

Experiências com o Ensino de Arquiteturas e Programação Paralela: da Construção à Avaliação

Guilherme Galante
Ciência da Computação
Universidade Estadual do Oeste do Paraná - Unioeste
Cascavel, Paraná, Brasil
guilherme.galante@unioeste.br

Resumo—Este trabalho tem como propósito compartilhar as experiências acumuladas ao longo de dois anos na ministração de uma disciplina de programação no âmbito do curso de Ciência da Computação da Unioeste, campus Cascavel. Ao longo deste artigo, delinea-se a jornada desde a formulação curricular da disciplina até a avaliação por parte dos estudantes.

Index Terms—Programação Paralela, Arquiteturas Paralelas, OpenMP, MPI, OpenACC.

I. INTRODUÇÃO

No cenário tecnológico atual, as demandas por poder computacional crescem continuamente e a busca por soluções eficientes e escaláveis se tornou um desafio cada vez mais importante. Diversas áreas do conhecimento hoje se pautam pela resolução de problemas computacionais envolvendo grandes quantidades de dados e/ou demandando processamento de alto desempenho [1]. O uso de arquiteturas paralelas emergiu como uma resposta promissora para atender a essas necessidades, permitindo a resolução de problemas cada vez maiores e mais complexos de modo cada vez mais rápido. Tanto que hoje em dia, todos os computadores apresentam pelo menos um recurso paralelo, que pode incluir instruções vetoriais, *multithreading*, múltiplos núcleos, múltiplos processadores e aceleradores [2].

Considerando esse contexto, é necessário fornecer uma formação sólida sobre o assunto para que o aluno de Ciência da Computação (e cursos relacionados) compreenda de forma efetiva o funcionamento das arquiteturas paralelas, bem como, esteja apto para desenvolver código paralelo de modo a aproveitar plenamente as capacidades dos hardwares atuais [3].

Reconhecendo essa necessidade, as atuais diretrizes curriculares de ciência da computação já preveem a obrigatoriedade da inclusão da área de programação paralela. Iniciativas internacionais como a Força-Tarefa ACM/IEEE CS sobre Currículos de Computação (CS2023)¹ recomenda a inclusão de assuntos relacionados à computação paralela no currículo de graduação. A *Accreditation Board for Engineering and Technology* (ABET)² exige que todos os programas de Ciência da Computação acreditados abordem tópicos relacionados à área. O *NSF/IEEE-TCPP Curriculum Initiative on Parallel and Distributed Computing*³ propõe um currículo básico para

alunos de graduação de Computação, com a premissa de que cada aluno de graduação deve atingir um nível de habilidade específico em relação a tópicos relacionados à computação paralela.

No Brasil, as Diretrizes Curriculares Nacionais para os cursos de graduação na área da Computação⁴, preveem para cursos de Engenharia da Computação habilidades e competências para “*analisar e avaliar arquiteturas de computadores, incluindo plataformas paralelas e distribuídas, como também desenvolver e otimizar software para elas*”. Na opinião deste autor, por falha da resolução, tal requisito não é exigido para cursos de Ciência da Computação. No entanto, nos Referenciais de Formação para os Cursos de Graduação em Computação da Sociedade Brasileira da Computação (2017)⁵, a formação sobre sistemas paralelos e distribuídos são desejados na formação do aluno de Ciência da Computação.

Infelizmente, no Plano Político Pedagógico (PPP) do curso de Ciência da Computação da Unioeste, campus Cascavel, não foi incorporada disciplina regular que contempla especificamente conceitos de programação paralela. Alguns assuntos são cobertos pontualmente em disciplinas como Sistemas Operacionais e Arquitetura de Computadores. No entanto, o PPP prevê a oferta de disciplinas Optativas, as quais os acadêmicos devem cursar obrigatoriamente pelo menos duas delas ao longo do curso. Assim, nos anos letivos de 2021 e 2022 ofertou-se a disciplina *Programação Paralela* com o objetivo de complementar a formação dos alunos, contemplando conceitos de arquiteturas paralelas e programação paralela. Assim, o objetivo deste trabalho é apresentar as experiências vivenciadas nos dois anos de oferta desta disciplina, desde a sua concepção até a avaliação realizada pelos estudantes.

