = 4, numNP = (numelX+1)*(numelY+1);
// numelX/Y: numero de divisões na figura em relação ao eixo X/Y; numNP: numero total de nós.

double nodes[numNP][dim];
```

O objetivo deste exercício é fazer a divisão do objeto em elementos, dando início ao método. Para isso, fizemos uma função que, dados os vértices do objeto, e a quantidade de “cortes” feitos em cada direção, preenchesse o array de nós.

```
div2d(dim, minX, maxX, minY, maxY, numelX, numelY, numNP, nodes);
//minX/Y e maxX/Y indicam os vértices da figura; numelX/Y indicam o número de cortes nos eixos X/Y.

printcoord(npX, npY, nodes); // Imprime na tela o array de nós
```

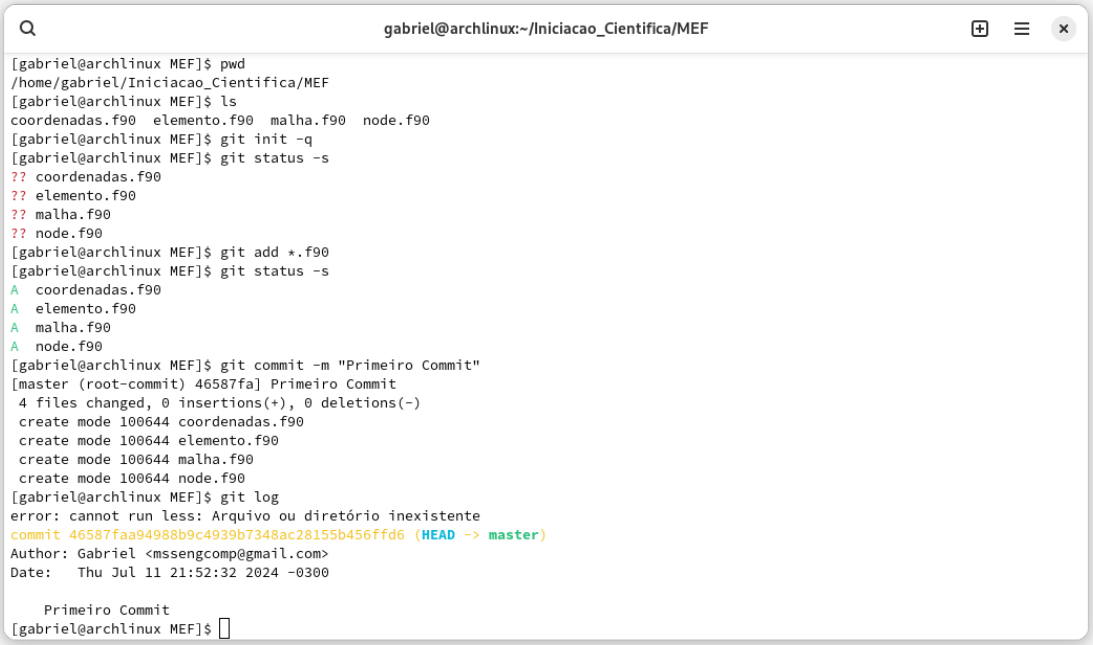
Com os nós calculados, ainda resta identificar os elementos aos quais pertencem, para isso, fizemos mais uma função. Dados o número de nós presente no elemento, o total de elementos esperado, e a referência para o array dos nós, a função preenche o array das conectividades, que guarda os índices dos elementos que os compõem.

```
int nen = 4;     //nen: numero de nós por elemento.
double conect[numelX*numelY][nen];
conect2d(nen, numelX*numelY, nodes, conect);
printconect(numelX*numelY, conect); // Imprime na tela o array de elementos
```

3. RESULTADOS

Através do estudo realizado, conseguimos redigir um documento que aborda os tópicos mais relevantes para o uso do Git, e julgamos que a leitura desse documento pode auxiliar a introduzir o versionamento de forma eficaz em projetos de software.

Visitamos de forma breve a história dos Sistemas de Versionamento, com o intuito de entender as demandas que os sistemas mais usados tiveram de suprir, até chegarmos ao Git. Em relação ao Git, descrevemos de forma rápida e de fácil leitura o seu funcionamento, com exemplos dos comandos necessários.



[IMAGEM 01] - Exemplo de uso do Git.

Apresentamos também o GitHub, em um conteúdo breve, expondo suas principais características. Por fim, abordamos os Workflows voltados para versionamento, Gitflow e Integration-Manager, com destaque para o Gitflow, que julgamos ser o mais interessante para o desenvolvimento do simulador.

Sendo o principal recurso matemático do simulador, o método dos elementos finitos deveria ser compreendido com certa profundidade. Implementamos as funções propostas nos exercícios, de modo a obtermos uma lista de elementos e as coordenadas de seus nós. A partir desses dados foi possível compreender a modelagem dos elementos finitos, e já podemos planejar aspectos fundamentais do simulador.

```
gabriel@archlinux:~/Iniciacao_Cientifica/Praticas
[gabriel@archlinux Praticas]$ gcc -o div2d div2d.c
[gabriel@archlinux Praticas]$ ./div2d

Coordenadas dos nós:

[0.000000 0.000000] [1.000000 0.000000] [2.000000 0.000000] [3.000000 0.000000] [4.000000 0.000000] [5.000000 0.000000] [6.000000 0.000000]

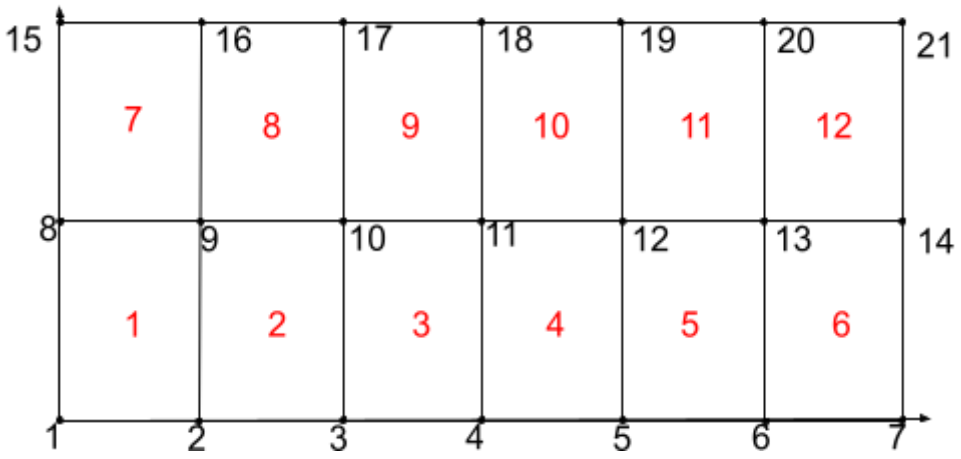
[0.000000 1.000000] [1.000000 1.000000] [2.000000 1.000000] [3.000000 1.000000] [4.000000 1.000000] [5.000000 1.000000] [6.000000 1.000000]

[0.000000 2.000000] [1.000000 2.000000] [2.000000 2.000000] [3.000000 2.000000] [4.000000 2.000000] [5.000000 2.000000] [6.000000 2.000000]


Elementos e o índice de seus nós:

0: 0 1 8 7
1: 1 2 9 8
2: 2 3 10 9
3: 3 4 11 10
4: 4 5 12 11
5: 5 6 13 12
6: 6 7 14 13
7: 8 9 16 15
8: 9 10 17 16
9: 10 11 18 17
10: 11 12 19 18
11: 12 13 20 19
[gabriel@archlinux Praticas]$
```

[IMAGEM 02] - Saída das funções propostas.



[IMAGEM 03] - Malha representada.

4. CONCLUSÃO

O projeto até o momento consistiu em preparações para o desenvolvimento do simulador, visto que não havia, no início, conhecimento suficiente dos conteúdos necessários para fazê-lo. Ao término da vigência desta bolsa, é possível dizer que adquirimos familiaridade suficiente para dar sequência de forma produtiva, e assim esperamos alcançar uma primeira versão do simulador.

REFERÊNCIAS BIBLIOGRÁFICAS:

[1] Scott Chacon & Ben Straub. PRO GIT, Everything you need to know about Git. 2. ed. Apress, 2014.

[2] Ajay Harish. Finite Element Method – What Is It? FEM and FEA Explained. SymScale Blog, 2024.

RELATÓRIO DE ATIVIDADES

Título do Projeto: Análise da aplicação BEAST 1.10 no ambiente nó Sequana do SDumont

Nome do bolsista: Guilherme Freire da Silva Dornelas

Nome do orientador: D.Sc. Carla Osthoff Ferreira de Barros

Nome do coorientador: D.Sc. Kary Ann del Carmen Ocaña Gautherot / Micaella Coelho

Tipo de bolsa: PIBIC

Período do relatório: 01/09/2023 a 31/08/2024

INTRODUÇÃO

O estudo analisou o desempenho da aplicação BEAST 1.10, integrada à biblioteca BEAGLE 3, em ambientes de computação de alto desempenho (CAD). Os dados genômicos foram processados nos nós *multithreads* e GPU do supercomputador Santos Dumont, SDumont. O objetivo foi avaliar o desempenho e a escalabilidade do BEAST 1.10 no ambiente computacional do SDumont, visando otimizar a alocação de recursos e sua utilização eficiente. Exploramos o impacto do tamanho dos dados genômicos, a influência dos modelos evolutivos e o parâmetro *chainLength* em diferentes configurações de filas, *threads* e nós do SDumont.

O BEAST é um programa multiplataforma para análise Bayesiana de sequências moleculares. Unifica a reconstrução filogenética molecular com evolução complexa de características discretas e contínuas, datação por tempo de divergência e modelos demográficos coalescentes em um mecanismo de inferência estatística eficiente usando integração MCMC. Em estatística, os métodos MCMC compreendem uma classe de algoritmos para amostragem de uma distribuição de probabilidade.

OBJETIVOS

- O projeto visa desenvolver estudos de desempenho da aplicação BEAST, de forma a obter a melhor alocação de recursos computacionais. Analisando o desempenho da aplicação de filogenômica e evolução molecular computacional em GPU, no ambiente de computação de alto desempenho (CAD), mais especificamente nos recursos computacionais do SDumont.
- Estabelecer análise comparativa do BEAST e a BEAGLE 3, nos ambientes CPU e GPU.
- Realizar análises de desempenho e escalabilidade, visando otimizar o uso do ambiente computacional.

CONFIGURAÇÃO DOS DADOS

Aplicados de 3 conjuntos de dados com diferentes tamanhos. As execuções foram realizadas através do *software* BEAST 1.10, integrado à BEAGLE 3. Os ambientes selecionados são de CPU e GPU. Utilizando o parâmetro *chainLength* nas execuções. O *chainLength* especifica o número de etapas que a cadeia MCMC fará antes de terminar. Esse número depende do tamanho do conjunto de dados, da complexidade do modelo e da precisão da resposta necessária.

Os dados biológicos pertencem aos genomas do vírus do Dengue, do tipo DENV1, com o genoma humano como hospedeiro, foram obtidos em formato FASTA das bases de dados do Sistema *Bacterial and Viral Bioinformatics Resource Center* (BV-BRC) e do *National Center for Biotechnology Information* (NCBI). Os genomas em formato FASTA foram alinhados com o *software*

de alinhamento múltiplo MAFFT. Os alinhamentos em formato FASTA foram convertidos ao formato (XML), com *software* BEAUti v1.8.4. Esses conjuntos de dados foram chamados:

1)"DENV1_World" composto com 2.624 genomas completos de todos os países (https://drive.google.com/file/d/1nHfNzs6W1xJKZFoqfaO6zK4q7zZItJ-H/view?usp=drive_link);
2)"DENV1_SouthAmerica" composto com 93 genomas completos dos países de América do Sul (https://drive.google.com/file/d/1--NspBhc0c89FqZkgbzQt0GvAvRH7N5r/view?usp=drive_link);
3)"DENV1_Brazil" composto com 43 genomas completos do Brasil (https://drive.google.com/file/d/1eI6TwEP_4expsR6iBAqJ5ohmxMLHomPv/view?usp=drive_link).

A Tabela 1 apresenta informações referente ao conjunto de dados, composto por "DENV1_World", "DENV1_SouthAmerica" e "DENV1_Brazil", esses dados são do Vírus da Dengue do sorotipo (DENV1), os dados genômicos estão relacionados à estrutura e função do genoma de um organismo. A Taxa de Genomas (*Táxon*) refere-se ao código genético de um organismo, da sequência de DNA. O Alinhamento de *Sites* é organizado através de pares ou entre múltiplas sequências, sendo representados como globais e locais. Nos tamanhos dos arquivos, o maior dado é o genoma "DENV1_World" (27 MB). O formato de dados do (DENV1) é de Nucleotídeos.

Conjunto de dados	Taxa (Genomas) número*	Alinhamento *Sites***	Tamanho Arquivo	Formato Dados
DENV1_World - Genoma	2624	11.195	27 MB	Nucleotídeos
DENV1_SouthAmerica - Genoma	93	10.954	0,99 MB	Nucleotídeos
DENV1_Brazil - Genoma	43	10.917	458 KB	Nucleotídeos

* *Táxon*. Um grupo taxonômico de organismos. Neste código, os táxons podem ser organismos ou espécies.

** *Site*. Posição individual discreta, normalmente um único nucleotídeo (ou aminoácido).

Tabela 1. Informações dos dados genômicos do vírus da Dengue

CONFIGURAÇÃO DO EXPERIMENTO 1

As execuções com o nó Sequana do SDumont. Cada dado passou por 4 execuções no ambiente computacional de CPU e GPU. Buscando o melhor uso de threads e utilização do ambiente computacional do SDumont. Os experimentos realizados com o *software* BEAST 1.10, como entrada 3 conjuntos de dados (XML): "DENV1_World" (27 MB), "DENV1_SouthAmerica" (0,99 MB), "DENV1_Brazil" (458 KB). Os parâmetros foram Beagle 3, *chainLength* "100000" e "20000000" e ambiente CPU e GPU.

O ambiente CPU, composto por 2x Intel Xeon Cascade Lake Gold 6252, com 48 núcleos (24 *cores* por CPU), 384 GB de memória RAM. O ambiente de GPU, 384 GB de memória RAM, 4x *NVIDIA* Volta V100 GPU. As execuções variando o número de *threads* (1, 4, 8, 12, 24, 32 e 48). As filas se caracterizam por (*sequana_cpu* e *sequana_gpu*).

ANÁLISES DE DESEMPENHO

Esta seção apresenta os resultados dos experimentos usando o *chainLength* "100000" e "20,000000", executando em (1, 4, 8, 12, 24, 32 e 48) *threads*, no ambiente CPU e as execuções com *chainLength* "100000" e "20,000000" no ambiente GPU. Ambos os testes, com os arquivos "DENV1_World" (W), "DENV1_SouthAmerica" (SA), "DENV1_Brazil" (BR).

A Figura 1 apresenta os resultados de tempo total de execução (TTE) com o *chainLength* "100000", com CPU, usando (1, 4, 8, 12, 24, 32 e 48) *threads*. O objetivo foi determinar em cada arquivo o uso adequado de *threads* e recursos computacionais. Obtemos o menor (TTE) com 8 *threads*. O conjunto de genoma "SA" o TTE foi 0,20 e "BR" 0,26, esses arquivos podem ser considerados de tamanho pequeno. O conjunto "W" que pode ser considerado de tamanho grande, o TTE foi de 15,49 minutos.

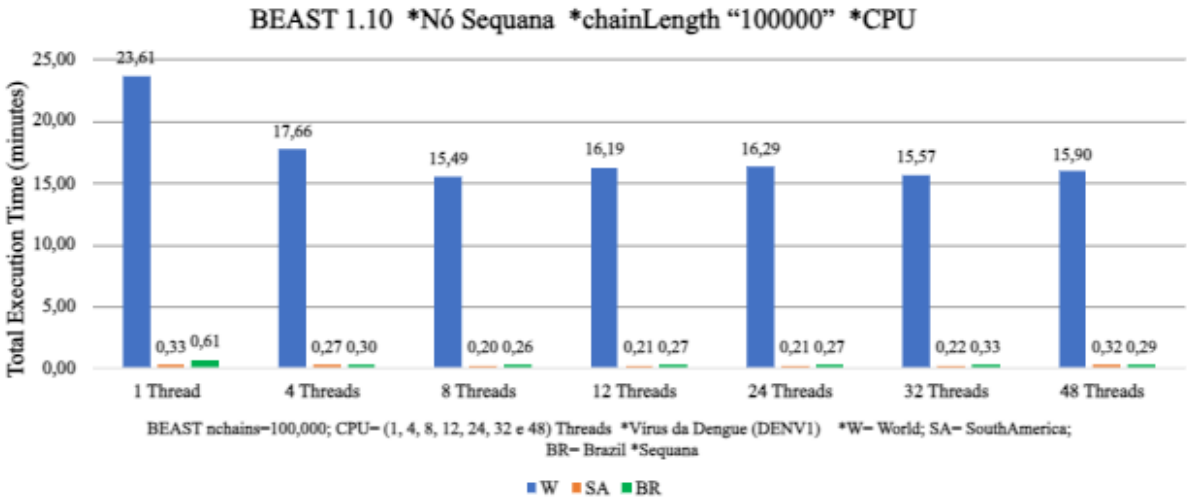


Figura 1. TTE DENV1 CPU (1, 4, 8, 12, 24, 32 e 48) *threads* e *chainlength* "100,000" *Sequana

A Figura 2 apresenta os resultados de tempo total de execução (TTE) com o *chainLength* "100000", executado em GPU. O objetivo foi determinar em cada arquivo o uso adequado de recursos computacionais do nó Sequana do SDumont. Os conjuntos de genomas "SA" o TTE 0,18 e "BR" 0,22. O conjunto de genoma "W" o TTE foi 4,01 minutos, conseguimos um melhor desempenho.

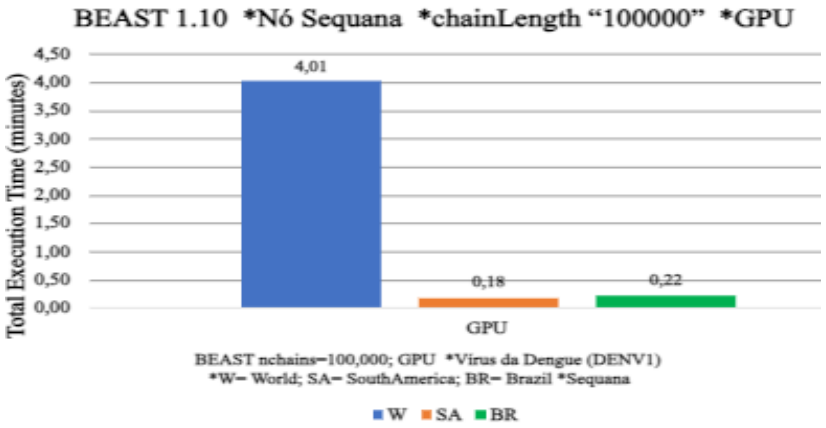


Figura 2. TTE DENV1 - GPU e *chainlength* "100,000" *Sequana

A Figura 3 apresenta os resultados de tempo total de execução (TTE) com o *chainLength* "20,000000", executado em CPU, usando 1, 4, 8, 12, 24, 32 e 48 *threads*. O objetivo foi determinar para cada arquivo qual o melhor uso de *threads* e recursos computacionais do nó Sequana do SDumont. Obtemos o menor tempo total de execução com 8 *threads*, os conjuntos de genoma "SA" o TTE 37,03 minutos e "BR" 50,21 minutos. O conjunto "W", o TTE de 1440,96 minutos.

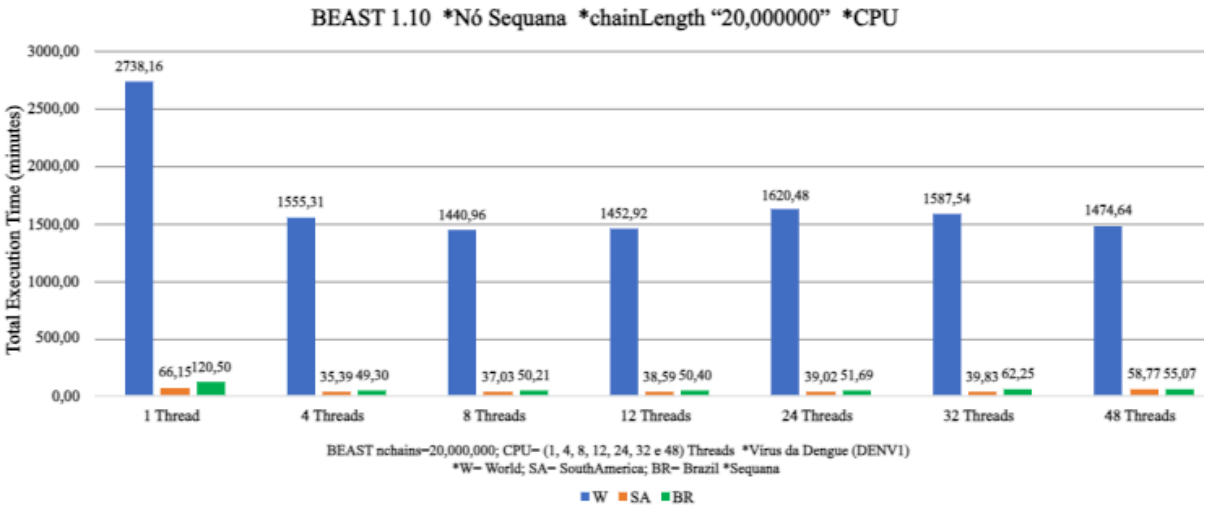


Figura 3. TTE DENV1 CPU (1, 4, 8, 12, 24, 32 e 48) *threads* e *chainlength* "20,000000" *Sequana

A Figura 4 apresenta os resultados de tempo total de execução (TTE) com o *chainLength* "20,000000", executado em GPU. O objetivo foi determinar para cada arquivo qual o melhor uso de recursos computacionais do nó Sequana do SDumont. O conjunto de genoma "SA" o TTE foi 32,80 e "BR" 38,93. O conjunto de genoma "W", o TTE foi de 660,95 minutos, conseguimos um melhor desempenho.

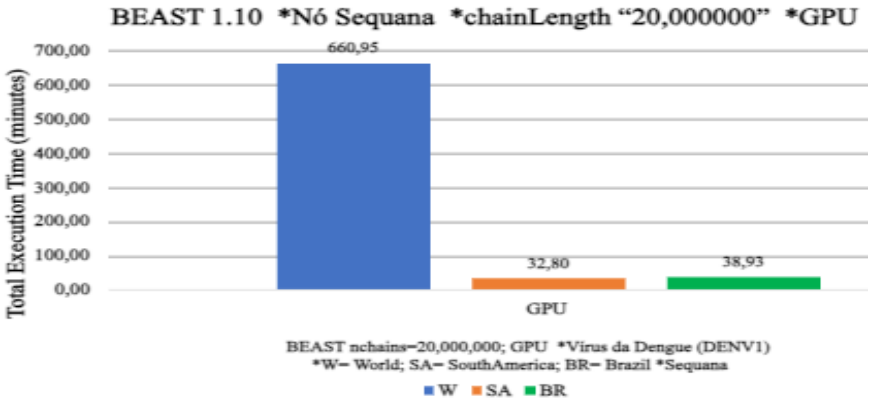


Figura 4. TTE DENV1 - GPU e *chainlength* "20,000000" Nó Sequana

CONFIGURAÇÃO DO EXPERIMENTO 2

As execuções com o nó Sequana e com modelos evolutivos. Cada dado passou por 4 execuções no ambiente computacional de CPU e GPU. Buscando o uso adequado do ambiente computacional do SDumont. Os experimentos com o BEAST 1.10, utilizando os conjuntos de dados (XML):"DENV1_World"(W)(27MB),"DENV1_SouthAmerica"(SA)(0,99MB),"DENV1_Brazil"(BR) (458 KB). O objetivo foi analisar o comportamento do BEAST 1.10 com variação de modelos evolutivos. Utilizamos a Beagle 3, o parâmetro *chainLength* de "100000" e "20000000".

O ambiente CPU, composto por 2x Intel Xeon Cascade Lake Gold 6252, com 48 núcleos (24 *cores* por CPU), 384 GB de memória RAM. O ambiente de GPU, 384 GB de memória RAM, 4x *NVIDIA* Volta V100 GPU. Optamos por execuções com 8 *threads*, pois nos testes anteriores foi o menor (TTE). Variação dos modelos evolutivos (JC, GTR, HKY e TN93). Nas análises as filas (*sequana_cpu* e *sequana_gpu*).

ANÁLISES DE DESEMPENHO

A Figura 5 apresenta os resultados de tempo total de execução (TTE) com o *chainLength* "100000", executado em CPU, com 8 *threads*. A finalidade das execuções foi avaliar os modelos evolutivos, no ambiente nó Sequana e buscar o uso mais adequado de recursos computacionais. O menor (TTE) do conjunto "W", que pode ser considerado como de tamanho grande, foi com o modelo (JC), o TTE de 9,70 minutos. Os conjuntos "SA" com o modelo (TN93) o TTE 0,17 e o "BR"com o modelo (HKY) o TTE de 0,26.

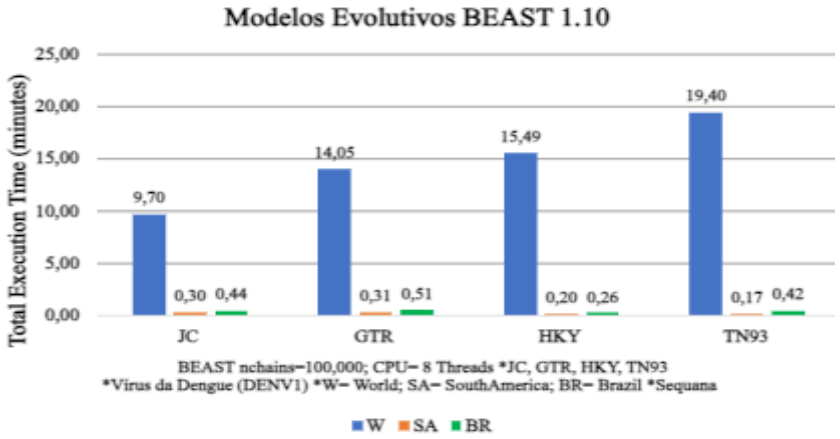


Figura 5. TTE DENV1 *CPU 8 *threads chainlength* "100,000" *Nó Sequana *JC, GTR, HKY, TN93

A Figura 6 apresenta os resultados de tempo total de execução (TTE) com o *chainLength* "100000", executado em GPU. A finalidade das execuções foi avaliar os modelos evolutivos, no ambiente nó Sequana e buscar o uso mais adequado de recursos computacionais. O (TTE) foi com modelo (JC), o conjunto "W", o TTE foi de 2,86 minutos. Os conjuntos "SA" o TTE foi 0,15 e "BR" o TTE de 0,18.

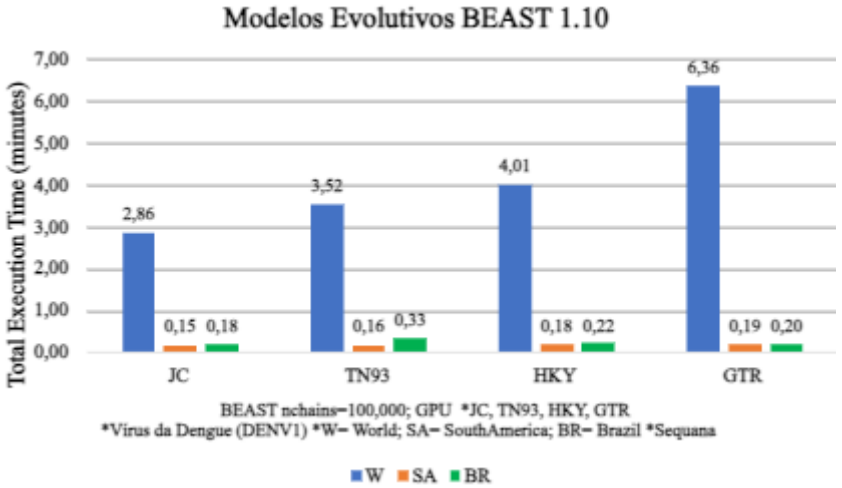


Figura 6. TTE DENV1 *GPU - *chainlength* "100,000" *Nó Sequana *JC, TN93, HKY e GTR

A Figura 7 apresenta os resultados de tempo total de execução (TTE) com o *chainLength* "20,000000", executado em CPU, com 8 *threads*. A finalidade das execuções foi avaliar os modelos evolutivos, no ambiente nó Sequana e buscar o uso mais adequado de recursos computacionais. O menor (TTE) do conjunto "W", foi no modelo (JC) o TTE de 1196,58 minutos e o "SA" o TTE foi 31,82 minutos. O conjunto "BR" com o modelo (HKY) o TTE de 50,21 minutos.

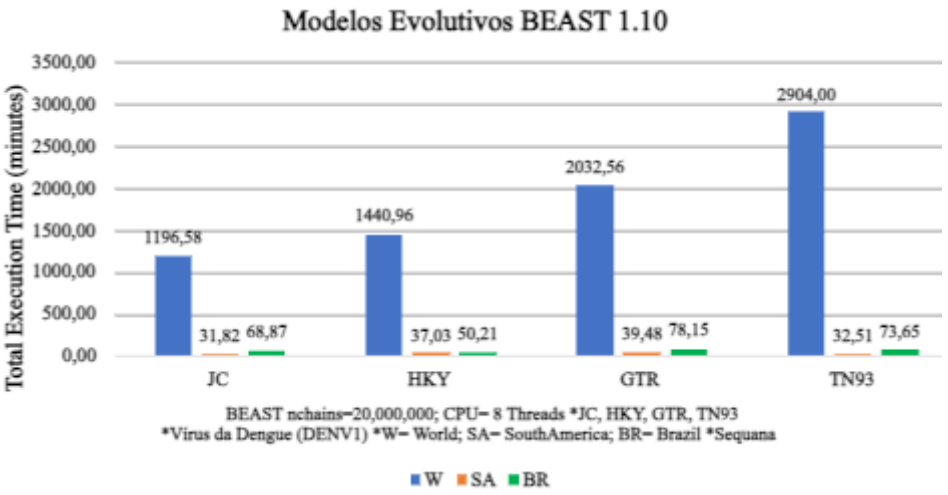


Figura 7. TTE DENV1 *CPU 8 *threads* e *chainlength* "20000000" *Nó Sequana *JC, HKY, GTR e TN93

A Figura 8 apresenta os resultados de tempo total de execução (TTE) com o *chainLength* "20,000000", executado em GPU. A finalidade das execuções foi avaliar os modelos evolutivos, no ambiente nó Sequana e buscar o uso mais adequado de recursos computacionais. O menor (TTE) foi com modelo (JC), o conjunto "W", o TTE foi de 445,68 minutos. Os conjuntos "SA" o TTE foi 28,57 minutos e "BR" o TTE de 31,88 minutos.

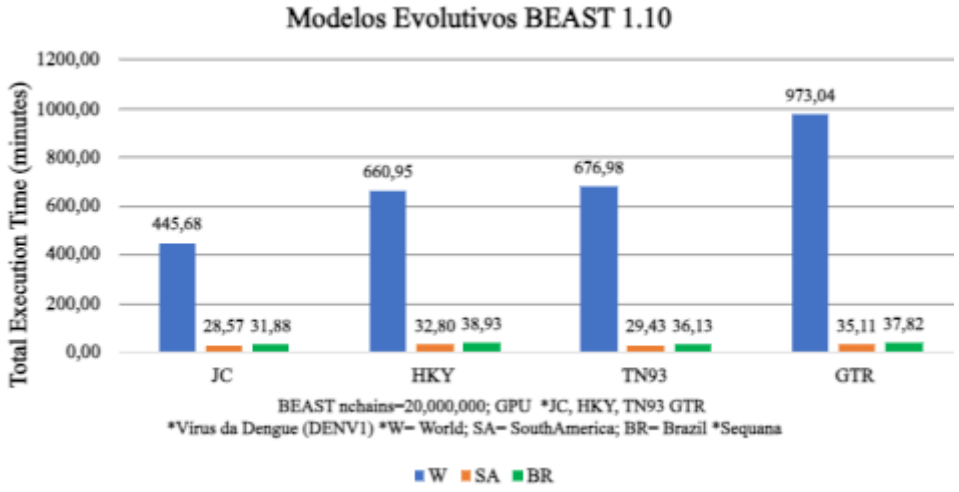


Figura 8. TTE DENV1 *GPU - *chainlength* "20000000" *Nó Sequana *JC, HKY, TN93 e GTR

CONCLUSÃO

As execuções do BEAST 1.10 e a BEAGLE 3, com variabilidade no número de *chainLength* de "100000" e "20,000000", nas configurações de CPU (*Cascade Lake*) e GPU (*Nvidia* V100) do nó Sequana, sugerem o uso da configuração GPU apresentando o menor tempo de execução. Podemos concluir que a configuração computacional mais eficiente no ambiente CPU é com 8 *threads* e para arquivos considerados grandes o uso de GPU, no ambiente do nó Sequana.

Nas análises de desempenho do BEAST 1.10, com variabilidade dos modelos evolutivos e no parâmetro de *chainLength* de"100000" e "20,000000", nas configurações de CPU (*Cascade Lake*) e GPU (*Nvidia* V100) do nó Sequana do SDumont, sugerem o uso da configuração GPU, com o menor tempo de execução. O ambiente computacional mais eficiente é com uso do modelo "JC" e para arquivos grandes é recomendado no ambiente do nó Sequana, a utilização de GPU (*Nvidia* V100).

REFERÊNCIAS

- Ocaña KA, Dávila AM. Phylogenomics-based reconstruction of protozoan species tree. Evol Bioinform Online. 2011; 7:107-21. doi: 10.4137/EBO.S6861. Epub 2011 Jul 31. PMID: 21863127; PMCID: PMC3153122.
- 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.
- ERAD RJ - Exploração de Módulos Paralelo Híbrido de Bioinformática para Ambientes GPU de Supercomputação - 1 de dezembro de 2020 Autores: G Dornelas, M Coelho, K Ocaña, C Osthoff.
- Ocaña K, Coelho M, Terra R, Freire G, Santos M, Cruz L, Galheigo M, Carneiro A, Fagundes B, Carvalho D, Cardoso D, Meneses E, Gadelha L, Osthoff C, Developing Efficient Scientific Gateways for Bioinformatics in Supercomputer Environments Supported by Artificial Intelligence. In: (ISC High Performance 2021).
- Coelho, Micaella et al. Desenvolvimento de um Framework de Aprendizado de Máquina no Apoio a Gateways Científicos Verdes, Inteligentes e Eficientes: BioinfoPortal como Caso de Estudo Brasileiro. *In*: Simpósio em Sistemas Computacionais de Alto Desempenho (WSCAD), 23., 2022, Florianópolis/SC. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2022. p. 205-216. DOI: https://doi.org/10.5753/wscad.2022.226377.

RELATÓRIO DE ATIVIDADES - BOLSA PIBIC/LNCC

1) Dados Gerais

Título do projeto: Metodologia de auditoria de código e planejamento de otimização aplicada no núcleo dinâmico do modelo MONAN
Bolsista: Isabel de Freitas Barboza
Orientadores: Roberto Pinto Souto, Eduardo Lucio Mendes Garcia
Tipo de bolsa: PIBIC
Período do relatório: março - julho de 2024
Início da Bolsa: maio 2023

2) Objetivos

Objetivos abordados:

1. Preparação para o progresso do projeto com o estudo aprofundado de métricas de desempenho e das ferramentas de análise de desempenho escolhidas.
2. Estudos e análises com aplicação MPAS que se adere a proposta da pesquisa.
3. Introdução às métricas do POP-COE

3) Introdução

Modelos de previsão numérica de tempo (PNT), devido à grande quantidade de cálculos que são realizados durante a sua execução, são aplicações que demandam o uso de computação de alto desempenho. Está em desenvolvimento um novo modelo comunitário unificado do sistema terrestre – o **MONAN** (**M**odel for **O**cean-**La**N**d**-**A**tmosphere predictio**N**), Modelo para Previsão dos Oceanos, Superfícies Terrestres e Atmosfera (na sigla em português), para produzir previsões com ênfase na região tropical e foco sobre a América do Sul, em diferentes escalas espaciais e de tempo, buscando incluir as necessidades dos setores produtivo e social, que substituirá os atuais modelos atmosféricos atualmente em operação. O MONAN é um programa institucional do MCTI, coordenado pelo Instituto Nacional de Pesquisas Espaciais (INPE) e, nesta fase inicial do projeto, o LNCC está participando no desenvolvimento computacional, auxiliando na avaliação de desempenho dos códigos candidatos ao núcleo da dinâmica do modelo. Para auxílio a esta finalidade, é necessário um bom conhecimento de linguagem de programação a fim de bem entender como foi realizada a implementação dos modelos numéricos. No que diz respeito ao desempenho computacional dos modelos, também é de grande importância o entendimento da estratégia de paralelização empregada nos seus códigos fonte. Portanto, este relatório contém os estudos e resultados das análises de desempenho do MPAS (**Model for Prediction Across Scales), Modelo Para Previsão Em Escalas.** Um estudo preliminar de desempenho paralelo desta aplicação que está bem próxima do que é o modelo **MONAN**. Também contém o estudo e demonstração da utilização de metodologia e ferramentas de avaliação de desempenho já empregadas com sucesso no **"Performance Optimisation and Productivity Centre of Excellence in Computing Applications"** (**POP-COE**: https://pop-coe.eu), ou Centro de Excelência de Otimização de Desempenho e Produtividade em Aplicações Computacionais, numa tradução livre.

4) Material e Métodos

A metodologia do trabalho foi composta por três etapas principais:

1. **Estudo do MPAS**: Iniciou-se com o estudo do MPAS, um modelo atmosférico desenvolvido pelo NCAR e LANL, utilizado para previsões de tempo e clima em diversas escalas espaciais.

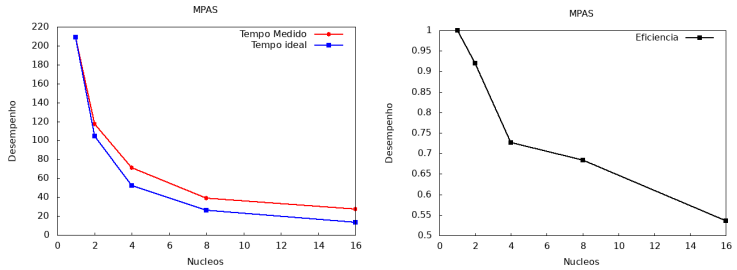


Figura 1 - A e B - Resultados do MPAS 120Km obtidos no Sdumnont

MPAS Atmosphere
O componente atmosférico do MPAS utiliza uma malha de Voronoi centroidal não estruturada e escalonamento de grau C. Uma malha hexagonal que pode ser ajustada para ter resolução maior ou menor em diferentes regiões.

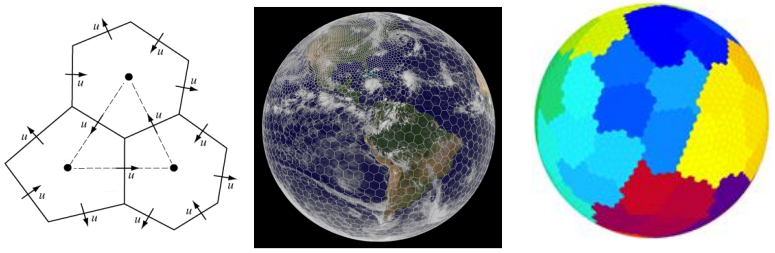


Figura 2 - A - Malha de Voronoi , B Resolução no globo (Fonte : https://mpas-dev.github.io/) , C -Resolução no globo com a divisão entre processos (Fonte : SC23_SCC_MPAS-A-Webinar, 2023)

Isso permite que executemos um modelo global em escalas de resolução de tempestades onde for mais importante e com resolução mais baixa em todos os outros lugares para conservar os recursos computacionais.

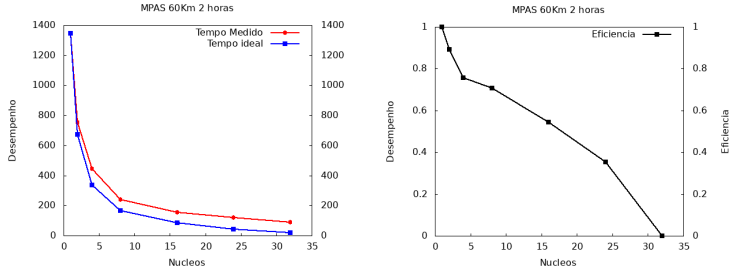


Figura 3 - A e B - Resultados do MPAS 60Km obtidos no Sdumnont

2. **Análise de desempenho com Scalasca e HPCToolkit**: Em seguida, aprofundou-se nas ferramentas Scalasca e HPCToolkit para analisar custos de execução, ineficiências e escalabilidade de programas paralelos, utilizando as ferramentas de visualização CubeGUI e HPCviewer.
3. **Estudo das métricas do POP-COE**: Por fim, foram estudadas as métricas do POP-COE, que quantificam o impacto dos fatores de paralelização, permitindo comparar o desempenho paralelo e identificar ineficiências nos códigos.

5) Resultados e Discussão

Análise de um resultado do MPAS com HPCtoolkit no HPCviewer

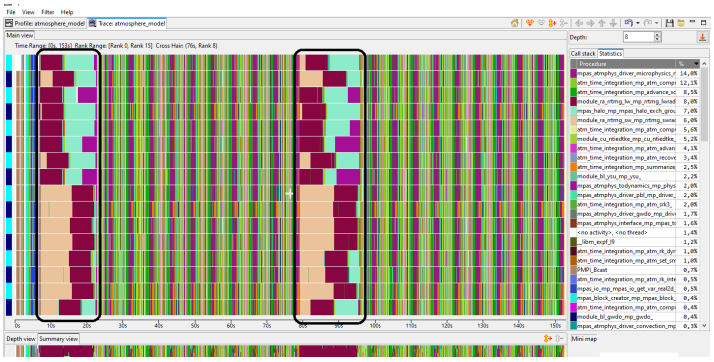
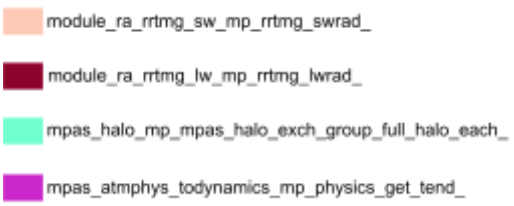


Figura 4- MPAS com 16 processos em 60km em 2 horas

A imagem acima contém o resultado da execução do MPAS com 16 processos em 60km em 2 horas. Olhando esta imagem, vemos dois trechos evidenciados. Para entendermos, primeiro precisamos compreender que a imagem mostra uma linha do tempo (perfil de execução do mpas) que detalha a atividade dos processos ao longo do tempo. O trecho marcado possui uma diferença do tempo gasto nas funções nos diferentes processos. Essas são as Funções:



Possíveis Razões para as Diferenças Observadas:

- Diferenças nas Cargas de Trabalho: A distribuição de trabalho pode não estar perfeitamente equilibrada entre os processos. Alguns processos podem ter subdomínios mais complexos ou mais pesados (por exemplo, mais nuvens ou terrenos mais variados), resultando em tempos diferentes.

- Interações entre Física e Dinâmica: Certas funções de física atmosférica podem ser mais computacionalmente intensivas e afetar apenas partes específicas da simulação (por exemplo, áreas com mais radiação solar durante o dia). Isso cria variações no uso de CPU.

Observando a imagem e a distribuição das cores entre os diferentes processos, podemos tirar as seguintes conclusões:

Função de Radiação de Ondas Curtas (module_ra_rrtmg_sw_mp_rrtmg_swrad_):

> A radiação de ondas curtas é uma energia radiante produzida pelo sol com comprimentos de onda que variam do infravermelho ao visível e ao ultravioleta. A radiação de ondas curtas é, portanto, exclusivamente associada às horas de luz do dia para um local específico na superfície da Terra. ("Shortwave Radiation." *www.hec.usace.army.mil*).

- Representada pela cor rosa claro (salmão)
- As diferenças na carga de trabalho podem ser causadas por variações na intensidade da radiação solar em diferentes partes do modelo.

Função de Radiação de Ondas Longas (module_ra_rrtmg_lw_mp_rrtmg_lwrad_):

> Radiação de onda longa se refere à emissão contínua de ondas eletromagnéticas pela superfície da Terra, principalmente na forma de radiação infravermelha, que é absorvida e reemitida por gases de efeito estufa na atmosfera. ("Long Wave Radiation" www.sciencedirect.com, 2013).

- Representada pela cor vermelha.
- Esta função apresenta variações entre os processos, mas parece estar mais distribuída uniformemente em comparação com a radiação de ondas curtas.
- Radiação de ondas longas está presente tanto de dia quanto de noite, o que pode explicar uma distribuição mais uniforme.

Função de Troca de Halo - Comunicação entre processos (mpas_halo_mp_mpas_halo_exch_group_full_halo_each_):

- Representada pela cor azul.
- A função de troca de halo mostra grandes blocos de execução em alguns processos, indicando períodos de comunicação intensa.
- Processos com maior necessidade de troca de halo podem ter mais tempo gasto nesta função, refletindo a complexidade da comunicação de dados entre subdomínios.

Função de Tendências Físicas (mpas_atmphys_todynamics_mp_physics_get_tend_):

- Representada pela cor rosa.
- Essa função tem períodos de execução menores, mas ainda varia entre os processos.
- As diferenças podem ser causadas pela complexidade dos cálculos físicos em diferentes subdomínios.

Considerações sobre Dia e Noite:

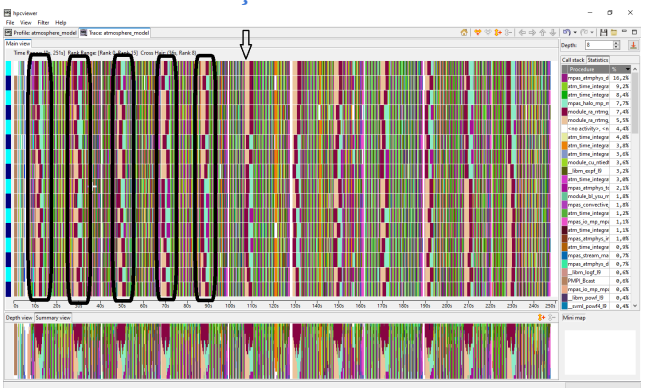


Figura 5 - MPAS com 16 processos em 120km em 24 horas

Com o resultado da imagem acima podemos ter uma melhor noção da simulação em relação a noite e dia pois temos uma simulação de 24 horas do perfil de execução do mpas.

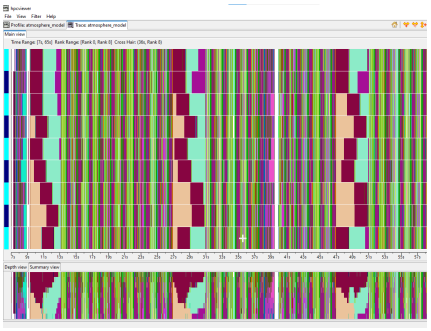


Figura 6 - Imagem ampliada

Agora as diferenças observadas anteriormente também podem ser explicadas pelo Ciclo Diurno

Durante o Dia:

- A função de radiação de ondas curtas será mais ativa, pois a radiação solar direta influencia o aquecimento da superfície terrestre e da atmosfera.