O restante deste artigo é organizado como segue. A Seção II apresenta alguns trabalhos relacionados. A Seção III apresenta a concepção do plano de ensino da disciplina. A Seção IV descreve a infraestrutura utilizada. O desenvolvimento da disciplina é apresentado na Seção V. Na Seção VI apresentam-se os pareceres dos alunos sobre a disciplina. Por fim, a Seção VII conclui o relato.

⁴Resolução CNE/CES 5/2016. Diário Oficial da União, Brasília, 17 de novembro de 2016, Seção 1, págs. 22-24.

⁵<https://www.sbc.org.br/documentos-da-sbc/send/127-educacao/1155-referenciais-de-formacao-para-cursos-de-graduacao-em-computacao-outubro-2017>

¹<https://csed.acm.org/>

²<https://www.abet.org/>

³<https://tcpp.cs.gsu.edu/curriculum/>

II. TRABALHOS RELACIONADOS

Considerando a importância do assunto, diversos trabalhos na literatura abordam questões relacionadas ao ensino de arquiteturas e computação paralela nos currículos de Computação.

A discussão sobre a inclusão do ensino de computação paralela remonta aos anos 90. Naquela época, Johnson et al. [4] anteciparam que o paralelismo representaria o futuro da computação e, portanto, deveria ocupar um lugar central nos programas de estudo de ciência da computação. Com essa visão em mente, eles propuseram o desenvolvimento de um curso introdutório abrangente em computação paralela para estudantes iniciantes, identificando conteúdos-chave e delineando diretrizes iniciais, bem como possíveis metodologias a serem adotadas.

Vinte e cinco anos depois, Vasconcelos et al. [5] apresentam um estudo onde verificam se os calouros estão preparados para compreender, aprender e reter alguns conceitos relacionados à computação paralela. Os resultados apontam que os alunos gostaram da abordagem de ensino proposta, tornando-se conscientes das vantagens inerentes à utilização da programação paralela como meio de melhorar o desempenho de um programa.

Grossman et al. [6] detalham a estrutura, os princípios e as experiências da disciplina de Fundamentos de Computação Paralela (COMP 322), adquiridos em 6 anos de ensino para alunos do segundo ano do curso na Rice University. De modo similar, Saule [7] descreve suas experiências no ensino de Computação Paralela e Distribuída para alunos de graduação. O autor descreve as características do curso e aponta algumas dificuldades como a falta de materiais unificados que contemplem todos os conteúdos previstos.

Chitra et al. [8] descrevem a implementação de uma disciplina de Computação Paralela para alunos de pós-graduação do Departamento de Engenharia de Ciência da Computação do Thiagarajar College of Engineering (Índia). Foi adotada uma abordagem de aprendizado baseada em atividades para ministrar o curso, e vários paradigmas e tecnologias de programação paralela, como OpenMP, MPI e CUDA foram contemplados. A avaliação da implementação do curso revela que é essencial introduzir a computação paralela já no currículo de graduação.

Martins [9] apresenta em sua dissertação de mestrado uma avaliação do uso de desafios de programação (como aqueles presentes em maratonas de programação) no aprendizado de programação paralela, considerando aspectos técnicos e motivacionais. O autor conclui que o uso de desafios de programação afeta positivamente o aprendizado de programação paralela, estimula o desenvolvimento de soluções criativas e promove um ambiente saudável e competição entre os alunos.