- A função de radiação de ondas longas continua ativa, contabilizando a emissão térmica da Terra e da atmosfera.

Durante a Noite:

- A função de radiação de ondas curtas (`rrtmg_sw_mp_rrtmg_swrad_`) terá pouca ou nenhuma contribuição, pois não há radiação solar direta.

- A função de radiação de ondas longas (`rrtmg_lw_mp_rrtmg_lwrad_`) torna-se dominante no balanço radiativo, representando a perda de calor da superfície terrestre e da atmosfera para o espaço.

Resultado do MPAS com Scalasca no CubeGUI

O espaço de desempenho do Cube é estruturado em três dimensões:

- **Dimensão Métrica**: Contém métricas como tempo de comunicação e perdas de cache.
- **Dimensão do Programa**: Inclui a árvore de chamadas do programa, mapeando os valores das métricas.
- **Dimensão do Sistema**: Contém os componentes executados em paralelo, como processos ou threads, dependendo do modelo de programação paralela.

Ele utiliza três navegadores de árvore acoplados, permitindo visualizar uma dimensão em relação a outra.

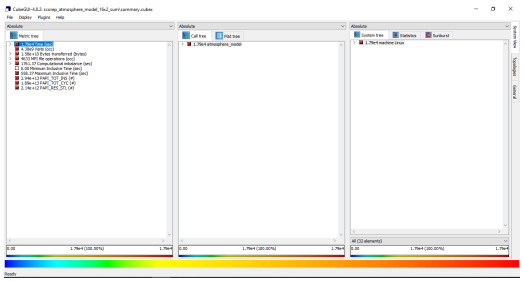


Figura 7 - Dimensão métrica, Dimensão do programa e Dimensão do sistema

Os nós na árvore de métrica representam **métricas**. Os nós na dimensão do programa da figura, representam **caminhos de chamadas** formando uma **árvore de chamadas**. Os nós na dimensão do sistema representam **máquinas, nós, processos ou threads** de cima para baixo.

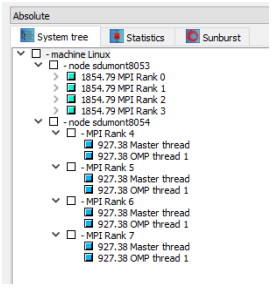


Figura 8 - MPAS com 2 nós 8 processos MPI e 2 threads

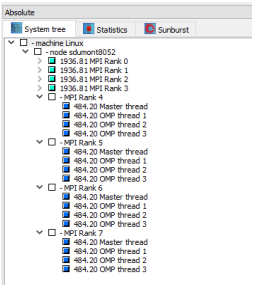


Figura 9 - MPAS com 8 processos MPI e 4 thread

Conseguimos ver detalhadamente, nas imagens, as divisões do tempo entre os processos assim como as respectivas threads de cada processo. A soma dos tempos das threads corresponde ao tempo do processo e a soma dos processos corresponde ao tempo do nó .

Resultados Métricas POP-COE no CubeGUI

As métricas POP são calculadas como eficiências variando de 0 a 1, onde valores mais altos são melhores. Em geral, eficiências acima de 0,8 são aceitáveis, enquanto valores mais baixos indicam problemas de desempenho que necessitam ser explorados. Com os resultados obtidos foi possível visualizar essas eficiências na ferramenta CubeGUI.

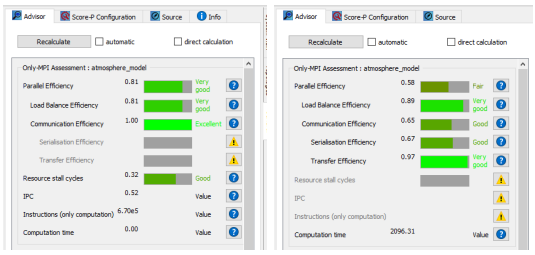


Figura 10 - A e B - Visualização na ferramenta CubeGUI

- **Imagem 1:** A aplicação apresenta uma boa **eficiência global**, especialmente na **comunicação**, mas com espaço para melhorias no **balanceamento de carga** e na espera por recursos.
- **Imagem 2:** A **eficiência paralela** é menor devido a problemas de **comunicação** e **serialização**, apesar do bom **balanceamento de carga**. O tempo de **computação** elevado indica uma redução na **eficiência global.**

Global Efficiency - Eficiência global: Descreve o quão bem a paralelização da aplicação está funcionando.

A **Eficiência de Computação** : Mede o desempenho da carga computacional com diferentes números de processos. Calculada comparando o tempo total gasto em computação. Em uma aplicação ideal, a CE é 1, indicando um tempo de computação constante.

Parallel Efficiency - Eficiência paralela: Avalia a execução paralela do código. Calculada como o produto da Eficiência de Balanceamento de Carga (LB) e da Eficiência de Comunicação (CommE).

Load Balance Efficiency - Eficiência de balanceamento de carga: Reflete a distribuição de trabalho entre processos. Calculada pela razão entre o tempo médio e o tempo máximo gasto em computação.

Communication Efficiency - Eficiência de comunicação : Avalia o impacto da comunicação entre processos. Calculada como o produto da Eficiência de Serialização (SerE) e da Eficiência de Transferência (TE).

Eficiência de Serialização (SerE): Descreve a perda de eficiência devido a dependências entre processos.

Eficiência de Transferência (TE): Mede ineficiências devido ao tempo de transferência de dados.

6) Conclusões

As principais conclusões obtidas no decurso do trabalho realizado destacam

1. A importância dos conceitos iniciais da computação;
2. A compreensão dos conceitos de linguagem, comandos e ambiente de trabalho;
3. O uso adequado das ferramentas disponíveis como as ferramentas de análise de desempenho escolhidas;
4. A análise detalhada dos processos e threads;
5. Observação de ganho de desempenho com paralelização da aplicação MPAS;

Essas conclusões fornecem uma boa estrutura para o prosseguimento do projeto, permitindo avançar com confiança e eficiência nas etapas subsequentes.

7) Referências Bibliográficas

MELLOR-CRUMMEY, John; ADHIANTO, Laksono; ANDERSON, Jonathon; *et al*. HPCToolkit User's Manual Version 2024.01. [s.l.: s.n.], 2024. Disponível em: <https://hpctoolkit.org/manual/HPCToolkit-users-manual.pdf>.

ADHIANTO, Laksono ; TUTORIAL, Hpctoolkit. HPCToolkit Graphical User Interface. [s.l.: s.n.], 2021. Disponível em: <https://www.nersc.gov/assets/Uploads/HPCToolkit-GUI.pdf>. Acesso em: 4 jul. 2024.

The HPCToolkit Introduction to Rosa Course Scientific Computing Group. [s.l.: s.n., s.d.]. Disponível em: <https://www.cscs.ch/fileadmin/user_upload/contents_publications/tutorials/crayXT5introduction_course/NSS_The_HPC_Toolkit.pdf>. Acesso em: 4 jul. 2024.

SXS COLLABORATION. SpECTRE: Profiling. Spectre-code.org. Disponível em: <https://spectre-code.org/profiling.html>. Acesso em: 4 jul. 2024.
Model for Prediction Across Scales (MPAS) | Mesoscale & Microscale Meteorology. Ucar.edu. Disponível em: <https://www.mmm.ucar.edu/models/mpas>. Acesso em: 4 jul. 2024.

Model for Prediction Across Scales | National Center for Atmospheric Research. Ucar.edu. Disponível em: <https://ncar.ucar.edu/what-we-offer/models/model-prediction-across-scales-mpas>. Acesso em: 4 jul. 2024.

MPAS. Github.io. Disponível em: <https://mpas-dev.github.io/>. Acesso em: 4 jul. 2024.

CubeGUI. Helmholtz Research Software Directory, 2023. Disponível em: <https://helmholtz.software/software/cubegui>. Acesso em: 4 jul. 2024.

Cube User Guide. Fz-juelich.de. Disponível em: <https://apps.fz-juelich.de/scalasca/releases/cube/4.3/docs/manual/userguide.html>. Acesso em: 4 jul. 2024.

POP Standard Metrics for Parallel Performance Analysis | Performance Optimisation and Productivity. Pop-coe.eu. Disponível em: <https://pop-coe.eu/node/69>. Acesso em: 4 jul. 2024.

BECKER, Daniel. Scalasca. Scalasca.org. Disponível em: <https://www.scalasca.org/>. Acesso em: 4 jul. 2024.

Efficiency Metrics in a POP performance audit. [s.l.: s.n.], 2015. Disponível em: <https://pop-coe.eu/sites/default/files/pop_files/metrics.pdf>. Acesso em: 4 jul. 2024.

Learning Material / Documentation | Performance Optimisation and Productivity. Pop-coe.eu. Disponível em: <https://pop-coe.eu/further-information/learning-material>. Acesso em: 4 jul. 2024.

Scalasca: Introduction. Fz-juelich.de. Disponível em: <https://apps.fz-juelich.de/scalasca/releases/scalasca/2.5/docs/manual/intro.html>. Acesso em: 4 jul. 2024.

MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO

GOVERNO FEDERAL BRASIL UNIÃO E RECONSTRUÇÃO

LNCC Laboratório Nacional de Computação Científica

Relatório de atividades

Bolsa PIBIC/LNCC
Projeto: Simulação 3D do voo inaugural de Santos Dumont no 14 bis
Bolsista: Jonatas Halliday Sant Anna Nascimento
Orientador: Jauvane C. de Oliveira

- **Dados Gerais**

Projeto de Iniciação científica desenvolvido por Jonatas Halliday Sant Anna do Nascimento orientado por Jauvane C. de Oliveira com o título Simulação 3D do vôo inaugural de Santos Dumont no 14 bis através do PIBIC-LNCC.

- **Objetivos**

O projeto em questão tem por objetivo desenvolver uma simulação 3D do primeiro voo de Alberto Santos Dumont com o 14 bis, no campo de Bagatelle em 23 de Outubro de 1906. Tal simulação será usada posteriormente pelo Museu Casa de Santos Dumont, localizado em Petrópolis/RJ, em comemoração do sesquicentenário de Santos Dumont e do Dia do Aviador. A simulação incluirá, além do 14 bis e seu piloto, o local onde ocorreu o voo, bem como a reação da platéia lá presente.

- **Introdução**

Todo o projeto foi feito utilizando o ambiente Unity e Unity Hub para o desenvolvimento. Este ambiente é conhecido pela comunidade por ser capaz de criar simulações 2D e 3D, além de jogos dos mais variados níveis de complexidade. Neste ambiente utiliza-se a linguagem C# adaptada para unity para desenvolver os scripts utilizados no projeto. Usando um modelo do 14 bis, foi adicionado um terreno com algum grau de elevações, ou seja, não totalmente plano e vegetação característica - grama. O projeto procura ser fidedigno ao voo e portanto obedece a escala do ambiente de simulação utilizando a distância percorrida de 60 m, a uma altura de cerca de 2 m do solo. Para obter um maior grau de imersividade foi introduzida uma segunda câmera que simula o(a) observador(a) do local exato de onde Santos Dumont

esteve no 14 bis durante o voo, que durante a animação ocorrerá em forma de loop, alternando a câmera a cada reinício. A simulação obedece também os ângulos de elevação e descida do 14 bis original.

- **Métodos**

A implementação dos métodos foi feita através do modelo de classes e relacionamentos, ou seja, orientada a objetos. Foi anexado à cena do 14 bis duas câmeras existentes. Uma com a visão na parte de trás da aeronave e a segunda simulando o local do piloto, onde Santos Dumont esteve durante o voo.

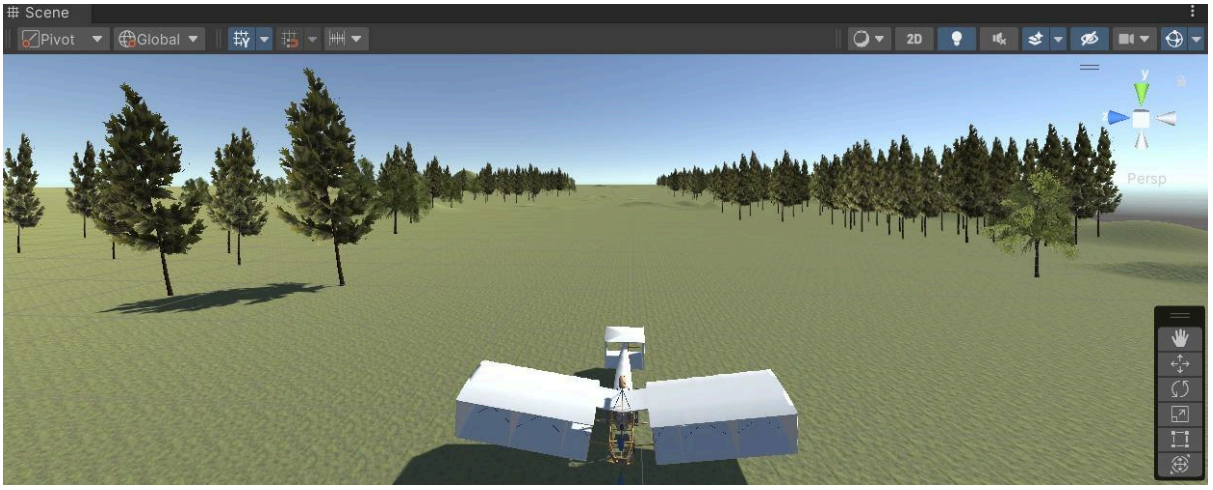


Figura 1: 14 bis antes de iniciar a animação

O terreno é composto de árvores de quatro tipos diferentes de árvores, sendo elas chamadas de: "baum hd pine fbx", "baum ld0", "baum hd med fbx" e "baum ld2". Esses modelos foram adquiridos da Unity Asset Store chamada 3D Trees. Para a movimentação vertical do 14 bis optou-se por utilizar uma funcionalidade presente do ambiente Unity chamada de NavMesh. Essa funcionalidade usa inteligência artificial para escanear o terreno e detectar outros objetos. Especificamente para o voo foi utilizado um objeto Caminho que é composto de 3 planos, totalizando os 60 m percorridos. Foi removida a textura deles (Mesh Renderer) para simular o voo e usando um outro objeto sem renderização, chamado Objetivo, que simula o ponto final, ou seja onde o 14 bis deve pousar. O modelo do 14 bis é do tipo fbx(*filmbox*) e teve todas as suas componentes separadas para ser possível ter movimentos independentes, como no caso da hélice. A lógica da rotação da hélice é simples e está anexado ao objeto da hélice no script "*Rotation.cs*": há um valor máximo de rotação possível, obtido por observação, e enquanto a animação ocorrer entre um determinado intervalo temporal, a velocidade de rotação aumenta, e após esse intervalo ela diminui até parar, já com o 14 bis parado.

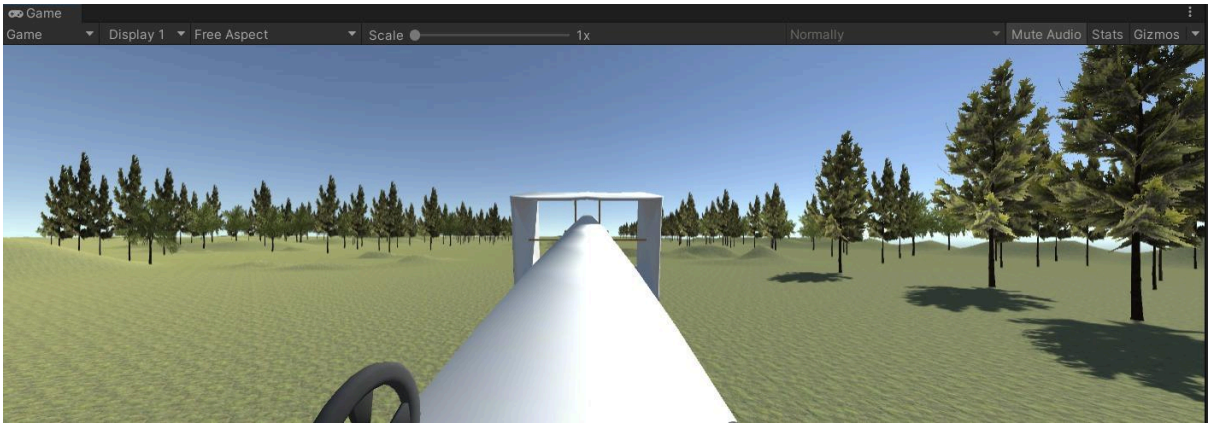


Figura 2: ângulo de visão de Santos Dumont durante o voo

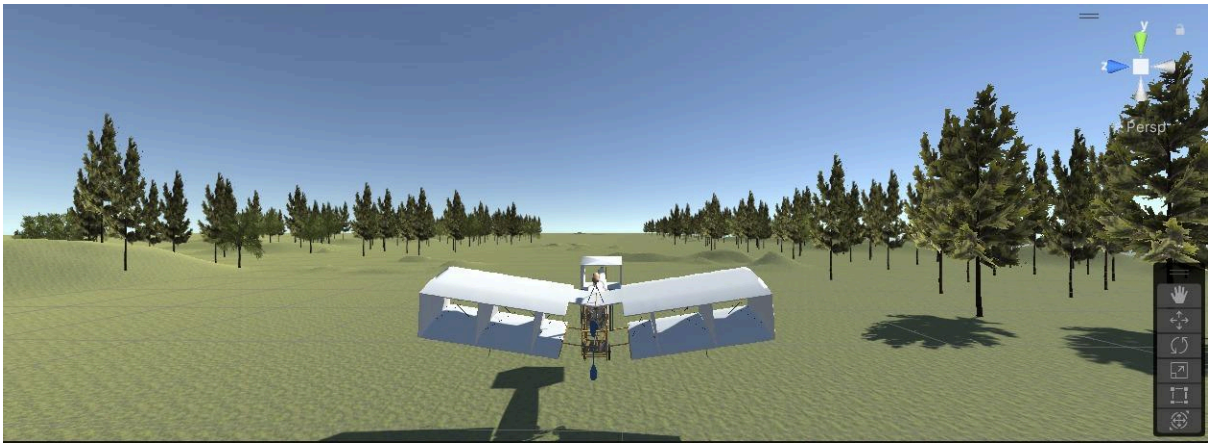


Figura 3: 14 bis com a simulação em execução

O script intitulado "*Colisao.cs*" tem por finalidade permitir que o 14 bis voe planando sobre os três planos sem a renderização de forma suave. Ele é responsável pela movimentação do 14 bis e pela mudança de ângulo do mesmo, fazendo o eixo z aumentar sua inclinação a uma taxa de 0.04 graus, e a mesma quantidade para a descida. Vale ressaltar que para simular o voo de forma fidedigna o ângulo de inclinação permanece 0.86 graus em relação ao plano de voo. O script "*Movement.cs*" é responsável pelo aumento da velocidade gradual do 14 bis- assim como ocorreu o voo original- fazendo esta aumentar durante a maior parte do percurso,mais precisamente entre os 18 e 50 s de simulação e entre 51 e 69s a velocidade do 14 bis começa gradualmente a decrescer. Concomitante ao movimento do 14 bis, ocorre também o movimento da hélice, descrito anteriormente. Para a troca entre as câmeras, um script intitulado "*ChangeCamera.cs*" que ao perceber que percorreu toda animação, simplesmente habilita a câmera que estava desabilitada e desabilita a que estava habilitada fazendo com que a cada reinício, garanta que a câmera esteja com a visão desejada.

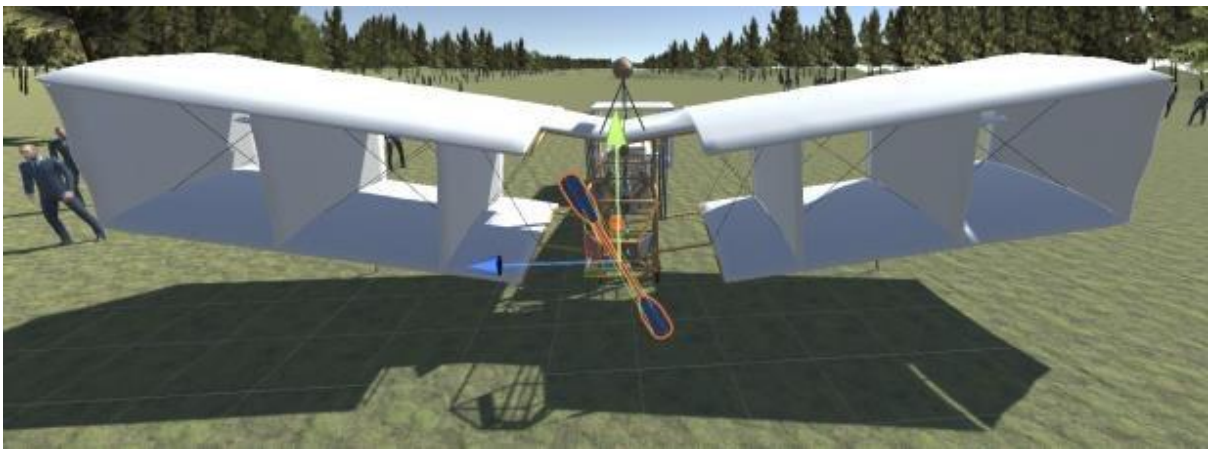


Figura 4: 14 bis em plano de voo com a hélice e turbulência implementados

Para simular a trepidação ou turbulência inerente ao movimento de voo, foi anexado ao objeto pai do 14 bis, aquele que contém todos os outros objetos que o compõem, um script intitulado "*Turbulance.cs*" que é responsável por simular a turbulência. Nele, utiliza-se o ângulo mínimo de elevação da asa do 14 bis como -**π**/ 9 e o máximo como **π**/ 9 radianos. Em seguida normaliza-se o vetor que irá variar entre esses ângulos mínimo e máximo. Isso faz com que o limite do seno seja deslocado em uma unidade para obtermos uma oscilação contínua que irá obedecer a equação: $taxaRotação = (sin(Contador(tempoDaCena) + 1)/2.$

Por fim utiliza-se a interpolação linear(*Mathf.Lerp*) entre esses dois ângulos utilizando a taxa de rotação descrita acima para então utilizar uma função *built-in* que rotaciona as asas do 14 bis ao longo da animação.

- **Resultados e Discussão**

  Uma animação contundente e realista está sendo desenvolvida e finalizada no projeto em questão, cabendo refinamentos na dinâmica do voo, na animação da plateia e também na paisagem de Paris na época, no início do século XX. Atualmente o projeto se encontra na adaptação do protótipo para ser projetado em 3D, utilizando 2 projetores, a ser utilizado no Museu Casa de Santos Dumont para divulgação ao público.

- **Conclusões**

O objetivo principal do projeto foi atingido: a animação do primeiro voo de Santos Dumont implementada. Por fim, para obtermos um modelo fidedigno e mais realista é necessário um refinamento nas animações das

pessoas presentes no campo de Bagatelle, na dinâmica do voo e na paisagem que remonte mais precisamente a Paris da época. Tal desenvolvimento trouxe o conhecimento necessário ao aluno para o desenvolvimento de outros projetos mais complexos, o que será explorado nos próximos períodos da bolsa PIBIC/LNCC.

- **Referências bibliográficas**

Unity Essentials: Unity Essentials Pathway
Representation of 3D vectors and points: Vector3 - Scripting API
Unity Essentials Working with NavMesh Agents:Working with NavMesh Agents - Unity Learn
Unity Documentation Transform RotateAround: Scripting API: Transform.Rotate
Linear Interpolation: Linear interpolation - Wikipedia
Unity Documentation Mathf.Lerp: Scripting API: Mathf.Lerp
Unity Documentation Mathf.Sin: Scripting API: Mathf.Sin
Unity Documentation Invoke: MonoBehaviour.Invoke - Scripting API
Unity Documentation GameObject SetActive: Unity - Scripting API: GameObject.SetActive

Relatório - Projeto de Iniciação Científica

Título: Seleção de Conformações Representativas da Proteína Mpro do Vírus SARS-CoV-2 Visando a Descoberta de Novos Tratamentos para a COVID-19

Bolsista: Julia Francisco dos Santos
Orientador: Laurent Emmanuel Dardenne
Coorientador: Matheus Müller Pereira da Silva
Tipo de bolsa: Iniciação científica (CNPQ)
Período: Novembro/23 a Agosto/24
Instituição: Laboratório Nacional de Computação Científica (LNCC/MCTI)

Palavras-chave: SARS-CoV-2, Mpro, Docking Molecular, Crossdocking, Reposicionamento de Fármacos
Área e Subárea: Bioinformática, Modelagem Molecular

Introdução

A pandemia de COVID-19, causada pelo vírus SARS-CoV-2, trouxe desafios significativos para a saúde global. Ao longo de mais de três anos, a pandemia deixou quase 7 milhões de óbitos e aproximadamente 770 milhões de casos confirmados no mundo inteiro (WHO, 2023).

Os vírus pertencentes à família Coronaviridae, do gênero Betacoronavírus, capazes de causar infecções em humanos, correspondem à espécie SARS-CoV, MERS-CoV e SARS-CoV-2. Evidenciando a relação taxonômica do vírus causador da COVID-19 com outros anteriormente já descobertos. Giannis et al. (2020), relata que em 2003 uma epidemia causada pelo vírus SARS-CoV-1 surgiu em Guangdong, identificado como “síndrome respiratória aguda”, se espalhando gradativamente por outros países. Assim como, em 2012, outro coronavírus foi identificado na Arábia Saudita como “síndrome respiratória do Médio Oriente”, causada pelo vírus MERS-CoV.

Sabendo que o SARS-CoV-2 apresenta relação taxonômica com o vírus causador da síndrome respiratória aguda, e com o vírus causador da síndrome respiratória do Médio Oriente, é possível definir a importância da taxonomia viral para os avanços na ciência, por permitir o fornecimento de comparação com características biológicas já descritas, a fim de identificar rapidamente um agente desconhecido e investigar um tratamento ou prevenção das patologias desenvolvidas por ele.

A princípio, é de suma importância compreendermos os mecanismos virais que levam ao ciclo de replicação do SARS-CoV-2 no organismo. A infecção se inicia a partir de uma ligação entre a glicoproteína S do vírus com uma enzima presente na célula hospedeira, denominada receptor de ECA2 (enzima conversora da angiotensina), facilitada por um receptor secundário NRP1 (neuropilina-1). Ocorrendo a endocitose a partir desse reconhecimento.

Ao entrar na célula o vírus libera seu material genético, então a replicase transcreve todo o segmento de RNA fita positiva em uma fita de RNA complementar, negativa, que serão traduzidos em componentes estruturais do vírus.

A Main Protease (M pro ou 3 CL pro) é responsável por clivar poliproteínas traduzidas a partir do segmento genômico viral, levando à liberação das proteínas virais funcionais. Ou seja, permitindo que o vírus desempenhe sua funcionalidade nas células.

Assim, influenciando no desenvolvimento da nova partícula viral, transmitindo o vírus livre e maduro por exocitose a outra célula. Garantindo a replicação.

De acordo com Guedes et al. (2020), a enzima Mpro é uma cisteína protease com um papel central na replicação viral. Ela é responsável por clivar as poliproteínas pp1a e pp1ab, ocasionando a liberação das proteínas virais NSP4 à NSP16. Se considerarmos a Mpro como um alvo molecular, sabe-se que a inibição da clivagem mediada pela Mpro impede a maturação de enzimas essenciais, como a NSP12 e a NSP13, bloqueando assim a replicação viral.

Portanto, a estrutura da cristalografia de raios X do SARS-CoV-2 Mpro mostra que esta protease pode ser encontrada nos estados monomérico e dimérico, e este último representa o estado funcional no contexto biológico.

O estudo da Mpro como alvo molecular, auxilia no desenvolvimento de novas moléculas promissoras a se tornarem fármacos, buscando aumentar a eficácia terapêutica e possibilitar formas alternativas de tratamento por menor custo.

Com a estratégia in silico se torna possível o acoplamento molecular que almeja prever o bloqueio da proteína que sintetiza o material genético ou bloqueando a proteína que promove a interação entre o vírus e a célula humana, auxiliando no desenvolvimento de fármacos.

Uma das abordagens computacionais mais utilizadas na descoberta de novos fármacos é o *docking* molecular, que tem por objetivo predizer o modo de ligação e a afinidade de uma pequena molécula em seu alvo farmacológico, geralmente uma proteína. Considerar a flexibilidade do alvo proteico é crucial em muitos casos, sendo a inclusão de múltiplas conformações representativas de um mesmo alvo uma estratégia importante para aumentar as chances de sucesso dessa abordagem.

Tendo em vista o grande conjunto de dados moleculares na bioinformática, como estruturas de proteínas e ligantes, é essencial o processo de clusterização. Esse processo permite o agrupamento de estruturas similares com base em características como conformação tridimensional, alinhamento entre sequências, interações químicas ou propriedades físico-químicas. Portanto, ao agrupar as estruturas moleculares, é possível identificar representantes de cada grupo. Focando em estruturas mais relevantes para o estudo, reduzindo a complexidade computacional e permitindo uma análise eficiente das estruturas mais promissoras.

Alguns compostos inibidores já são relatados na literatura e demonstrada sua relação com a Main Protease. Em sua maioria, relacionados à inibição da ação protease, associados ao aumento da meia-vida plasmática e tropismo pulmonar favorável, e atividade antiviral.

Objetivo geral

Selecionar, preparar e validar um conjunto de estruturas representativas da flexibilidade da proteína Mpro de SARS-CoV-2, um dos mais importantes alvos moleculares contra o vírus, tendo em vista a vasta produção científica acumulada sobre este alvo que tem culminado como resposta dos esforços globais para o tratamento da COVID-19.

Objetivos específicos

- Revisão da literatura científica atual sobre a Mpro de SARS-CoV-2.

- Obter todas as estruturas da Mpro de SARS-CoV-2 disponíveis em bancos de dados como o Protein Data Bank (PDB).
- Realizar a triagem das estruturas obtidas com base em critérios como modos de ligação, qualidade dos modelos, clusterização por similaridade e outros, visando a realização de experimentos de *docking* molecular.

Metodologia

Revisão bibliográfica

Inicialmente, realizamos uma revisão da literatura disponível sobre o tema a partir de pesquisas anteriores, para garantir maior compreensão em relação ao projeto proposto. **

Dados estruturais da Mpro

Foram obtidos os dados para o desenvolvimento do projeto, utilizando majoritariamente o Protein Data Bank (PDB), selecionando as estruturas que atendessem aos critérios solicitados na ferramenta de busca avançada. Sendo esses critérios a linhagem taxonômica (Severe acute respiratory syndrome coronavirus 2), obtida através do NCBI (National Center for Biotechnology Information), o número de classificação da enzima (3.4.22.69), obtido no PDB e os seus códigos de acesso (P0DTD1, P0DTC1), obtidos a partir do Uniprot (Universal Protein Resource). Após a análise dos resultados, foi realizado o download dos dados em CSV para manipulação em Python. Sendo esses dados, PDB IDs, métodos experimentais, resoluções e sequência proteica.

Filtragem dos dados obtidos

Com as ferramentas de programação, utilizando programas como Google Collaboratory e Visual Studio Code, selecionamos as estruturas por diversos critérios, bem como a entrada de cada método experimental, diferentes sequências proteicas, filtro para resoluções abaixo de 2.0 Å para difração de raio-X e abaixo de 4.0 Å para microscopia eletrônica. Em relação às estruturas obtidas por RMN (Ressonância Magnética Nuclear), esta foi selecionada manualmente pois havia apenas uma que atendesse aos filtros anteriormente aplicados. Para lidar com a construção de dataframes, foi utilizada a biblioteca Pandas na linguagem de programação Python.

Dados dos ligantes

Utilizando o mesmo banco de dados PDB, selecionamos os ligantes que atendessem aos critérios solicitados na ferramenta de busca avançada, os mesmos que os utilizados para buscar dados referentes a Mpro. Após a análise dos resultados obtidos na busca avançada, foi realizado download dos dados em JSON para manipulação em Python. Sendo os dados referentes aos ligantes: IDs, fórmula estrutural e ligantes de interesse.

Ao obtermos o arquivo JSON com informações acerca dos ligantes, ainda que pequenas moléculas também estivessem listadas em relação a estrutura cristalográfica, o ligante de interesse era denominado no arquivo como “Subject of investigation”.

Utilizando a biblioteca RDKit, a partir da fórmula estrutural dos ligantes, foi calculado o peso molecular de cada um deles. Por fim, foram adicionados ao dataframe final apenas as estruturas de interesse para o projeto, com peso molecular abaixo de 1500 Da.

Clusterização

Após filtrar os dados, foi realizado o download de todas as estruturas contidas no dataframe final, a partir do site do RCSB PDB, no formato PDBx/mmCIF. Após a coleta das estruturas, removemos os ligantes de todas os complexos e selecionamos os resíduos de aminoácidos de referência presentes no sítio ativo da Mpro, possibilitando o alinhamento de todas as estruturas entre si. Esses resíduos de aminoácidos foram selecionados a partir de informações sobre o sítio ativo da Mpro contidas na literatura científica.

Segundo Zhenming et al. (2020), ao detalhar as interações específicas do inibidor proposto, consegue-se selecionar os resíduos de aminoácidos importantes para definir o sítio ativo da Mpro. Sendo eles: T24-T25, H41-Y54, F140-G143, H163-H172, F185-Q192.

De acordo com Hu et al. (2022), a Mpro do SARS-CoV-2 apresenta 12 resíduos de cisteína, sendo que a C145 é a cisteína mais importante localizada no sítio catalítico, pois faz parte da díade catalítica Cis145-His41, de extrema importância funcional. É importante levarmos em consideração o sítio ativo da enzima, capaz de se ligar e desempenhar suas funções acerca do metabolismo humano

Para o carregamento das estruturas no Pymol, utilizamos scripts Python para carregá-las. Com os scripts, buscamos alinhar todas as estruturas a partir do resíduo C145, pertencente a díade catalítica.

Após o alinhamento, o próprio Pymol a partir dos comandos utilizados é capaz de calcular o RMSD de cada par de estruturas alinhadas. Quando obtêm-se a matriz de distância de todas as estruturas, utilizamos algoritmos para realizar a clusterização, alinhando estruturas possivelmente semelhantes em grupos, denominados “clusters”. Alguns desses algoritmos podem ser utilizados em Python, implementados no pacote chamado Scikit Learn, como Agglomerative Clustering, DBSCAN e Spectral Clustering.

Resultados e discussão

A partir dos dados estruturais da Mpro, obtidos do PDB, foi construído um dataframe inicial com 1354 estruturas. Separando cada uma dessas estruturas por método experimental, tivemos como resultado 1329 estruturas por difração de raios X; 4 estruturas por difração de nêutrons; e apenas 1 por solução RMN (Ressonância Magnética Nuclear), geralmente usadas para estudar as estruturas tridimensionais de proteínas, ácidos nucléicos e outras biomoléculas em solução.

Ao filtrarmos as sequências, foram obtidas 134 sequências diferentes entre si. Essas diferenças podem ser explicadas por variações genéticas e mutações naturais, pois, levando em consideração a morfologia estrutural do SARS-CoV-2, este é um vírus envelopado que possui RNA de fita simples, de polaridade positiva, apresentando maior tendência a desenvolver mutações.

Os dados sobre os ligantes relacionados a Mpro foram obtidos através do PDB, adicionando as informações necessárias ao dataframe anterior. Com isso, selecionamos apenas os ligantes de interesse, denominados no PDB como “Subject of Investigation”. Em uma estrutura PDB é comum encontrar pequenas moléculas associadas à proteína. O ligante de interesse, é analisado como um inibidor, substrato ou outra molécula que interaja no sítio ativo. O DMSO (Dimetil sulfóxido), outra molécula comum em estruturas PDB, é utilizado como solvente orgânico em experimentos bioquímicos, aparecendo na estrutura cristalizada se estiver presente durante a cristalização. Também podem ser encontrados

íons, como por exemplo o íon cloreto, presentes na estrutura devido ao ambiente cristalino ou às condições de purificação e cristalização.

Além dessa informação descrita acima sobre os ligantes, também consideramos o ID de cada um deles e separamos as fórmulas estruturais, com representação por SMILES. Essa etapa é importante para calcular o peso molecular, utilizando a biblioteca RDKit em Python.

Em seguida, removemos as estruturas obtidas por difração de raio X com resolução acima de 2.0Â, e as estruturas obtidas por microscopia eletrônica, com resolução acima de 4.0Â.

Para a construção do dataframe, filtramos as estruturas de interesse para a pesquisa, retirando as que não possuíam ligantes e as que não apresentavam sequência Wild Type (WT).

Após aplicarmos todos os filtros citados, obtivemos 769 estruturas.

Em seguida, foi realizada a técnica de clusterização utilizando o programa Pymol. Selecionamos os resíduos de aminoácidos presentes no sítio ativo da cadeia A. Sendo eles: T24-T25, H41-Y54, F140-G143, H163-H172, F185-Q192, demonstrados por Zhenming et al. (2020). Levando em consideração a díade catalítica da Mpro Cis-His, a qual apresenta os resíduos H41 e C145, selecionamos o resíduo C145 preferencialmente para alinhar todas as estruturas contra todas.

Para realizar o alinhamento entre as estruturas a partir do resíduo C145, utilizamos o Pymol em Python.

Conclusão

A partir do reconhecimento estrutural da Main protease do SARS-CoV-2 e sua relação com estruturas e atividades já conhecidas, busca-se reconhecer possíveis inibidores desse alvo molecular, que possam atuar em novos fármacos, a fim de estabelecer um tratamento adequado para a doença causada pelo SARS-CoV-2. Inibindo a atividade da protease principal. Assim, impedindo o desenvolvimento da nova partícula viral a barrando a transmissão de um possível vírion por exocitose.

Perspectivas

- Organizar as estruturas em clusters utilizando o programa Pymol.
- Calcular o RMSD apenas dos resíduos de aminoácidos do sítio ativo.
- Realizar experimentos de docking e crossdocking.
- Realizar experimentos de docking e reposicionamento de fármacos utilizando o programa DockThor, de forma que as estruturas resultantes deste trabalho serão integradas ao portal DockThor-VS, o qual já conta atualmente com conjuntos específicos para o SARS-CoV-2. O portal DockThor-VS está acoplado ao ambiente de supercomputação Santos Dumont e é acessível gratuitamente para toda comunidade científica.

Referências

1. World Health Organization (WHO). (2023). Retrieved from https://covid19.who.int/ at 13/10/2023.
2. GIANNIS, D.; ZIOGAS, I. A.; GIANNI, P. Coagulation disorders in coronavirus infected patients: COVID-19, SARS-CoV-1, MERS-CoV and lessons from the past. Journal of Clinical Virology, v. 127, p. 104362, jun. 2020.
3. Guedes, I. A., Costa, L. S., Dos Santos, K. B., Karl, A. L., Rocha, G. K., Teixeira, I. M., ... & Dardenne, L. E. (2021). Drug design and repurposing with DockThor-VS web server focusing on SARS-CoV-2 therapeutic targets and their non-synonym variants. Scientific Reports, 11(1), 5543.
4. Jin, Z., Du, X., Xu, Y., Deng, Y., Liu, M., Zhao Y., Zhang B., ... & (2020). Structure of Mpro from SARS-CoV-2 and discovery of its inhibitors. Nature 582, 289–293 (2020).
5. Hu, Q., Xiong, Y., Zhu, G. H., Zhang, Y. N., Zhang, Y. W., Huang, P., & Ge, G. B. (2022). The SARS‐CoV‐2 main protease (Mpro): structure, function, and emerging therapies for COVID‐19. MedComm, 3(3), e151

Relatório de Atividades

Título: Desenvolvimento de um conjunto de teste para complexos de inclusão de ciclodextrinas. Otimização e validação por meio do DockThor.

Bolsista: Juliana do Nascimento Amaral.
Orientador: Laurent Emmanuel Dardenne.
Coorientadores: Isabella Alvim Guedes e Letícia Cristina de Assis.
Instituição: Laboratório Nacional de Computação Científica (LNCC/MCTI).

Tipo de bolsa: PIBIC/CNPq.
Período da bolsa: Novembro de 2023 a agosto de 2024.

Introdução

As ciclodextrinas (CDs) são oligossacarídeos cíclicos compostos por unidades de glicose, ligadas por ligações α-1,4, formadas pela degradação enzimática do amido (Crini et al., 2018). Existem três tipos principais de CDs: α-CD, β-CD e γ-CD, com seis, sete e oito subunidades de D-glicose, respectivamente. A estrutura das CDs é um cone truncado, com uma cavidade interna hidrofóbica e uma superfície externa hidrofílica (Sandilya et al., 2020). Essas cavidades permitem a captura seletiva de moléculas hóspedes (ligantes), formando complexos de inclusão. Devido a essa capacidade, as CDs são aplicadas em diversas áreas, especialmente para melhorar a solubilidade e a biodisponibilidade de medicamentos. A química das CDs é de grande interesse científico, mas até agora não há ferramentas de ancoragem específicas para complexos de ciclodextrina-ligante (CD-ligante). O programa DockThor (disponível gratuitamente no portal web www.dockthor.lncc.br), foi desenvolvido pelo Grupo de Modelagem Molecular em Sistemas Biológicos (GMMSB) para complexos proteicos (de Magalhães et al., 2014; Santos et al., 2020; Guedes et al., 2021a, 2024). O Dockthor é baseado em grade projetado para o ancoramento de ligantes altamente flexíveis e receptores rígidos. Ele emprega um algoritmo genético de solução múltipla e a função de pontuação do campo de força molecular *MMFF94S* para a previsão de pose. Além disso, a predição de afinidade e classificação dos complexos de proteína-ligante são realizados com a função de pontuação empírica linear *DockTScore* (Guedes et al., 2024). Enquanto que o portal DockThor-VS têm mostrado excelentes resultados em estudos comparativos com programas de ancoramento molecular convencionais, alcançando uma taxa de sucesso de 80% na previsão do modo de ligação e uma acurácia na predição de afinidade comparável aos melhores modelos da literatura (R ~ 0.7) (Santos et al., 2020; Guedes et al., 2021b).

Diante disso, este projeto visa testar complexos de CD-ligante no programa DockThor para avaliar sua eficácia na predição dos modos de ligação.

Objetivos

Desenvolver conjuntos de teste contendo complexos de ciclodextrina-ligante (CD-ligante) e testá-lo por meio de ancoramento molecular, usando o programa DockThor.

Objetivos específicos:

- Realizou-se uma pesquisa na literatura sobre as estratégias de ancoramento molecular visando ciclodextrinas como estrutura receptora;
- Selecionamos e preparamos, através do programa Maestro, dos conjuntos de teste específicos para CD-ligante;
- Serão realizados experimentos de reancoramento utilizando o programa DockThor para avaliar a capacidade preditiva do programa.

Metodologia

Neste primeiro período de bolsa vinculada ao projeto, foram realizadas as atividades descritas a seguir, sendo que algumas atividades ainda estão em desenvolvimento.

Revisão bibliográfica

Com base nas revisões da literatura, sobre as estratégias de ancoramento molecular aplicadas aos complexos CD-ligante, estão sendo obtidas informações importantes para a execução desse trabalho, tais como: (a) conceito de ciclodextrinas, os tipos existentes, a natureza dos complexos de inclusão e suas aplicações; (b) principais fontes para obtenção e seleção das estruturas tridimensionais dos complexos CD-ligante; (c) abordagens sobre a técnica de ancoramento molecular (como, reancoramento e rmsd); (d) como preparar conjuntos de testes do complexo CD-ligante, como entrada para o reancoramento molecular.

Seleção de conjuntos de teste para os complexos CD-ligante:

Selecionamos duas fontes de dados de conjuntos de testes diferentes para validar as metodologias de ancoramento receptor-ligante do programa DockThor, considerando ciclodextrinas como estruturas receptoras. O primeiro conjunto de dados, nomeado como "*cdCSD2023",* foi obtido no *Cambridge Structural Database* (CSD), este banco contém

estruturas tridimensionais de complexos de CDs com diversas moléculas hóspedes. O CSD é uma fonte altamente confiável, incluindo detalhes químicos, cristalográficos e experimentais dos complexos. É acessível no portal https://www.ccdc.cam.ac.uk/ e requer uma assinatura (Moghadam et al., 2017). O segundo conjunto de dados, chamado *"Amaral2023"* é composto por dois subconjuntos, nomeado como "*GH2023"* e *"Sampl2023",* retirados da literatura e estão em fase de construção (Henriksen and Gilson, 2017a; Amezcua et al., 2021).

Preparação de conjuntos de teste para os complexos CD-ligante:

Na preparação do conjunto "*cdCSD2023"* está sendo adotado o seguinte protocolo:

1. Íons e cofatores metálicos que apresentam incompatibilidade ao DockThor, solventes que não participaram diretamente da interação CDs-ligante foram removidos, usando o programa *PyMOL* versão 2.5.0 (Schrodinger and DeLano, 2020).

2. Os ligantes e receptores mantiveram seus estados de protonação preditos de acordo com o pH experimental utilizado para determinar a estrutura 3D. Nos casos em que não houve registro desses estados, adotamos as condições fisiológicas (pH = 7), utilizando a ferramenta *PrepWizard* do Maestro com os preditores de *pKa Epik* e *Propka* (Maestro, Schrödinger,LLC,New York, NY, 2024).

3. Todas as estruturas de ligantes e receptores processadas foram salvas como arquivos .mol2 e .pdb, respectivamente para os cálculos de ancoramento molecular subsequentes.

Resultados e Discussão

Os conjuntos de teste contendo complexos de CD-ligante foram desenvolvidos e serão avaliados por meio de reancoramento molecular utilizando o programa DockThor. O conjunto "*cdCSD2023"* inclui 135 complexos de β-CD-ligante, reduzidos a partir de 148, que foram obtidos do *CSD* (Groom et al., 2016).

O conjunto *"Amaral2023"* é composto por "*GH2023" e "Sampl2023".* O "*GH2023" (*https://github.com/mobleylab/benchmarksets) foi coletado do *github,* sendo formado por dois subconjuntos: *cd-set1*, com 22 α-CD-ligante (Rekharsky et al. 1997; Henriksen e Gilson, 2017), e *cd-set2,* com 21 complexos de β-CD-ligante (Wickstrom et al., 2013; Zhang et al., 2016; Henriksen and Gilson, 2017b). O *"Sampl2023"* que possui complexos de CD-ligante selecionados no desafio cego de Avaliação Estatística de Modelagem de Proteínas e

RELATÓRIO PARCIAL DE ATIVIDADES - PROJETO DE INICIAÇÃO CIENTÍFICA
BIOINFORMÁTICA, BANCO DE DADOS E ENGENHARIA DE COMPUTAÇÃO

Título do Projeto:
Avaliação de métodos de aprendizado de máquina em gateways científicos de bioinformática

Instituição:
Laboratório Nacional de Computação Científica

Nome do Aluno:
Kelen Brito Souza

Nome do orientador e coorientador:
D.Sc. Kary Ann del Carmen Ocaña Gautherot (Tecnologista Pleno – LABINFO/LNCC, Orientador)
D.Sc. Carla Osthoff Ferreira de Barros (Tecnologista Sênior – SEPAD/COTIC//LNCC, Coorientador)
M.Sc. Micaella Coelho Valente de Paula (Analista de Sistemas – Petrobras, SEPAD/COTIC//LNCC, Coorientador)

Tipo de bolsa: Iniciação Científica - PIBIC

Período do relatório: 10 de Janeiro a 11 Julho de 2024

1. Objetivo

O objetivo desta pesquisa é investigar o desempenho do programa de bioinformática PA-Star em variadas arquiteturas e configurações no supercomputador Santos Dumont (SDumont). PA-Star foi desenvolvida pela equipe colaboradora da UnB para o alinhamento múltiplo de sequências (MSA - *Multiple Sequence Alignment* em inglês) utilizando CPU. As análises até agora se concentraram em comparar duas estruturas do SDumont: MESCA2 e Nvidia (nvidia_long). Planeja-se, para etapas futuras, desenvolver um workflow que permita examinar o impacto das arquiteturas de memória compartilhada e distribuída em processos de alinhamento múltiplo de sequências (MSA).