Ainda no contexto de métodos e abordagens de ensino, Younis et al. [10] investigam como o uso de Aprendizagem Baseada em Projetos pode ajudar no desenvolvimento da capacidade de programação paralela dos alunos. Eles mostram que projetos práticos, aprendizado em equipe e trabalho em

equipe são maneiras benéficas de abordar conteúdos relacionados à programação paralela. Mais recentemente, Newhall et al. [11] apresentam o projeto curricular, os métodos de ensino (baseados em metodologias ativas), e os objetivos de um curso de nível introdutório de computação paralela para alunos de séries iniciais.

Um interessante estudo de como o ensino de computação paralela vem se desenvolvendo nos últimos anos é apresentado por Duraes et al. [12]. O estudo, em forma de revisão sistemática, identifica os aspectos essenciais relacionados aos métodos e técnicas para ensinar computação paralela, recursos educacionais, disciplinas ensinadas, currículo e a satisfação dos alunos em aprender computação paralela. Já no trabalho de Bachiega et al. [13] apresenta-se um levantamento dos projetos pedagógicos de computação com foco no ensino de programação paralela em trinta universidades distribuídas na América do Norte, Europa, Ásia, América do Sul, África e Oceania.

Com base nos trabalhos citados, conclui-se que o ensino de programação paralela é um tópico relevante e em constante evolução no campo da ciência da computação. Algumas conclusões principais incluem:

- 1) Importância reconhecida: Desde os anos 90, há o reconhecimento da importância da programação paralela no campo da ciência da computação;
- 2) Variedade de abordagens: Existem várias abordagens e metodologias diferentes para ensinar programação paralela. Por exemplo, alguns estudos destacam a introdução de cursos abrangentes desde o início da graduação, enquanto outros focam em estratégias de aprendizado baseado em projetos e desafios de programação;
- 3) Desafios: Alguns desafios comuns incluem a falta de materiais unificados, a busca por abordagens criativas para motivar os alunos e a importância de introduzir o conceito de programação paralela nos currículos de graduação;
- 4) Resultados positivos: Muitos dos estudos mencionados indicam resultados positivos em relação ao ensino de programação paralela. Os alunos geralmente avaliam positivamente as abordagens de ensino propostas e reconhecem a importância da computação paralela no contexto atual.
- 5) Pesquisa contínua: O campo do ensino de programação paralela está em constante evolução, com pesquisadores explorando diferentes abordagens e avaliando sua eficácia.

III. CONSTRUINDO O PLANO DE ENSINO

Um plano de ensino bem estruturado deve começar com os conceitos básicos e progredir gradualmente para tópicos mais avançados, permitindo que os alunos construam uma base sólida antes de se aventurarem em áreas mais complexas. O entendimento da área da computação paralela exige uma exploração cuidadosa das complexidades envolvidas. A transição de uma abordagem sequencial para uma paralela

exige que os alunos compreendam os conceitos fundamentais, como concorrência, sincronização, comunicação, granularidade e balanceamento de carga. Mas antes disso, eles precisam estar familiarizados com as diferentes arquiteturas paralelas, como multiprocessadores e multicomputadores, e entender como as características das arquiteturas podem impactar no projeto e implementação das aplicações. Além disso, a participação em projetos práticos pode oferecer aos alunos a oportunidade de aplicar seus conhecimentos teóricos em contextos concretos, consolidando assim sua compreensão.

Dessa forma, buscou-se construir o plano de ensino da disciplina de Programação Paralela (CSC3323) tendo esse cenário em mente. A disciplina é ofertada de modo semestral, com 68 horas/aula previstas. Pode ser cursada por qualquer aluno do curso, a partir da segunda série⁶, embora recomendasse cursá-la após as disciplinas de Organização e Arquitetura de Computadores e Sistemas Operacionais que fornecem uma base de conceitos importantes para a compreensão desta disciplina.

A. Ementa

A ementa da disciplina contempla os seguintes itens: Arquiteturas paralelas: taxonomias, computadores de memória compartilhada e distribuída; Desenvolvimento de Programas Paralelos em Arquiteturas de Memória Compartilhada e Distribuída e GPUs; Avaliação de Desempenho de Programas Paralelos.