2. Introdução

O alinhamento múltiplo de sequências (MSA) é uma técnica essencial na bioinformática, que consiste em alinhar três ou mais sequências biológicas — como DNA, RNA ou proteínas — para descobrir áreas de semelhança que podem indicar relações funcionais, estruturais ou evolutivas. Executar MSA em CPUs apresenta desafios computacionais, particularmente com sequências extensas e complexas ou grandes volumes de dados. Embora as CPUs sejam versáteis, elas podem ser mais lentas que as GPUs para cálculos paralelos. Contudo, vários algoritmos de MSA são adaptados às CPUs, tirando proveito de sua capacidade de processamento paralelo e memória.

Neste estudo, utilizamos o PA-Star, um programa baseado no algoritmo de busca A-star (A*), que é eficiente na identificação do caminho de custo mínimo entre dois pontos em um grafo direcionado e ponderado. O A* é uma adaptação do algoritmo de Dijkstra, diferenciando-se pela função de avaliação que decide o próximo nó a ser explorado.

O algoritmo A* é aplicável a qualquer problema de busca de caminho. No entanto, nosso foco é o MSA, implementado pelo programa PA-Star, que executa uma busca paralela A-Star para solucionar o MSA. Existe também uma versão serial do programa, msa_astar, mas apenas a versão paralela é usada nesta pesquisa. O PA-Star, projetado para Linux, é estável, mas enfrenta desafios de memória. Inicialmente, utilizamos a arquitetura Mesca2 do SDumont, que possui a maior memória RAM, e depois comparamos com a arquitetura Nvidia (nvidia_long), que tem configuração semelhante.

3. Metodologia

3.1. Dados de Entrada

Os dados foram extraídos do repositório astar-msa, de dados previamente testados pelo criador do programa PA-Star Daniel Sundfeld. Dentre as sequências testadas estão as que são encontradas na pasta Benchmark disponibilizadas pelo criador no repositório chamado “seqs”. Foram testadas em ordem sugerida pelo próprio Daniel Sundfeld, consistindo em sequências iniciais de tipo .fasta: glg, 1sbp, 1aboA e 1ac5. E posteriormente, com base na escalabilidade das sequências iniciais foram testadas as nove sequências de tipo .fasta, que seriam mais complexas e precisariam de mais memória, que são: gal4, 1gpb, arp, 1sesA, 2myr, 2cba, 1hvA, 2ack e actin.

3.2. O Ambiente Computacional Santos Dumont

O supercomputador Santos Dumont (SDumont) tem uma capacidade de processamento de 5,1 Petaflop/s e apresenta uma configuração híbrida com 1.134 nós computacionais. Esses nós incluem uma maioria de CPUs multicore e uma quantidade significativa de dispositivos many-core como GPUs e MICs. O nó MESCA2 se destaca com 240 núcleos e 6 Tb de memória RAM, e há um nó especializado para Inteligência Artificial com 8 GPUs NVIDIA Tesla V100-16Gb. Os nós são interconectados por uma rede Infiniband de alta performance e baixa latência, e o SDumont possui um sistema de arquivos paralelo Lustre com 1,7 PBytes de armazenamento, além de um sistema secundário com 640 TBytes. Todos os programas, algoritmos e ferramentas necessárias para bioinformática foram configurados e instalados no ambiente do supercomputador Santos Dumont.

3.3. Arquitetura de memória compartilhada e distribuída no SDumont

Até então foram analisadas duas arquiteturas de memória distintas no SDumont para verificar qual apresenta melhor eficácia e performance durante o alinhamento múltiplo de sequências usando o PA-Star. O sistema com maior foco nessa pesquisa, foi o Mesca2, visto que o maior desafio do PA-Star é memória suficiente.

Então foi usado o Mesca2, que é uma arquitetura de memória compartilhada, com seus 240 núcleos e 6 TB de RAM, possibilita um acesso direto e veloz aos dados por parte dos processadores, sendo ideal para operações que necessitam de uma comunicação e sincronização ágeis, sem a complexidade de sistemas externos. E como uma comparação foi selecionada a arquitetura Nvidia Ivy Bridge, com 64 GB de RAM por nó, adequada para atividades que são divididas e não requerem um amplo compartilhamento de dados entre os nós. Neste caso, cada nó opera com sua memória local e a comunicação se dá através da rede, o que pode acarretar em latências superiores se comparadas ao acesso imediato proporcionado pela memória compartilhada.

3.4. Desempenho computacional do PA-Star em 1 nó no Mesca2 e no Nvidia (nvidia_long)

Primeiramente, foi executado um script de submissão no Mesca2, para definir a escalabilidade das sequências iniciais de tipo .fasta: glg, 1sbp, 1aboA e 1ac5. Foram testadas as configurações de 1, 12, 24, 48, 60, 120 e 240 threads para avaliar o desempenho do Mesca2.

Após ser revelado que a melhor configuração nos testes iniciais foi de 24 threads, e levando em consideração que todas as sequências testadas são similares em tamanho, foi executado um script de submissão no Mesca2 com 24 threads, para avaliar o desempenho no alinhamento das sequências que seriam mais complexas: gal4, 1gpb, arp, 1sesA, 2myr, 2cba, 1hvA, 2ack e actin.

Depois disso foi feita uma comparação entre as arquiteturas Mesca2 e Nvidia, e como o PA-Star só roda em apenas um nó foi comparado usando as configurações de 1, 12 e 24 threads, a fim de verificar o melhor desempenho na análise das sequências iniciais. Nas nove sequências mais complexas foram testados em 24 threads apenas.

4. Resultados e Discussões

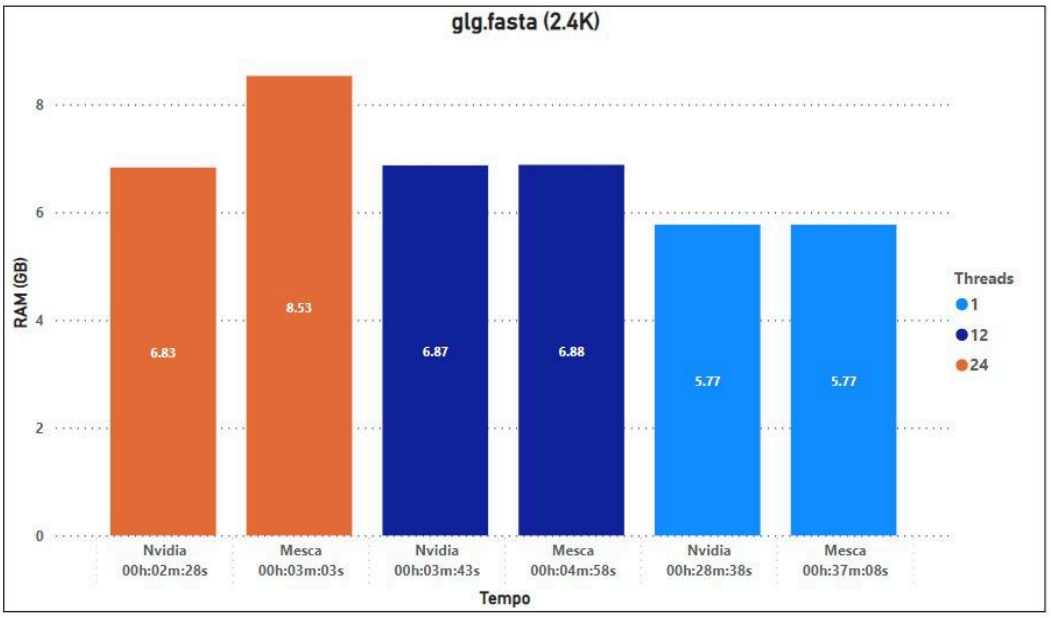


Figura 1: glg.fasta testado no Mesca e Nvidia

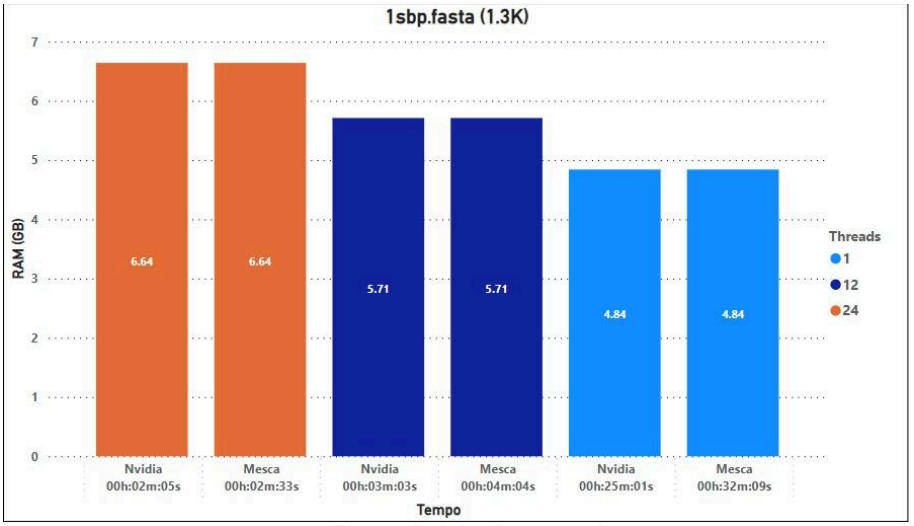


Figura 2: 1sbp.fasta testado no Mesca e Nvidia

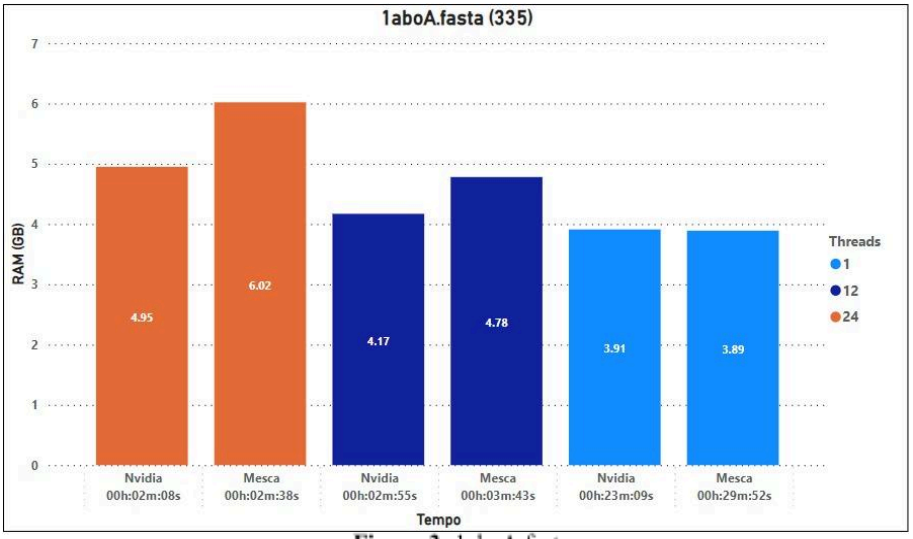


Figura 3: 1aboA.fasta testado no Mesca e Nvidia

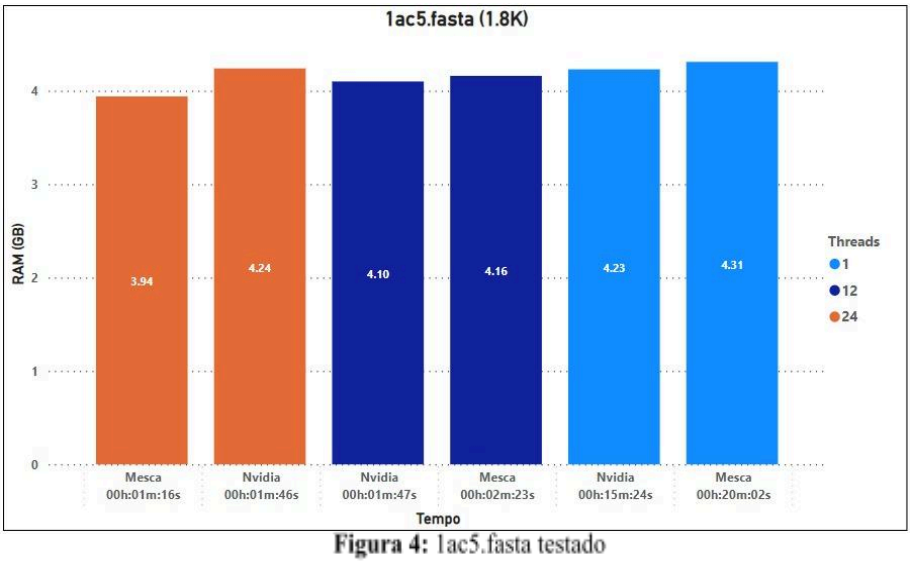


Figura 4: 1ac5.fasta testado no Mesca e Nvidia

As figuras de 1 a 4 apresentam dados obtidos com as sequências iniciais em formato .fasta: glg, 1sbp, 1aboA e 1ac5, ao executar um script de submissão no Mesca2 e no Nvidia. Para avaliar a escalabilidade desses primeiros arquivos, foram testadas configurações com 1, 12, 24, 48, 60, 120 e 240 threads. Primeiramente, o desempenho foi analisado no Mesca2, e posteriormente, essas configurações foram utilizadas no Nvidia, considerando que o programa só roda em 1 nó, limitando o teste a no máximo 24 threads, que é o máximo por nó no Nvidia. Observou-se que com poucas threads (como 1 e 12 threads), o tempo de execução é muito longo. Com 24 threads, o desempenho foi o melhor, apresentando o tempo de execução mais rápido. A quantidade de RAM necessária aumentou quase proporcionalmente ao número de threads.

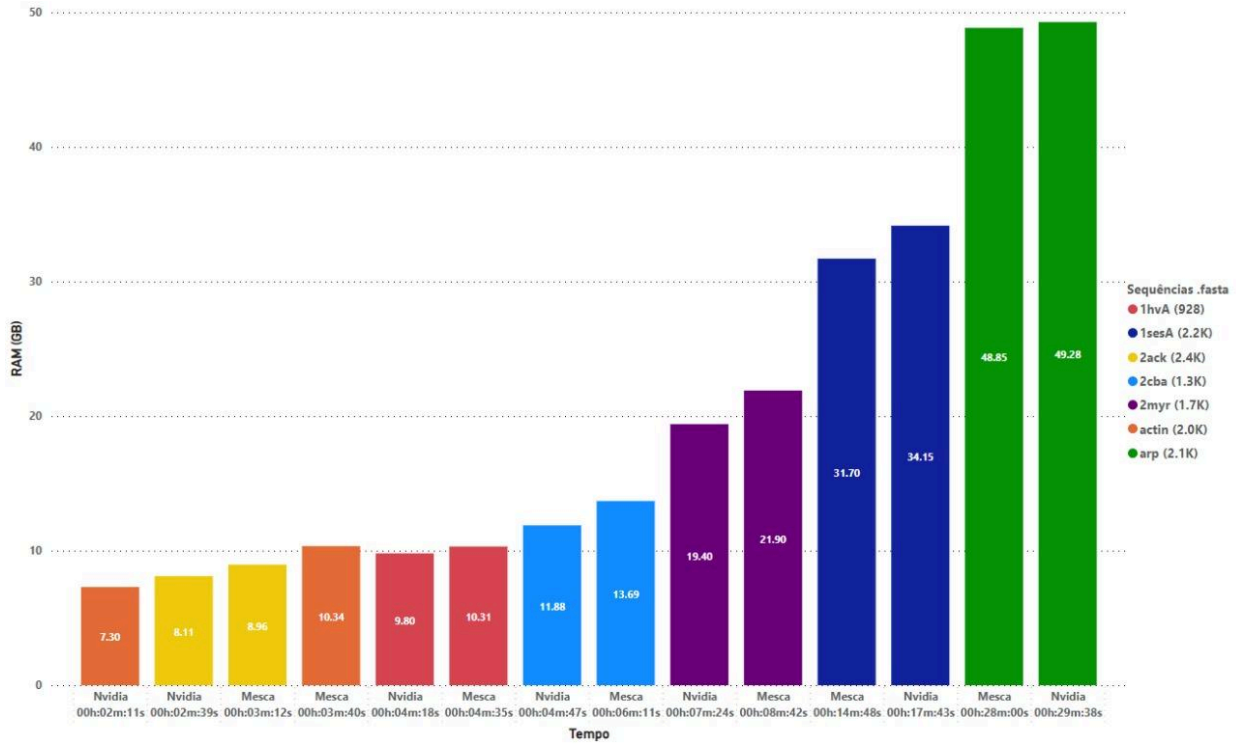


Figura 5: sequências mais complexas no Mesca e Nvidia

A Figura 5 apresenta os resultados da segunda parte da pesquisa, que consistiu em testar sequências mais complexas no script de submissão com 24 threads, configuração que se revelou a melhor nos testes iniciais. Todas as sequências testadas são próximas em tamanho. As sequências mais complexas - gal4, 1gpb, arp, 1sesA, 2myr, 2cba, 1hvA, 2ack e actin - foram testadas no Mesca2 e no Nvidia. Os resultados mostraram que, quanto maior o tamanho e a complexidade da sequência, mais tempo é necessário para rodar o script. As sequências gal4 e 1gpb não estão apresentadas no gráfico visto que não conseguiram rodar no Nvidia nas condições estabelecidas, portanto não foi possível comparar as arquiteturas. O Mesca2 demonstrou ser suficiente para rodar todas as sequências nas condições apresentadas.

5. Conclusões

Diante da pesquisa e testes feitos do PA-Star no alinhamento múltiplo de sequências (MSA) no SDumont, foi possível analisar que mesmo com tamanhos parecidos as sequências apresentam singularidades e complexidades diferentes. Em todo caso, foi visto que o Mesca2 consegue ter um desempenho muito bom no processamento dessas sequências. Não havendo problemas de falha, ou *timeout*, sempre conseguindo processar e entregar resultados.

Porém quando falamos na comparação do Mesca2 e Nvidia nas condições apresentadas neste trabalho, percebe-se uma facilidade maior do Mesca2 no processamento de sequências mais complexas e que demandam mais memória. Como no caso das sequências gal4, 1gpb, arp e 1sesA, somente o Mesca2 apresenta um processamento rápido, visto que o Nvidia nem conseguiu processar essas sequências nas mesmas configurações. Já nos outros casos de sequências mais simples e que exigem menos do SDumont é notável a vantagem do NVidia.

Isto tudo se deve ao fato de que o Mesca2 acaba sendo uma máquina extremamente potente e que sobra desempenho em tarefas simples, deixando assim threads ociosas. Porém quando falamos de sequências mais complexas e maiores (que provavelmente é o futuro dessa pesquisa) ele revela sua superioridade e agilidade no alinhamento de tais sequências.

Os próximos passos dessa pesquisa serão de grande importância para se aprofundar no uso Mesca2 nas próximas sequências, obtendo assim uma melhor compreensão do funcionamento do PA-Star no Santos Dumont, através de um workflow visando analisar as arquiteturas estudadas nos processos de alinhamento múltiplo de sequências (MSA).

6. Referências Bibliográficas

[1] SUNDFELD, Daniel; RAZZOLINI, Caina; TEODORO, George; BOUKERCHE, Azzedine; MELO, Alba Cristina Magalhães Alves de. PA-Star: A disk-assisted parallel A-Star strategy with locality-sensitive hash for multiple sequence alignment. Journal of Parallel and Distributed Computing, v. 112, p. 154-165, 2018.

[2] LIMA, Daniel Sundfeld. Alinhamento primário e secundário de sequências biológicas em arquiteturas de alto desempenho. 2017. xx, 167 f., il. Tese (Doutorado em Informática) — Universidade de Brasília, Brasília, 2017.

[3] SDUMONT - LNCC. Manual. Disponível em: <https://sdumont.lncc.br/support_manual.php?pg=support>. Acesso em: 10 jul. 2024.

[4] CODE ACADEMY. Code Documentation. Disponível em: <https://www.codecademy.com/resources/docs/ai/search-algorithms/a-star-search>. Acesso em: 10 jul. 2024.

[5] ZAFALON, Geraldo Francisco Donegá. Aplicação de estratégias híbridas em algoritmos de alinhamento múltiplo de sequências para ambientes de computação paralela e distribuída. 2014. Tese (Doutorado em Sistemas Digitais) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2014. doi:10.11606/T.3.2014.tde-28082015-120515. Acesso em: 10 jul. 2024.

[6] UNICAMP. Instituto de Computação. Disponível em: <https://www.ic.unicamp.br/~zanoni/teaching/mo640/2021-2s/aulas/03-MSA.pdf>. Acesso em: 10 jul. 2024.

PROGRAMA DE BOLSAS PCI/LNCC

Relatório Final de Atividades

TÍTULO DO PROJETO: Pesquisa em GPU: Multiplicação de matrizes densas.

Bolsista: Luiza Augusto Tavares

Orientador: Carla Osthoff Ferreira de Barros e Micaella Coelho

Nível de bolsa: PIBIC

Período de bolsa: Outubro de 2023 a setembro de 2024

1. Objetivo

Este projeto tinha como objetivo realizar uma análise comparativa do desempenho entre duas GPUs, observando o comportamento durante a execução de um código otimizado pela NVIDIA em cada uma delas. As duas arquiteturas analisadas foram a GeForce GTX 1660, pertencente à máquina local do CENAPAD, e a NVIDIA V100, do SDumont.

2. Introdução

Este relatório apresenta a pesquisa desenvolvida entre o período de outubro a julho de 2024, que teve como objetivo explorar a utilização da biblioteca cuBLAS, uma implementação do BLAS (Basic Linear Algebra Subprograms) sobre a plataforma de execução CUDA™ da NVIDIA. A biblioteca cuBLAS permite o acesso aos recursos computacionais da Unidade de Processamento.

No decorrer deste projeto, foi utilizada a função cublasGemmEx, que calcula o produto de duas matrizes A e B, multiplicando o resultado por um escalar alpha e somando-o com o produto de uma matriz C por um escalar beta. O objetivo principal foi executar o código em duas arquiteturas distintas e gerar uma curva de desempenho comparativa entre duas GPUs.

As duas arquiteturas analisadas foram a GeForce GTX 1660 e a NVIDIA V100. A GeForce GTX 1660, lançada em 2019, combina eficiência energética e desempenho sólido para jogos e criação de conteúdo. Baseada na arquitetura Turing, possui 1408 núcleos CUDA e 6 GB de memória. Ideal para jogos em Full HD, destaca-se pela boa relação custo-benefício. Ela é uma escolha popular para usuários que buscam desempenho competente sem gastar muito.

Por outro lado, a NVIDIA V100, lançada em 2017, é uma GPU de alto desempenho voltada para data centers e inteligência artificial, baseada na arquitetura Volta. Com 5120 núcleos CUDA, 640 núcleos Tensor e 16/32 GB de memória HBM2, oferece desempenho excepcional para aprendizado profundo e computação de alto desempenho. Suportando tecnologias como NVLink, ela melhora a comunicação entre GPUs. A V100 é ideal para treinamento de IA, simulações científicas e análises de dados, sendo uma escolha premium para tarefas que exigem poder de processamento massivo.

Os resultados obtidos nesta pesquisa destacam o desempenho do código em duas arquiteturas distintas, possibilitando a análise e a compreensão das capacidades das arquiteturas de GPUs para executar operações complexas de álgebra linear. Essa análise auxilia identificar as vantagens e limitações de cada sistema ao lidar com cálculos intensivos.

3. Materiais e métodos

O ambiente utilizado para os experimentos incluiu a GPU da máquina local da sala do CENAPAD, equipada com uma GeForce GTX 1660 baseada na arquitetura Turing, com 6 GB de memória, e o supercomputador SDumont, que conta com uma NVIDIA V100 da arquitetura Tesla com 32 GB de memória.

Utilizamos um gerador de matrizes capaz de fornecer duas matrizes (A e B) conforme as necessidades de cada projeto, podendo ser matrizes densas, esparsas ou vetores. O código empregado para a realização da multiplicação das matrizes foi otimizado pela NVIDIA, utilizando a biblioteca cuBLAS e a função cublasGemmEx, que calcula o produto de duas matrizes A e B.

Para a execução do código, foram utilizados dois arquivos de entrada, consistindo em duas matrizes densas (A e B) com as mesmas dimensões. As dimensões das matrizes utilizadas nos experimentos foram 128, 256, 512, 1024, 2048, 4096 e 8192.

4. Resultados e discussões

Os resultados das execuções das matrizes em cada GPU foram coletados e organizados em uma planilha de dados. Com base nesses dados (Tempo de transferência, Kernel e transferência + Kernel), foram gerados gráficos individuais para cada tempo registrado.

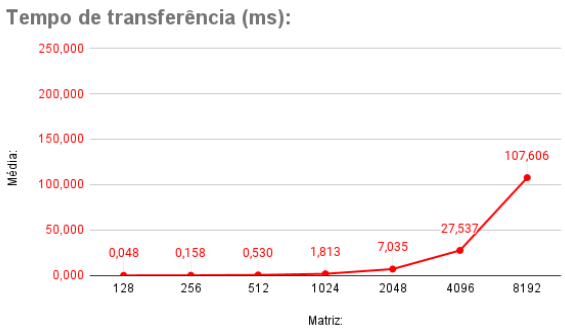


Figura 1: Tempo de trans. na máquina do CENAPAD.

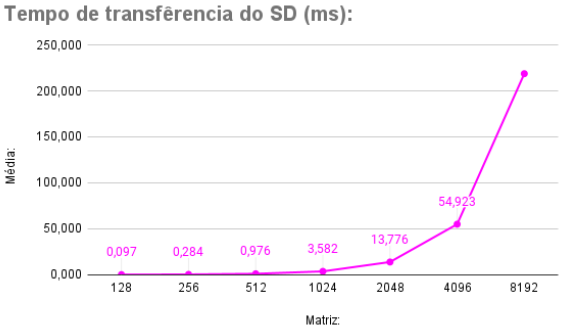
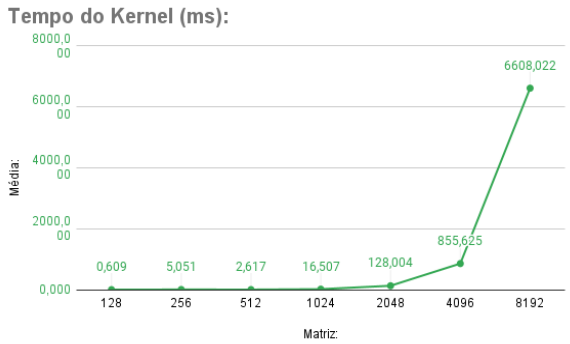


Figura 2: Tempo de trans. no SDumont.

A Figura 1 exibe os tempos de transferência na máquina CENAPAD, enquanto a Figura 2 mostra os dados correspondentes ao SD. A análise das figuras revela que o tempo de transferência aumenta conforme o crescimento das dimensões das matrizes em ambas as arquiteturas. Entretanto, é notável que, no SD, os tempos de transferência são consistentemente superiores para as mesmas dimensões de matrizes quando comparados com a máquina CENAPAD.



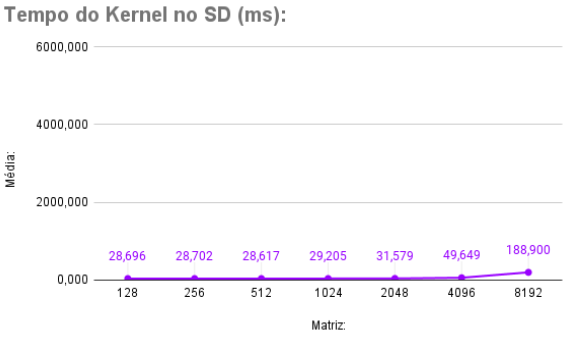
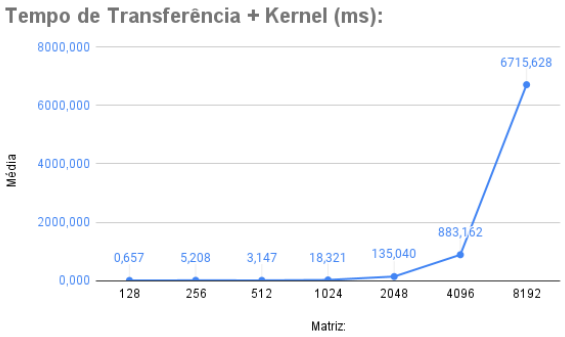


Figura 3: Tempo do Kernel na máquina do CENAPAD. ***Figura 4: Tempo do Kernel no SDumont.***

A Figura 3 exibe os tempos de execução do Kernel na máquina CENAPAD, enquanto a Figura 4 mostra os dados correspondentes ao SD. A análise das figuras revela que, na máquina CENAPAD, o tempo de execução do Kernel aumenta conforme as dimensões das matrizes crescem. No SD, no entanto, observa-se uma pequena oscilação entre as matrizes de dimensões 128 e 512. É notável que, no SD, os tempos de execução do Kernel são consistentemente superiores para matrizes de dimensões menores, mas para matrizes de dimensões maiores, os tempos do Kernel no SD são significativamente menores quando comparados à máquina CENAPAD.



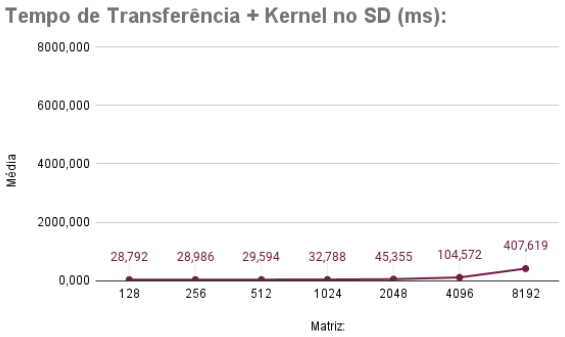


Figura 5: Tempo de Trans.+Kernel na máquina do CENAPAD. Figura 6: Tempo de Trans.+Kernel no SDumont.

A Figura 5 apresenta os tempos de transferência e execução do Kernel na máquina CENAPAD, enquanto a Figura 6 exibe os dados correspondentes ao SD. A análise das figuras revela que o tempo de transferência e execução do Kernel aumenta conforme o crescimento das dimensões das matrizes em ambas as arquiteturas. É notável que, no SD, os tempos de transferência e execução do Kernel são consistentemente superiores para matrizes de dimensões menores, mas para matrizes de dimensões maiores, os tempos de transferência e execução do Kernel no SD são significativamente menores quando comparados à máquina CENAPAD.

5. Conclusão e trabalhos futuros

A análise comparativa de desempenho entre as GPUs GeForce GTX 1660 e NVIDIA V100 revelou insights importantes sobre suas capacidades e adequação para diferentes tipos de tarefas.

Os experimentos realizados, utilizando a biblioteca cuBLAS e a função cublasGemmEx para multiplicação de matrizes, evidenciaram as diferenças de desempenho entre as duas GPUs. A GeForce GTX 1660 teve um desempenho adequado para matrizes de menor dimensão, enquanto a NVIDIA V100 superou significativamente em matrizes de grandes dimensões, confirmando sua eficiência em operações de álgebra linear complexas. No entanto, observou-se um resultado inesperado no tempo de transferência, que foi menos eficiente na V100, mesmo com sua tecnologia mais avançada.

Esses resultados são fundamentais para entender as limitações e potencialidades de cada arquitetura, contribuindo para a otimização de desempenho em supercomputadores e avanços nas técnicas de computação paralela. A escolha da GPU ideal deve considerar o tipo de aplicação e os requisitos específicos de desempenho, garantindo assim o melhor aproveitamento dos recursos disponíveis.

Como trabalho futuro, pretendemos nos aprofundar nas arquiteturas e investigar por que o tempo de transferência foi menos eficiente no SDumont, apesar da tecnologia mais avançada da V100.

6. Referências

NVIDIA. cuBLAS Library Samples. Disponível em: <https://github.com/NVIDIA/CUDALibrarySamples/tree/master/cuBLAS/Extensions/GemmEx>. Acesso em: 21 março 2024.

NVIDIA. Compare GeForce Graphics Cards. Disponível em: <https://www.nvidia.com/pt-br/geforce/graphics-cards/compare/?section=compare-16>. Acesso em: 18 junho 2024.

NVIDIA. cuBLAS Documentation. Disponível em: <https://docs.nvidia.com/cuda/cublas/#cublasgemmex>. Acesso em: 16 maio 2024.

NVIDIA. NVIDIA V100 Tensor Core GPU. Disponível em: <https://gzhls.at/blob/ldb/0/d/a/3/8c61e1573a93d7326dd3df2e255305da5853.pdf>. Acesso em: 24 junho 2024.

SDumont. *Manual de Suporte*. Disponível em: <https://sdumont.lncc.br/support_manual.php?pg=support#.> Acesso em: 23 maio 2024.

Relatório de Atividades

TÍTULO DO PROJETO: Estudo em GPU: Multiplicação de Matrizes Esparsas x Densas

Bolsista: Mariana Aguiar Ribeiro

Orientador: Carla Osthoff e Micaella Coelho

Tipo de bolsa: PIBIC

Período do relatório: Julho de 2023 a Julho 2024

1. Objetivo

O projeto tem como objetivo fazer um estudo sobre o ambiente do Supercomputador Santos Dumont e através da comparação do processamento da multiplicação de matrizes esparsas por matrizes densas em uma GPU local, GeForce GTX 1660, e na GPU NVIDIA Tesla V100 presente no supercomputador, realizar uma análise de desempenho.

2. Introdução

Este relatório mostra o estudo feito sobre o ambiente do Supercomputador Santos Dumont em sua totalidade, por meio da implementação da biblioteca cuSPARSE da NVIDIA na otimização de código em CUDA de multiplicação de matrizes esparsas e matrizes densas. O estudo foi feito mediante comparação do processamento do código de multiplicação em uma GPU local, GeForce GTX 1660, que apresenta a arquitetura Turing, e da GPU NVIDIA Tesla V100, que se localiza no nó sequana do Santos Dumont e tem como arquitetura a Volta.

A biblioteca “cuSPARSE”, foi desenvolvida pela NVIDIA para GPUs, em que seu foco são operações matemáticas em matrizes esparsas, onde a maioria de seus elementos são zeros. Ela faz parte do CUDA Toolkit, que são ferramentas que permitem a execução de aplicações que usam o processamento paralelo das GPUs da NVIDIA. A biblioteca utiliza o formato “CSR”, ou, *Compressed Sparse Row*, sendo uma forma eficiente de armazenar as matrizes esparsas. No estudo a rotina aplicada foi a “cusparseSpMM”, que faz a multiplicação de matrizes esparsas por densas, as densas têm a maioria de seus elementos diferentes de zero.

A Arquitetura de Turing, utilizada na GeForce GTX 1660, foi introduzida pela NVIDIA em 2018, tendo suas características principais como a primeira arquitetura a introduzir núcleos Ray Tracing, o que auxilia a ter gráficos mais realistas. Sua memória GDDR6 possibilitou uma maior largura de banda e eficiência energética. Já a arquitetura Volta foi constituída em 2017, sendo a primeira a integrar núcleos Tensor, acelerando operações de matrizes. Bem como a utilização da memória HBM2, tendo uma largura de banda maior e seu consumo de energia é ainda menor que o da GDDR. E, apresenta uma arquitetura unificada, juntando a computação e processamento gráfico, ocorrendo uma otimização.

3. Metodologia

Inicialmente, foram feitos estudos sobre a programação em CUDA e o ambiente do Santos Dumont através de aulas do Programa de Verão do LNCC, ministradas pelo professor Roberto Pinto e professora Denise Stringhini. Após, ocorreu a instalação da VPN do LNCC em uma máquina de uso pessoal e conexão com a máquina "146.134.20.103", que tem a GPU GeForce GTX 1660, através do mobaXTERM.

Ao fazer o download da biblioteca "CUDALibrarySamples" e execução da "cuSPARSE", foi possível implementar o código "spmm_csr_example.c" no terminal LINUX, o qual faz a multiplicação de matrizes esparsas por matrizes densas.

Foi feito o estudo sobre a biblioteca cuSPARSE e função cusparseSPMM usada no código, além da criação dos arquivos da Matriz A (ESPARSA) e da Matriz B (DENSA) utilizando um código gerador chamado "gerador.c", onde foi necessário selecionar o tipo de matriz (esparsa ou densa), depois o número de linhas e colunas, foram criadas matrizes com os seguintes tamanhos: 128, 256, 512, 1024, 2048, 4096 e 8192, e por fim, qual a densidade desejada, neste caso 30%.

Os últimos passos foram a conversão das matrizes esparsas do arquivo ".mtx" para ".txt", o qual eram gerados quatro arquivos: um para seus valores, dimensões, linhas e colunas. E a execução do código "spmm_csr_example.c" cinco vezes, para a multiplicação das matrizes esparsas e densas criadas anteriormente, sendo assim possível obter uma média das execuções para uma melhor análise.

3.1. Ambiente Computacional

Foi usado uma máquina local com GPU GeForce GTX 1660 que apresenta 1408 CUDA Cores, memória de 6 GB, memória compartilhada de 64 KB e cache L2 de 1 MB. Além disso a CPU utilizada nesse ambiente foi a Intel(R) Core(TM) i7-10700 CPU @ 2.90GHz.

No Santos Dumont foi utilizada a GPU Tesla V100, com poder de processamento paralelo, 5120 CUDA cores, 640 tenso cores, 32 GB de memória, 96 KB de memória compartilhada e 6MB de cachê L2. E a CPU utilizada foi a Intel Xeon Skylake 6252.

4. Resultados e Discussões

Os resultados das multiplicações de matrizes foram coletados e organizados em gráficos para uma melhor visualização. São eles: tempo de transferência, tempo do kernel e tempo do kernel+transferência, todos em milissegundos.

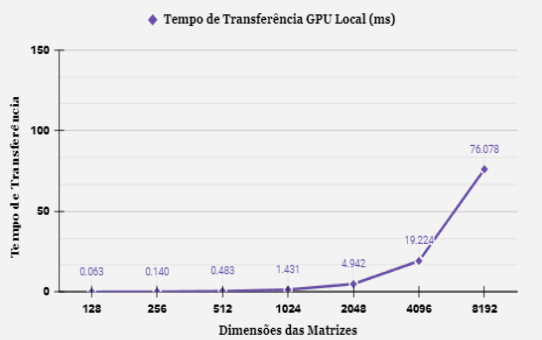


Figura 1: Tempo de Transferência na GPU Local.

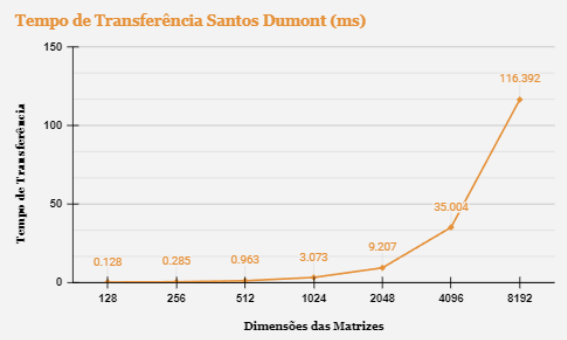


Figura 2: Tempo de Transferência no Santos Dumont.

As figuras 1 e 2 mostram o tempo de transferência, sendo ele o tempo que leva para a CPU e GPU trocarem os dados, onde o eixo x representa o tamanho das matrizes e o eixo y a média do tempo em milissegundos em que a multiplicação delas foram executadas. A figura 1 ilustra o desempenho da GPU Local, onde é possível observar que ao aumentar o tamanho das matrizes o tempo também cresceu. O mesmo ocorreu na execução no Santos Dumont, representada pelo gráfico 2, também é notável que o tempo nele foi maior do que na GPU Local pelo seu nível de complexidade.

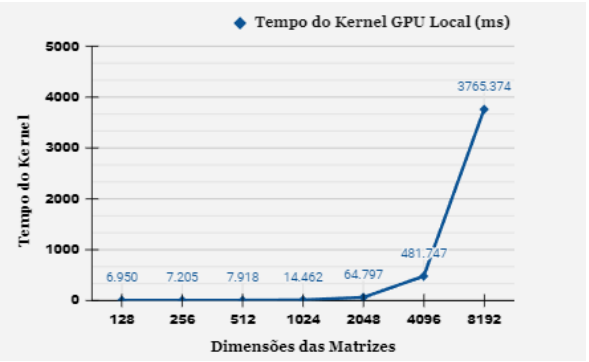


Figura 3: Tempo do Kernel na GPU Local.

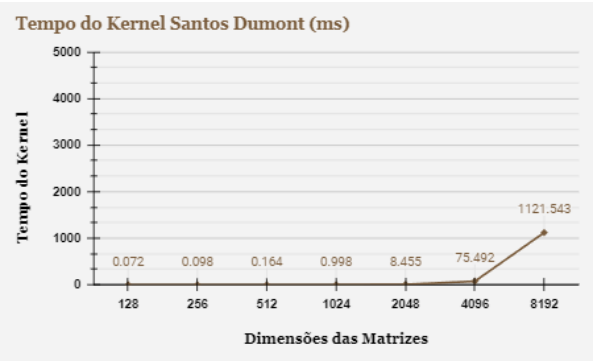


Figura 4: Tempo do Kernel no Santos Dumont.

As figuras 3 e 4 representam o desempenho do tempo do Kernel, o qual é rodado na GPU. Seguindo a mesma lógica dos gráficos 1 e 2, ao aumentar o tamanho da matriz seu tempo de execução também cresceu. A figura 3 representa a execução na GPU Local, já a figura 4 representa a execução no Santos Dumont, sendo ela a mais rápida.

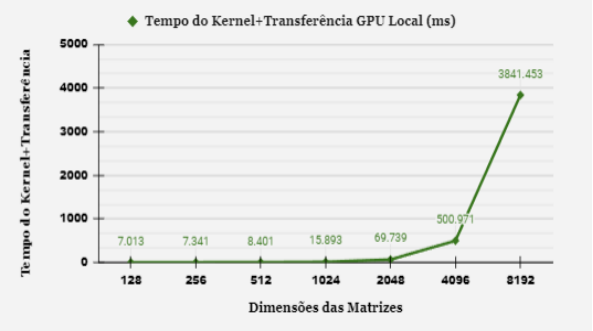


Figura 5: Tempo do Kernel+Transferência na GPU Local.

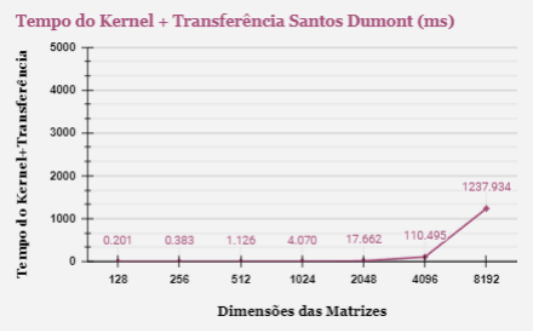


Figura 6: Tempo do Kernel+Transferência no Santos Dumont

As figuras 5 e 6 representam o tempo do Kernel+Transferência que é feito também entre CPU e GPU, sendo a soma dos dois. Contendo as mesmas características dos gráficos anteriores, percebemos que a execução no Santos Dumont (figura 6) também se destacou, sendo mais rápida e eficiente que a execução na GPU Local (figura 5).

5. Conclusões e Estudos Futuros

Diante do exposto, foi possível analisar que ao programar o mesmo código de multiplicação de matrizes densas por esparsas em duas GPUs distintas foram obtidos resultados de desempenho distintos. A GPU Local, GeForce GTX 1660 teve uma demora maior para gerar e executar a multiplicação de matrizes esparsas por densas, enquanto a GPU Tesla V100 foi mais rápida, mostrando seu importante papel na realização de cálculos matemáticos mais complexos, por seu processamento paralelo. Entretanto, foi observado que o tempo de transferência da V100 foi maior que o da GeForce GTX 1660, isso ocorreu por causa de sua complexidade e arquitetura sendo otimizada para outras tarefas.

Os próximos passos serão de suma importância, uma vez que será feito um estudo aprofundado sobre a causa da demora do tempo de transferência da GPU V100 ao multiplicar as matrizes, adquirindo uma maior compreensão sobre a arquitetura da GPU e seu funcionamento. Além de, implementar um código base de multiplicação de matrizes, não otimizado, para uma comparação com o código “spmm_csr_example.c” otimizado pela NVIDIA, também aumentando o entendimento sobre programação, multiplicação de matrizes e sobre o ambiente Santos Dumont, uma vez que ele será utilizado.

6. Referências Bibliográficas

ANTUNES FILHO, Amauri; XAVIER, Antunes Cesar Candido. Solução de sistemas lineares esparsos utilizando CUDA: uma comparação de desempenho em sistemas Windows e Linux. Revista de Ciências Exatas e Tecnologia, v. 8, n. 8, p. 181-195, 2013.

NVIDIA. Blog. Disponível em: https://developer.nvidia.com/blog. Acesso em: 10 jul. 2024.

NVIDIA. Volta GPU Architecture. Disponível em: https://www.nvidia.com/en-us/data-center/volta-gpu-architecture/. Acesso em: 10 jul. 2024.

NVIDIA. Turing Architecture. Disponível em: https://www.nvidia.com/en-us/geforce/turing/. Acesso em: 10 jul. 2024.

NVIDIA. cuSPARSE Library. Disponível em: https://docs.nvidia.com/cuda/cusparse/contents.html. Acesso em: 10 jul. 2024.

CORNELL VIRTUAL WORKSHOP. GPU Memory. Disponível em: https://cvw.cac.cornell.edu/gpu-architecture/gpu-memory/appendix. Acesso em: 10 jul. 2024.

GITHUB. CUDALibrarySamples: spmm_csr. Disponível em: https://github.com/NVIDIA/CUDALibrarySamples/tree/master/cuSPARSE/spmm_csr . Acesso em: 10 jul. 2024.

NVIDIA. Compare GeForce Graphics Cards. Disponível em: https://www.nvidia.com/en-us/geforce/graphics-cards/compare/?section=compare-16. Acesso em: 10 jul. 2024.

LABORATÓRIO NACIONAL DE COMPUTAÇÃO CIENTÍFICA. Suporte Manual do Santos Dumont. Disponível em: https://sdumont.lncc.br/support_manual.php?pg=support. Acesso em: 10 jul. 2024.

RELATÓRIO DE PROJETO DE INICIAÇÃO CIENTÍFICA

BIOINFORMÁTICA, BANCO DE DADOS E ENGENHARIA DE COMPUTAÇÃO

Título do Projeto Proposto

Comparação do Desempenho Computacional de *Workflows* Científicos de Transcriptômica em Arquiteturas HPC

Instituição

Laboratório Nacional de Computação Científica

Nome do Aluno

Reiglan Soares Di Lourenço

Nome do Orientador

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

D.Sc. Carla Osthoff Ferreira de Barros (Tecnologista Sênior – SEPAD/COTIC//LNCC, Coorientador)

Tipo de bolsa: PIBIC

Período do relatório: 28/06/2023 - 28/06/2024

1.Objetivo

O objetivo deste estudo é realizar uma análise computacional comparativa das arquiteturas MESCA2 e Ivy Bridge (Nvidia) do supercomputador Santos Dumont (SDumont) a respeito da memória compartilhada e distribuída usando como caso de estudo aplicações de bioinformática, mais especificamente o desempenho do *workflow* de transcriptômica ParslRNA-Seq.

2. Introdução

Os experimentos de bioinformática são complexos e exigem alto custo computacional, necessitando de tecnologias especializadas como *workflows* científicos, sistemas de gerência, e-Science, aprendizado de máquina e computação de alto desempenho (CAD). Gerenciar grandes volumes de dados de bioinformática por meio da modelagem de *workflows* científicos e adaptá-los a diferentes arquiteturas computacionais de CAD é um desafio significativo.

Workflows científicos de bioinformática são complexos porque refletem a adaptação de casos reais em áreas como biologia, astronomia e física, especialmente quando lidam com dados volumosos, heterogêneos ou multi-distribuídos. Esses *workflows* são mais bem suportados quando acoplados a tecnologias especializadas de Sistemas de Gerência de *Workflows* Científicos (SGWfC) e arquiteturas de CAD. Desenhar um *workflow* requer compreensão do problema biológico a ser resolvido, pois a natureza dos dados guiará a modelagem e adaptação na arquitetura computacional mais adequada.

O *workflow* de transcriptômica ParslRNA-Seq[1] para análise de expressão diferencial de genes (EDG) foi utilizado como estudo de caso na nossa análise de desempenho. Projetado para gerenciar grandes volumes de dados, ele foi modelado e gerenciado com a biblioteca Parsl em ambientes de CAD que facilita a modelagem, integração e automatização do *workflow*. As atividades do ParslRNA-Seq incluem Bowtie para mapear leituras, Sort para ordenar, Split para dividir arquivos,

[1] https://github.com/lucruzz/RNA-seq

HTSeq para contar leituras, Merge para indexar contagens e DESeq para análise estatística de EDGs. Os software que executam as atividades Bowtie2 e Sort (SAMTools) oferecem execução paralela, aumentando a eficiência e escalabilidade em ambientes de CAD.

3. Background

3.1 ParslRNA-Seq: *Workflow* Científico de Transcriptômica

ParslRNA-Seq (Silva et a 2021) é um *workflow* científico de transcriptômica para estudos de expressão diferencial de genes (EDG) modelado com a linguagem de programação Parsl e ser acoplado em ambientes de CAD do supercomputador SDumont. A Figura 1 apresenta o modelo conceitual do ParslRNA-Seq. Análises de EDGs são muito usadas na análise de dados transcriptômicos, permitindo a identificação de genes com expressão significativamente divergente entre condições experimentais. Essa abordagem serve para descobertas de novos transcritos e isoformas de genes, proporcionando *insights* importantes sobre processos biológicos complexos, como diferenciação celular, resposta a estímulos e desenvolvimento de doenças.

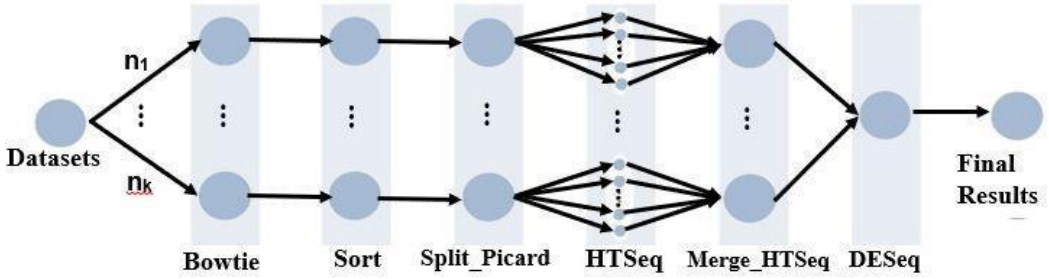


Figura 1. Modelagem Conceitual do *Workflow* Científico ParslRNA-Seq (6 atividades).

3.2 Parsl - Parallel Scripting Library

ParslRNA-Seq é implementado utilizando Parsl, uma ferramenta em Python projetada para execução de *workflows* em ambientes de CAD [Babuji et al., 2019]. Parsl[2] simplifica a criação de *workflows* ao permitir a integração de comandos externos diretamente no código Python através de decoradores (Apps), o que facilita o controle de sincronização das atividades. O motor de execução do Parsl é flexível, suportando diversos ambientes computacionais e abstraindo as complexidades do *workflow*, tornando mais simples sua implementação e integração com recursos computacionais.

O *workflow* utiliza dois modelos de execução: ThreadPoolExecutor e HTEx. ThreadPoolExecutor[3] é um modelo de execução que suporta *multithreading* em recursos locais, gerenciando um *pool* de *threads* para executar atividades de forma concorrente. A eficiência desse modelo é crucial para maximizar o desempenho computacional, reduzir erros e melhorar a produtividade [Silva e Yokoyama, 2011]. Por outro lado, o modelo HTEx permite a execução do *workflow* em múltiplos nós computacionais simultaneamente, facilitando o compartilhamento de dados entre as máquinas e proporcionando um controle refinado sobre a alocação de recursos e paralelismo necessário para uma execução eficiente das tarefas.

[2] https://parsl-project.org/
[3] https://docs.python.org/3/library/concurrent.futures.html.

4. Metodologia

4.1 Dados do Experimento

Os dados pertencem a um experimento real de RNA-Seq, extraídos do repositório público *Gene Expression Omnibus*[4] (GEO) e divididos em: (1) grupo de controle: SRR5445794, SRR5445795, SRR5445796 e (2) grupo de condições das vias Wnt: SRR5445797, SRR5445798, SRR5445799. O organismo é *Mus musculus* e o GEO.ID é GSE97763 (Plataforma Illumina HiSeq 2000 - *Mus musculus*).

As leituras de sequência foram alinhadas ao genoma de referência do rato (UCSC versão mm9) com Bowtie2. Para cada gene, o número de leituras mapeadas foi contado com Hhtseqcount. DESeq2 analisou a EDG a partir das matrizes das contagens do alinhamento e do mapeamento das sequências frente ao genoma de referência. Essas matrizes (arquivo GTF) contém o número de leituras que foram alinhadas de forma única (colunas) com os exames de cada gene nas amostras (colunas). Os seis arquivos de entrada totalizam 13GB e geram arquivo de saída de aproximadamente 74 GB.

4.2 O Ambiente Computacional Santos Dumont

O SDumont[5] 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. Todos os *softwares*, algoritmos, dependências de bioinformática (Bowtie, Samtools, Picard, HTSeq e DESeq2) e os componentes do Parsl foram alocados e instalados no ambiente do SDumont.

No presente experimento, utilizamos a máquina MESCA2, que possui 16 CPUs Intel Xeon Ivy Bridge totalizando 240 núcleos e 6 TB de memória RAM, caracterizada por uma arquitetura de memória compartilhada de alto desempenho, ideal para aplicações que requerem acesso rápido e simultâneo a grandes volumes de dados. O ambiente computacional do SDumont utilizado nas nossas análises é um sistema de clusters com memória distribuída, onde cada nó conta com 2 CPUs Intel Xeon E5-2695v2 (24 núcleos no total) e 64 GB de RAM DDR3, suportando tarefas segmentáveis e distribuídas entre nós através de rede Infiniband EDR de 100 Gb/s, sendo econômico e escalável horizontalmente. As execuções do experimento original, conforme publicadas por Silva et al. (2021)[6], foram realizadas em nós computacionais com 2 CPUs Intel Xeon E5-2695v2 Ivy Bridge, 24 núcleos (12 por CPU) e 64 GB de RAM. Todos os demais parâmetros do experimento foram mantidos iguais ao experimento original.

4.3 Arquitetura de memória compartilhada e distribuída no SDumont

Optamos por comparar dois tipos distintos de memória para avaliar sua eficiência e desempenho na execução do *workflow*. A memória compartilhada do MESCA2 oferece 240 núcleos e 6 TB, permitindo acesso direto e rápido aos dados entre os processadores, ideal para tarefas intensivas que demandam comunicação rápida e sincronização sem complexidade externa. Por outro lado, o Nvidia Ivy Bridge possui 64 GB de RAM por nó, configurado para tarefas segmentadas que não exigem compartilhamento extensivo de dados entre nós. Cada nó utiliza sua própria memória local, com

[4] https://www.ncbi.nlm.nih.gov/geo/
[5] https://sdumont.lncc.br/
[6] https://doi.org/10.5753/bresci.2021.15789

comunicação entre nós via rede, o que pode resultar em latências maiores comparado ao acesso direto da memória compartilhada.

O processador Intel Ivy Bridge, com clock de 2,4 GHz, possui uma arquitetura x86-64 de 64 bits, 15 núcleos por CPU, fabricado com tecnologia de 22 nm, e suporta Hyper-Threading, permitindo 2 *threads* por núcleo. Ele possui até 30 MB de cache L3. O nó de computação MESCA2, com memória compartilhada de grande capacidade (fat node), permite que múltiplos núcleos acessem a mesma região de memória, utiliza cache em três níveis (L1, L2 e L3), e possui suporte à NUMA para otimizar a latência de acesso à memória. A interconexão de alta velocidade e técnicas avançadas de gerenciamento de memória, como prefetching e cache coherence, garantem eficiência e desempenho elevado, além de suportar grandes quantidades de RAM para cargas de trabalho intensivas em dados. O prefetching antecipa a necessidade de dados, carregando-os para o cache antes de serem requisitados, enquanto a cache coherence assegura que todas as cópias dos dados em diferentes caches sejam consistentes, evitando conflitos e garantindo a integridade dos dados durante o processamento paralelo.

Em sistemas de memória distribuída, cada nó possui sua própria memória e cache local, acessíveis diretamente pelo processador do nó. A coerência de cache local é mantida dentro de cada nó, garantindo que as operações de leitura e escrita sejam consistentes para os dados armazenados localmente. No entanto, para manter a consistência entre os caches de diferentes nós, é necessário o uso de protocolos explícitos de comunicação, como a passagem de mensagens. Quando um nó precisa acessar dados que residem em outro nó, esses dados são transferidos pela rede e armazenados temporariamente no cache local do nó solicitante, o que ajuda a reduzir a latência de acesso. As políticas de substituição de cache, como o LRU (Least Recently Used), são aplicadas localmente em cada nó para gerenciar o conteúdo do cache, priorizando a remoção dos dados menos recentemente utilizados quando necessário liberar espaço para novos dados ou atualizações.

4.4 Metodologia sobre o desempenho computacional do *workflow* em 1 nó MESCA2

O experimento segue os seguintes passos:

Reexecução do workflow de Silva et al. (2021) em um único nó com 2 CPUs Intel Xeon Ivy Bridge de 2.4GHz, 64 GB de RAM e 24 núcleos.

1. Na presente pesquisa, o experimento foi realizado utilizando 16 CPUs Intel Ivy Bridge de 2.4GHz, totalizando 240 núcleos (15 por CPU) e 6 TB de memória RAM compartilhada.

Foram testadas configurações de 1, 2, 4, 6, 12, 18 e 24 threads, semelhantes ao experimento original, para avaliar o desempenho computacional no MESCA2.

5. Resultados e Discussão

Os resultados apresentados incluem uma comparação entre a análise original do workflow de Silva et al. (2021), realizada no Ivy Bridge (Nvidia) com 24 threads, e as novas análises feitas no MESCA2 com 240 threads.

5.1 Resultados de Desempenho do *workflow* em 1 nó MESCA2

Os resultados avaliam se o MESCA2 executando o *workflow* de forma mais eficiente do que o Nvidia, devido à sua maior capacidade computacional. Serão apresentados dois gráficos com curvas de

desempenho mostrando o tempo total de computação (TTE) no nó computacional com seus respectivos tipos de memória distribuída (Nvidia) Figura 2 e compartilhada (MESCA2) na Figura 3.

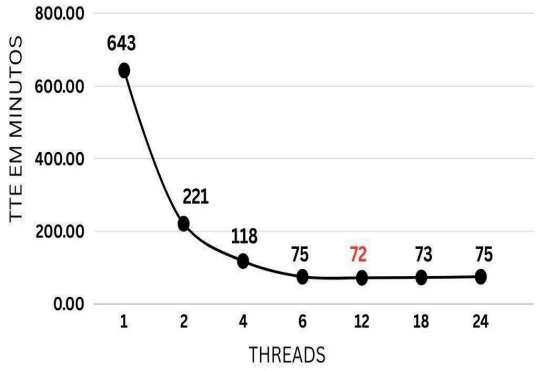


Figura 2 TTE do *workflow* em um nó Nvidia

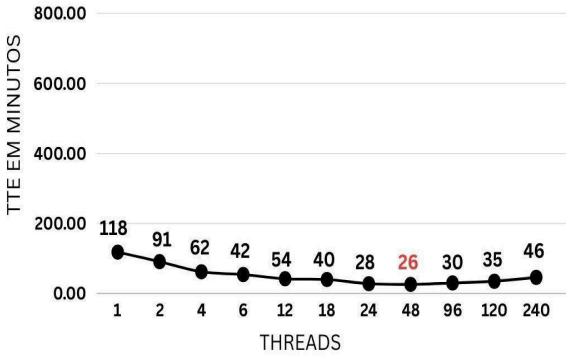


Figura 3 TTE do *workflow* em um nó Mesca2

A Figura 2 exibe o TTEdo *workflow* em um nó Nvidia com 24 *threads* (Silva et al., 2021), enquanto a Figura 3 mostra a execução no MESCA2 de 240 *threads*. A comparação inicial é feita com 1 e 24 *threads* em ambos os casos. Observamos que o MESCA2 superou o Nvidia, com uma melhoria de 81% na comparação com apenas 1 *thread* e uma redução de 61% no tempo com 24 *threads*. Além disso, o MESCA2 alcançou seu melhor desempenho utilizando 48 *threads*, com um tempo total de execução de 26 minutos. Após esse ponto, houve um aumento no tempo devido à falta de trabalho disponível para os *threads* adicionais.

Essa melhora no TTE na Figura 3 em comparação à Figura 2 também se deve ao tipo de memória utilizado. A memória compartilhada do MESCA2 é mais eficiente que a memória distribuída do nó Nvidia. O volume de dados processados entre os processadores têm uma eficiência maior quando a memória permite acesso mais rápido. A memória compartilhada do MESCA2 facilita uma comunicação e sincronização mais rápida entre os núcleos, resultando em um melhor desempenho no processamento de dados do *workflow* testado.

6. Conclusão

Os resultados obtidos nos experimentos destacam claramente a superioridade do MESCA2 em relação ao Ivy Bridge (Nvidia) para a execução do *workflow* de transcriptômica. O MESCA2 demonstrou uma significativa redução no tempo de execução em comparação com o Nvidia, especialmente notável ao utilizar 48 *threads*, onde alcançou seu melhor desempenho.

A eficiência da memória compartilhada do MESCA2 também se mostrou superior à memória distribuída do Nvidia nos testes realizados. Enquanto o Nvidia enfrentou limitações significativas mesmo com ajustes na quantidade de *threads*, o MESCA2 foi capaz de otimizar a utilização de recursos e melhorar substancialmente o tempo de execução.

As variações de desempenho observadas entre os dois sistemas reforçam a importância de selecionar a arquitetura de *hardware* adequada para aplicações que demandam alto desempenho e paralelismo intensivo. Considerando isso, os próximos passos incluem a continuidade da otimização do *workflow* de transcriptômica para diferentes ambientes computacionais, visando melhorar ainda mais a eficiência e a escalabilidade do processamento de dados genômicos.

Esses esforços não apenas beneficiarão pesquisadores e cientistas na análise de grandes conjuntos de dados biológicos, mas também contribuirão para o avanço da bioinformática em instituições como o LNCC e outras parceiras deste projeto.

7. Referências bibliográficas

[1] V. Marx, “Biology: The big challenges of big data,” Nature, vol. 498, no. 7453, pp. 255–260, Jun. 2013.

[2] J. Freire, D. Koop, and L. Moreau, Eds., Provenance and Annotation of Data and Processes, vol. 5272. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[3] M. Mattoso et al., “Towards supporting the life cycle of large scale scientific experiments,” International Journal of Business Process Integration and Management, vol. 5, no. 1, pp. 79–92, 2010.

[4] T. J. Hastie, R. J. Tibshirani, and J. H. Friedman, The elements of statistical learning: data mining, inference, and prediction, 2. ed., At 7. printing. New York, NY: Springer, 2013.

[5] G. Da San Martino and A. Sperduti, “Mining Structured Data,” IEEE Computational Intelligence Magazine, vol. 5, no. 1, pp. 42–49, Feb. 2010.

[6] L. Hirschman, J. C. Park, J. Tsujii, L. Wong, and C. H. Wu, “Accomplishments and challenges in literature data mining for biology,” Bioinformatics, vol. 18, no. 12, pp. 1553–1561, Dec. 2002.

[7] Cruz, L.; Coelho, M.; Terra, R.S.; Carvalho, D.; Gadelha, L.M.R.; Osthoff, C.; Ocaña, K.A.C.S. Workflows Científicos de RNA-Seq em Ambientes Distribuídos de Alto Desempenho: Otimização de Desempenho e Análises de Dados de Expressão Diferencial de Genes. In: Brazilian e-Science Workshop (BreSci 2021), 2021, Florianópolis, Santa Catarina. Anais do XV Brazilian e-Science Workshop. 2021.

[8] Cruz, L.; Coelho, M.; Gadelha, L.M.R.; Ocaña, K.A.C.S.; Osthoff, C. Avaliação de Desempenho de um Workflow Científico para Experimentos de RNA-Seq no Supercomputador Santos Dumont. In: Workshop de Iniciação Científica em Arquitetura de Computadores e Computação de Alto Desempenho (WSCAD 2020 - WIC), 2020. Anais do Workshop de Iniciação Científica em Arquitetura de Computadores e Computação de Alto Desempenho,2020.

[9] Ocaña, K.; Cruz, L.; Galheigo, M.; Coelho, M.; Carneiro, A.; Terra, R.; Gadelha, L.; Carvalho, D.; Boito, F.;Navaux, P.; Osthoff, C. ParslRNA-Seq: A scalable, efficient, and high-throughput RNAseq analysis workflow in supercomputers. IEEE/ACM Transactions on Computational Biology and Bioinformatics. 2023.

RELATÓRIO

ESTUDANTE IC
Ricardo dos Santos de Lima

ORIENTADOR
Regina Célia Cerqueira de Almeira

TÍTULO DO PROJETO DO ORIENTADOR
Introdução à Programação com o Scratch e Tutoriais

1. OBJETIVOS E METAS DO PROJETO DO ALUNO

O projeto tem como objetivo principal a elaboração de tutoriais, utilizando o software Scratch[1], para ensinar noções básicas de programação para estudantes de Ensino Fundamental e Médio. Ao todo, pretende-se elaborar trinta tutoriais abordando diferentes tópicos, envolvendo noções de programação e desenvolvimento de jogos. Ao término do desenvolvimento, o material será disponibilizado para acesso gratuito na página web do projeto. Com uma metodologia que combina instruções teóricas e atividades práticas, os tutoriais percorrem tópicos como lógica de programação, estruturas de dados e modelagem computacional.

A abordagem do projeto incluiu o desenvolvimento de recursos pedagógicos para facilitar a compreensão dos conceitos fundamentais de programação, bem como a criação de exercícios práticos para fixação dos conceitos apresentados. A ordem dos conteúdos apresentados nos tutoriais propicia o aprendizado gradual, iniciando pela apresentação dos recursos disponíveis no software Scratch, associados aos recursos de programação. Ao finalizar os tutoriais, o estudante será apresentado a noções de desenvolvimento de jogos, sendo a abordagem distribuída nos tutoriais. Ao término dos tutoriais, espera-se que os estudantes

[1] https://scratch.mit.edu/

possam estar aptos a iniciar o desenvolvimento de seus próprios projetos, desenvolvendo habilidades de pensamento lógico e resolução de problemas. Para além, durante a execução dos tutoriais, outras atividades podem ser associadas promovendo atividades em grupo e compartilhamento de ideias entre os alunos.

2. PRINCIPAIS ETAPAS EXECUTADAS PELO ALUNO NO PERÍODO

Etapa 1: Na primeira etapa do projeto foi realizada uma pesquisa bibliográfica para identificar e selecionar os principais conceitos relacionados à lógica de programação, estrutura de dados e modelagem computacional. Este levantamento foi importante para fundamentação e criação dos tutoriais, garantindo que o material desenvolvido seja abrangente, didático e que percorra por conteúdos fundamentais importantes para compreensão dos conteúdos fundamentais da programação. Um dos recursos utilizados mais específicos sobre a programação com Scratch, como “Scratch: Um jeito divertido de aprender programação” de H. Varela(2017) e “Oficina básica de Scratch” de Vieira, M. F., foram essenciais para estruturar os tutoriais, servindo como uma referência prática para a criação dos conteúdos didáticos.

Etapa 2: Baseando-se nas pesquisas bibliográficas, foram selecionados os conceitos essenciais para serem abordados nos tutoriais. Os principais tópicos incluem:

- Lógica de programação: são introduzidos os conceitos básicos de algoritmos e lógica, o uso de operadores aritméticos e lógicos, além de estruturas condicionais e de repetição.

- Estrutura de dados: são apresentados os tipos de dados básicos e operações, variáveis, listas e estruturas de controle de fluxo.

- Modelagem computacional, que percorre pela criação, manipulação de objetos e o uso de coordenadas para posicionamento de atores.

Etapa 3: Para assegurar que as atividades propostas nos tutoriais estivessem alinhadas com os objetivos educacionais e fossem adequadas para o público-alvo. Durante as reuniões foram discutidas as melhores abordagens pedagógicas. Foram realizadas as revisões dos conteúdos selecionados e também o ajuste das atividades conforme atendesse a necessidade dos estudantes. As reuniões foram importantes para discutir as abordagens adotadas e realizar ajustes na estrutura dos tutoriais para garantir que os materiais fossem eficazes para os professores utilizarem em sala de aula. Essa colaboração permitiu a criação de um plano estruturado e coeso.

3. PRINCIPAIS RESULTADOS OBTIDOS

Foi obtido um conhecimento sólido acerca das ferramentas e capacidades do software Scratch construído através de livros, vídeoaula na plataforma do YouTube e principalmente por intermédio das atividades práticas presentes nessas fontes, além das atividades presentes para a elaboração dos tutoriais que estavam no planejamento.

Ao final no período foram elaborados seis tutoriais que possuem como estrutura definida a apresentação do tópico abordado, descrição das atividades a serem elaboradas de forma sequencial, permitindo a reprodução em um segundo momento pelo estudante e a proposta de um desafio ao término. Os tutoriais elaborados durante o período abordaram os conteúdos:

- Tipos de variáveis e constantes existentes na programação. O objetivo é apresentar os conceitos relacionados à variável, exemplificando a sua aplicação e o uso na elaboração de um programa. A atividade prática aplica os conceitos apresentados no software Scratch. Ao término do tutorial, como atividade (desafio), o estudante deve implementar uma lista de variáveis.

- Os operadores aritméticos na programação e como funcionam no Scratch. O objetivo é descrever os diferentes operadores aritméticos e, em seguida, aplicar esses conceitos na criação de um jogo interativo. Inicialmente, os alunos aprendem a utilizar os operadores para desenvolver um jogo de adição. Ao final é desafiado a construir, de forma independente, um jogo de subtração. É explorado os operadores aritméticos básicos: adição, subtração, multiplicação e divisão. É visto como esses operadores podem ser utilizados para realizar cálculos simples e como aplicá-los dentro do ambiente de programação.

- As estruturas condicionais são descritas de formas detalhadas e aplicadas em duas atividades práticas ao longo do tutorial. A primeira atividade ensina aos estudantes a criar um algoritmo que solicite que o usuário insira um valor e, em seguida, verifica se esse valor é maior ou menor que dez. Esta atividade ajudará a entender como as estruturas condicionais podem ser utilizadas para tomada de decisões com base nos valores fornecidos. As outras atividades proporcionam uma compreensão mais aprofundada de como utilizar condições aninhadas para realizar verificações mais complexas. Os estudantes são desafiados ao final do tutorial a desenvolver um algoritmo que solicite dois números e determine se cada um deles é par ou ímpar, visando consolidar o aprendizado e encorajar a aplicação prática desses conceitos em contextos diferentes.

- Os operadores lógicos na programação é um conteúdo fundamental para o aprendizado, o tutorial descreve os conceitos relacionados ao tema, apresentando a tabela verdade para a compreensão de como esses operadores

funcionam. O estudante aprenderá a desenvolver a porta lógica “AND”, em que inclui a construção de exemplos práticos para visualização de como as portas lógicas operam e quais resultados elas produzem com diferentes combinações de entrada. Ao final do tutorial, os estudantes são desafiados a desenvolver a porta lógica “OR”, reforçando sua compreensão acerca dos operadores lógicos.

- As combinações de laços de repetição e estruturas condicionais na programação e no Scratch. O objetivo é reforçar os conceitos já abordados em tutoriais anteriores e apresentar as possibilidades criadas a partir da combinação dos conceitos. Este tutorial inclui três atividades práticas, inicialmente poderá ser desenvolvido um algoritmo que faz uma contagem de 1 a 20, informando quais números são pares e quais são ímpares.

  Em seguida, é elaborado um novo algoritmo que solicita ao usuário que insira dois números, A e B. Se A for maior que B, o algoritmo deverá criar um contador que inicie em A e termine em B. Caso contrário, o contador deverá iniciar em B e terminar em A. Se os números forem iguais, o algoritmo deverá retornar uma mensagem indicando que são iguais e que não é possível gerar contagem. Por fim, é criado um novo algoritmo que pede ao usuário cinco números e ao fim é retornada a soma deles.

  Essas atividades permitem a aplicação prática de laços de repetição e estruturas condicionais, como esses conceitos podem ser utilizados para criar programas dinâmicos, interativos e com controle de fluxo. Por fim, é dado o desafio para os estudantes criarem novos algoritmos.

- A manipulação de sons no Scratch é importante para o desenvolvimento de projetos mais complexos, neste tutorial, o objetivo é instruir a atribuir sons aos atores, enriquecendo a experiência interativa dos projetos elaborados pelos estudantes. Eles são instruídos a selecionar diferentes sons na biblioteca de sons do software e associá-los a ações específicas dos atores. Isso inclui sons para movimentos, colisões ou outros eventos que sejam relevantes dentro do projeto. Ao final, os estudantes possuirão habilidades necessárias para aprimorar seus programas, sendo desafiados a criar atores e adicionar sons a eles.

- O posicionamento dos atores em tela no Scratch. O objetivo deste tutorial é contribuir para compreensão do sistema de coordenadas do software e como posicionar os atores usando os valores dos eixos x e y. A atividade consiste em adicionar um ator ao palco e utilizar blocos de movimento para definir uma posição inicial e em seguida alterar a posição daquele ator, acompanhando o deslocamento horizontal e vertical na tela. Ao final, os estudantes são

desafiados a escolher um ator e fazer com que ele ocupe 4 posições diferentes no espaço.

4. DISCUSSÃO SUCINTA

O projeto de desenvolvimento de tutoriais em Scratch para introdução à programação é uma iniciativa importante para a educação contemporânea. Ao focar em alunos do Ensino Fundamental e Médio, proporciona uma base sólida em programação desde cedo, propiciando a formação de conhecimentos que poderá ser utilizado no cotidiano. A escolha do Scratch como plataforma é importante, pois sua interface visual é intuitiva e facilita a compreensão de conceitos complexos da programação. A metodologia proposta, que combina teoria e prática, é eficaz para o engajamento dos alunos e promoção da aprendizagem ativa. Ao aprenderem conceitos de lógica de programação, estrutura de dados e modelagem computacional, os alunos desenvolverão habilidades cruciais para o século atual, como pensamento crítico, resolução de problemas e criatividade.

A preparação de materiais para publicação online amplia o alcance do projeto, que permite que os recursos estejam acessíveis a um público maior e diverso. No mais, é importante considerar desafios como a necessidade de dispositivos adequados para todos os alunos, bem como a capacitação de professores para orientação e apoio ao aprendizado. A avaliação contínua do progresso dos alunos e a adaptação dos tutoriais conforme necessidades serão fundamentais.

5. DIFICULDADES ENCONTRADAS / CAUSAS E PROCEDIMENTOS PARA SUPERÁ-LAS

Algumas dificuldades encontradas ao longo deste período foram:

1. Dificuldade em planejar e estruturar tutoriais;
2. Pouca familiaridade com o software;
3. Escolha da abordagem/didática para explanação dos assuntos abordados nos tutoriais.

Alguns recursos gratuitos online foram importantes para a construção de um conhecimento prévio da plataforma Scratch, que foi aprimorado durante a elaboração dos tutoriais. A exploração de tutoriais disponíveis no próprio site do Scratch e vídeos educativos no YouTube permitiu a imersão inicial nos conceitos fundamentais. Além disso, a utilização de cursos disponíveis em plataformas de aprendizado, como disponibilizado por Guanabara (2019) em seu tutorial sobre criação de personagens e animações no Scratch, foi importante para entender como organizar o conteúdo de forma progressiva.

Para a estruturação dos tutoriais de forma eficaz, foi necessário importar uma abordagem que se desenvolvesse ao longo dos tutoriais, começando por conteúdos mais simples e fundamentais, avançando para tópicos mais complexos. O uso do conteúdo

roteirizado, como discutido por Varela (2017) em "Scratch: Um jeito divertidos de aprender programação", foi fundamental para o planejamento detalhado da estrutura dos tutoriais, garantindo uma progressão lógica e compreensível no ensino-aprendizagem da programação com Scratch.

6. REFERÊNCIAS BIBLIOGRÁFICAS

1. **VALENTE, José Armando et al.** O computador na sociedade do conhecimento. Campinas: Unicamp/NIED, p. 11-18, 1999.

2. **DA SILVA, Angela Carrancho.** Educação e tecnologia: entre o discurso e a prática. Revista Ensaio: Avaliação e Políticas Públicas em Educação, v. 19, n. 72, p. 527-554, 2011.

3. **NUNES, Felipe Becker et al.** Um estudo de caso sobre a importância do uso de objetos de aprendizagem no ensino fundamental como apoio pedagógico. In: Anais do Workshop de Informática na Escola. 2014. p. 542.

4. **LOPES, José Junio et al.** A introdução da informática no ambiente escolar. Clube do professor, v. 23, 2004.

5. **SEVERO, C. E. P. (2021).** Jogos com Scratch: em projetos práticos com linguagem de blocos. Brasil: Casa do Código.

6. **VARELA, H. (2017).** Scratch: Um jeito divertido de aprender programação. Brasil: Casa do Código.

7. **GUANABARA, Gustavo.** Como criar personagens, cenários e animações no Scratch 3.0 - Tutorial Completo. **2019.** Disponível em: https://www.youtube.com/watch?v=GrPkuk1ezyo&t=765s. Acesso em: 10 jul. 2024.

Salvador, 12 de JULHO de 2024.

Ricardo dos Santos de Lima

Estudante

Orientador (a)

LNCC Laboratório Nacional de Computação Científica

MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO

GOVERNO FEDERAL BRASIL UNIÃO E RECONSTRUÇÃO

Modelagem de vigas de Euler Bernoulli e Timoshenko

Equações Diferenciais Parciais

Bolsista: Tarsiane Ribeiro da Costa

Orientador: Jaime Ediberto Munoz Rivera

Bolsista de iniciação cientifica – CNPq

Período do desenvolvimento: 13/02/24 à 28/08/2024

I. Objetivo

Este projeto de iniciação científica tem como objetivo desenvolver um modelo matemático para a análise de vigas sujeitas a forças pontuais utilizando equações diferenciais de quarta ordem. As vigas são elementos estruturais fundamentais em engenharia civil e mecânica, e a compreensão precisa de seu comportamento sob diferentes tipos de cargas é crucial para garantir a segurança e a eficiência das estruturas.

Nessa etapa, usaremos forças aplicadas em intervalos específicos da viga, focando em vigas engastadas (fixas em ambas as extremidades). Observaremos o comportamento da viga conforme descrito na literatura e analisaremos detalhadamente os efeitos dessas forças pontuais sob as condições de contorno específicas de vigas engastadas. O objetivo é entender mais profundamente como essas forças influenciam a deflexão e outros parâmetros estruturais, contribuindo para uma análise mais precisa e abrangente do comportamento das vigas.

II. Introdução

O estudo das vigas e de seu comportamento sob cargas é um dos pilares da engenharia estrutural. Modelos matemáticos baseados em equações diferenciais permitem a análise precisa da deflexão, momento fletor, e outras características mecânicas das vigas. Neste projeto, focaremos em vigas sujeitas a forças pontuais, utilizando equações diferenciais de quarta ordem para descrever seu comportamento.

Nessa parte do projeto, submeteremos a viga a forças pontuais em apenas um intervalo específico, e para fins didáticos analisaremos três cenários possíveis:

Primeiro Cenário: A força será aplicada próxima à posição de engaste inicial. Espera-se observar uma deformação semelhante à Figura 1

(Figura de Engaste), onde a deflexão é mínima devido à proximidade do ponto de fixação.

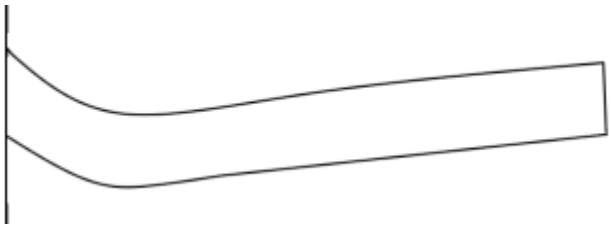


Figura 1: Primeiro cenário

Segundo Cenário: A força será aplicada a uma distância intermediária entre a posição de engaste inicial e o centro da viga. Neste caso, a deformação será semelhante à Figura 2 (Figura de Engaste), com uma deflexão maior do que no primeiro cenário, mas ainda menor do que a deflexão máxima.



Figura 2: Segundo cenário

Terceiro Cenário: A força será aplicada no centro da viga. Espera-se observar uma deformação simétrica e máxima, conforme ilustrado na Figura 3 (Figura de Engaste).

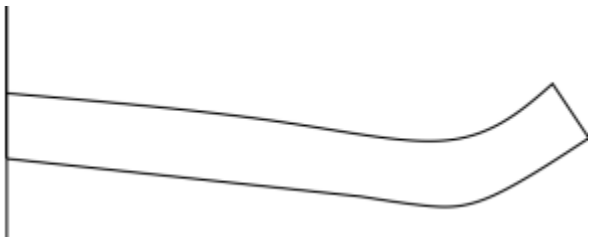


Figura 3: Terceiro cenário

Esses comportamentos são esperados porque, quanto mais próxima a força estiver da posição de engaste, menor será a deflexão. Por outro lado, quanto mais distante do ponto de fixação e mais próxima ao centro da viga, maior será a deflexão, com uma distribuição simétrica da deformação.

A análise detalhada desses cenários contribuirá para uma compreensão mais profunda dos efeitos das forças pontuais sobre as vigas engastadas, proporcionando uma base sólida para o desenvolvimento de modelos mais precisos e eficientes na engenharia estrutural

III. Material e Métodos ou Metodologia

A metodologia adotada neste estudo visa estudar a viga sendo submetida a uma força em um determinado intervalo dela, permitindo observar seu comportamento ao lidar com a força distribuída em apenas um intervalo específico e comparar com os demais cenários. Utilizaremos o software MATLAB para realizar os cálculos necessários e a plotagem dos gráficos.

Equação Governantes

Para descrever o comportamento da viga, utilizaremos a equação diferencial de quarta ordem:

$$u''''(x) = \frac{u(x+2h) - 4u(x+h) + 6u(x) - 4u(x-h) + u(x-2h)}{h^4}$$

Equação 1: Equação discretizada de quarta ordem

Procedimentos

Implementação do Modelo Matemático

Desenvolver o modelo matemático baseado na equação diferencial de quarta ordem, considerando as condições de contorno específicas para cada cenário.

Codificar o modelo matemático no MATLAB para permitir a análise e a solução das equações diferenciais.

IV. Resultado e Discussão

- Comparar os resultados obtidos nos três cenários para identificar como a posição da força influencia a deflexão e o momento fletor da viga.
- Validar os resultados com a literatura existente, verificando a precisão e a consistência dos modelos desenvolvidos.

V. Conclusões

Comportamento da Deflexão

A deflexão da viga engastada é fortemente influenciada pela posição da força aplicada. Conforme esperado, a deflexão é mínima quando a força está próxima ao ponto de engaste inicial e máxima quando aplicada no centro da viga.

Nos três cenários analisados, os resultados mostraram que quanto mais distante do ponto de engaste, maior é a deflexão observada. Este comportamento está alinhado com as previsões teóricas, confirmando a validade do modelo matemático utilizado.

Limitações e Trabalhos Futuros

Embora os resultados deste estudo sejam promissores, a análise foi limitada a forças pontuais em vigas engastadas. Trabalhos futuros podem expandir essa análise para incluir outras condições de contorno, diferentes tipos de cargas (distribuídas, por exemplo), e materiais variados.

Além disso, a investigação de efeitos dinâmicos e de não-linearidades materiais pode proporcionar uma compreensão ainda mais abrangente do comportamento das vigas sob diferentes condições de carga.

Considerações Finais

O estudo mostrou que o uso de equações diferenciais de quarta ordem, implementadas via MATLAB, é uma abordagem eficaz para a análise de vigas engastadas sob forças pontuais.

VI. Referências Bibliográficas

[1]. M. Renardy, R.C. Rogers; An Introduction to partial differential equations, Sringer-Verlag (1993) Vol. 1.

[2] R. V. Churchill; Fourier Series and boundary value problems, McGraw-Hill, New York (1988) Vol. I.

[3] J.E. Munoz Rivera; Introdução às Equações Diferenciais Parciais, LNCC Vol. 1, pages 1-230, (2004). ISBN 978-85-9996108-7

[4] J.E. Munoz Rivera; Teoria das distribuições e EDP, LNCC Vol. 1, pages 1-244, (2004). ISBN 85-99961-06-3

Laboratório Nacional de Computação Científica

Tipo de Bolsa e Período: Iniciação Científica — Dezembro 2022 - Agosto de 2024

Inteligência Artificial Aplicada ao Diagnóstico por Imagem

Bolsista: Matheus Molina Alves Lima

Orientador: Bruno Schulze

Petrópolis
2024

1 Objetivo

Este projeto tem como objetivo principal a aplicação de técnicas avançadas de Inteligência Artificial (IA) no diagnóstico médico, utilizando imagens de ressonâncias magnéticas, radiografias e/ou tomografias computadorizadas. Além disso, busca-se a integração da IA com a Computação Distribuída de Alto Desempenho, a fim de estabelecer uma infraestrutura computacional adequada para auxiliar na tomada de decisões médicas de forma ágil.

O foco central do projeto é proporcionar uma análise precisa e eficiente das imagens provenientes do banco de imagens, tanto de pacientes com comorbidades quanto de pacientes sem comorbidades. Isso inclui a avaliação de afecções, bem como o cálculo da área e percentual ocupado pelas mesmas.

Nessa etapa do projeto, o foco está no desenvolvimento e aprimoramento de novas funcionalidades, enquanto se continua com o treinamento e testes na rede neural. O objetivo é não apenas melhorar a precisão e eficiência dos resultados obtidos pela rede, mas também garantir que a tecnologia seja facilmente acessível ao público. Além disso, está sendo trabalhado para integrar a IA de forma mais eficaz nos processos de diagnóstico médico, facilitando seu uso por profissionais de saúde e proporcionando benefícios diretos aos pacientes.

2 Introdução

Com o objetivo de democratizar o acesso a ferramentas de diagnóstico assistido por inteligência artificial, foi iniciado o desenvolvimento de um portal web destinado à análise de imagens médicas. Este portal utiliza uma rede neural treinada para detectar tumores, oferecendo acesso gratuito ao público e permitindo que qualquer pessoa utilize a ferramenta como suporte na detecção de possíveis tumores na mama. Simultaneamente, a técnica de Transferência de Aprendizagem foi explorada para expandir as capacidades da rede neural, possibilitando a identificação de diferentes tipos de tumores. Este trabalho representa um avanço significativo no campo do diagnóstico médico, proporcionando uma solução inovadora e de baixo custo que pode beneficiar tanto profissionais de saúde quanto pacientes [1].

3 Material e Métodos ou Metodologia

Juntamente com o treinamento da rede neural, foi iniciado dois projetos em paralelo para aprimorar a utilização da rede neural no diagnóstico médico:

Portal Web: Com o intuito de disponibilizar gratuitamente na internet, idealizamos um portal web para que qualquer pessoa possa utilizar a rede treinada para identificar tumores de mama em exames, seja ultrassonografia, ressonância magnética ou tomografia.

Transferência de Aprendizagem: Para expandir os tipos de tumores que a rede neural pode identificar, testamos a possibilidade de usar a biblioteca TensorFlow para treinar a rede a identificar exames semelhantes de outras partes do corpo, como o fígado, por exemplo [2].

Para a criação de um portal web, foi essencial aprofundar os conhecimentos em HTML, CSS, JavaScript e PHP, desenvolvendo uma maior afinidade com interfaces visuais e formas eficientes de criar o backend, junto com a necessidade de criar interações diretas entre cliente e servidor. Além disso, foi utilizado o Laravel, um framework PHP open-source, para criar uma API capaz de receber imagens submetidas por usuários, salvar esses arquivos localmente, processá-los através da rede neural e devolver o resultado, indicando a presença ou não de um tumor no exame [3][4][5].

Além disso, também foi realizado um estudo sobre o Laravel, que provou ser uma ferramenta muito eficaz para o desenvolvimento de aplicações web robustas e escaláveis, oferecendo diversas ferramentas e recursos, como roteamento, migrações de banco de dados, autenticação e suporte a APIs. Para simplificar a instalação e configuração de ambientes de desenvolvimento, foi utilizada a plataforma Laragon, que se mostrou particularmente útil para trabalhar com Laravel. Laragon permitiu configurar rapidamente um ambiente completo com servidor web, banco de dados e todas as dependências necessárias, agilizando o processo de desenvolvimento [5][6].

A técnica de transferência de aprendizado envolveu a reutilização de um modelo pré-treinado em um conjunto de dados extenso e genérico, ajustando-o para aprender

características específicas de um novo conjunto de dados por meio de uma etapa de ajuste fino (fine-tuning). Esse método é especialmente útil quando o conjunto de dados de destino é pequeno, permitindo aproveitar o conhecimento aprendido em grandes conjuntos de dados. O TensorFlow, um popular framework de aprendizado de máquina e deep learning desenvolvido pelo Google, foi utilizado para implementar a transferência de aprendizado através de sua API Python.

4 Resultados e Discussão

A rede neural continua sendo treinada com o objetivo de aprimorar a acurácia e a velocidade de análise das imagens submetidas. Os ajustes em curso visam melhorar tanto a precisão das detecções quanto a eficiência do processo de análise.

O desenvolvimento do portal está avançando, com ênfase na criação de uma interface simples e amigável para os usuários comuns. O objetivo é facilitar o acesso e a interação com a ferramenta sem dificuldades significativas. Além disso, estão sendo analisadas as opções de hospedagem e o hardware necessário para suportar a rede neural de forma eficaz.

A técnica de Transferência de Aprendizagem foi descartada devido aos resultados insatisfatórios obtidos. A aplicação desta técnica resultou em baixa acurácia e numerosos erros na detecção de tumores, o que comprometeu a sua viabilidade para este projeto.

5 Conclusões

A análise revelou que a utilização da técnica de Transferência de Aprendizagem não se mostrou satisfatória nem eficiente para os objetivos do projeto. Esta abordagem não conseguiu atender às expectativas, especialmente no que diz respeito à identificação de tumores em exames de outras partes do corpo, como o fígado. Para possibilitar a aplicação da rede neural em novos contextos, será necessário realizar um estudo adicional sobre a viabilidade de re-treinar o modelo existente, explorando técnicas alternativas como o Aprendizado Incremental. Esta técnica, ao contrário do re-treinamento completo do modelo desde o início, propõe atualizar o modelo com novos dados de maneira incremental, permitindo uma adaptação gradual sem a necessidade de começar do zero. No entanto, até o momento, essa técnica ainda não foi testada nesta etapa do projeto, e sua viabilidade permanece a ser investigada.

Com relação ao Portal Web, ele continua sendo desenvolvido e testado rigorosamente. Além de verificar sua ligação com a Rede Neural para testes de acesso externo, estão sendo realizados testes de integração para garantir a eficiência e a segurança da transferência de dados. Também vem sendo analisado a viabilidade de diferentes opções de hospedagem, comparando as vantagens e desvantagens de uma hospedagem local versus a hospedagem em um servidor remoto. Essa análise considera não apenas o custo-benefício financeiro, mas também o desempenho em situações de grande escala, onde um número elevado de usuários pode enviar arquivos simultaneamente. A estabilidade, a velocidade de processamento e a capacidade de escalabilidade são fatores cruciais a serem considerados para assegurar que o portal possa suportar uma alta demanda sem comprometer a qualidade do serviço.

Referências

[1] TENSORFLOW. Transferência de aprendizado e ajuste fino. TensorFlow, Https://www.tensorflow.org/tutorials/images/transfer_learning?hl=pt-br., 26 jan. 2022.

[2] TENSORFLOW. Transfer learning and fine-tuning. GitHub, Https://github.com/tensorflow/docs/blob/master/site/en/tutorials/images/transfer_learning.ipynb, 25 jul. 2024.

[3] MOZILLA. HTML básico. MDN Web Docs, Https://developer.mozilla.org/pt-BR/docs/Learn/Getting_started_with_the_web/HTML_basics, 29 jul. 2024.

[4] CURSO EM VÍDEO. Curso de HTML5 [40 horas]. Curso em Vídeo, Https://www.cursoemvideo.com/curso/html5/, 27 mar. 2024.

[5] CURSO EM VÍDEO. Curso de PHP Básico[40 horas]. Curso em Vídeo, Https://www.cursoemvideo.com/curso/php-basico/, 30 mar. 2024.

[6] LARAVEL. Laravel documentation. Laravel, Https://laravel.com/docs/11.x, 10 maio 2024.

[7] LARAGON. Laragon documentation. Laragon, Https://laravel.com/docs/11.x, 1 mar. 2019.