B. Objetivos

Como objetivo da disciplina tem-se: “Apresentar uma visão geral dos principais modelos de arquiteturas paralelas; Capacitar o acadêmico com os conhecimentos sobre as principais questões envolvidas no desenvolvimento de programas paralelos; Introduzir os principais modelos de programação paralela para arquiteturas de memória compartilhada, memória distribuída e GPUs.”.

C. Conteúdo programático

O conteúdo programático da disciplina foi dividido em cinco módulos, descritos a seguir. Tal conteúdo foi pensado de modo que os conhecimentos sejam apresentados de modo gradual, sendo que cada módulo serve como base para os seguintes.

Módulo I - Introdução à Computação Paralela e de Alto Desempenho - Este é o módulo onde se apresenta a disciplina, mostrando qual a motivação e a necessidade do uso da computação paralela para a resolução de problemas de diversas áreas do conhecimento. Nesta parte também introduz-se diversos conceitos básicos da área, tais como: processo x *core* x *thread*, concorrência x paralelismo, paralelismo implícito x explícito, entre outros. Ao fim do módulo, apresenta-se uma visão geral do que será apresentado no restante da disciplina. **Módulo II** - Introdução às Arquiteturas Paralelas - Conhecer os diferentes tipos de arquiteturas paralelas é fundamental

no desenvolvimento de habilidades em computação de alto desempenho. Este módulo contempla quatro partes:

- 1) Classificação das arquiteturas paralelas;
- 2) Arquiteturas de memória compartilhada;
- 3) Arquiteturas de memória distribuída; e
- 4) Arquiteturas Vetoriais e GPUs.

Na Parte 1, apresenta-se uma visão geral dos níveis de paralelismo que podem ser explorado em um computador, abordando desde o nível de paralelismo de instruções em um pipeline, até arquiteturas distribuídas, como *grids* e *cloud*. Apresenta-se também como cada um dos tipos de arquitetura é classificada de acordo com a nomenclatura de Flynn.

As partes 2 a 4 abordam detalhadamente arquiteturas de memória compartilhada (processadores *multicore*, SMP e NUMA), memória distribuída (*clusters* e *grids*), e arquiteturas vetoriais (extensões vetoriais, máquinas vetoriais clássicas, aceleradores vetoriais e GPUs).

Módulo III - Projeto de Aplicações Paralelas - Projetar corretamente aplicações paralelas envolve conhecer bem o problema e como este pode ser trabalhado para se adaptar corretamente à arquitetura que será adotada, seja um processador *multicore*, *cluster* ou GPU. Enfim, as escolhas na fase de projeto são fundamentais para se obter um bom desempenho e um uso eficiente dos recursos de processamento.

Este módulo é subdividido em quatro partes:

- 1) Entendendo o código sequencial;
- 2) Detectando o paralelismo;
- 3) Modelos de aplicações; e
- 4) Metodologias de projeto: PCAM.

Nas duas primeiras partes do módulo, o objetivo é mostrar ao aluno como detectar as possibilidades de extração de paralelismo a partir de uma aplicação sequencial. Por exemplo, mostrar que muitas aplicações científicas e técnicas geralmente realiza a maior parte de seu trabalho em alguns trechos bastante restritos, como laços de repetição ou função específica. A detecção desses trechos, chamados de *hotspots*, pode ser realizada usando ferramentas de análise de desempenho. Na disciplina utilizou-se a ferramenta Intel VTune Profiler⁷, pela qual pode-se obter informações detalhadas (inclusive gráficas) sobre o desempenho da aplicação, permitindo identificar áreas críticas do código.

Na parte 3, foram abordados os principais modelos de aplicações, entre eles, Mestre-Trabalhador, *Data Parallel*, *Geometric SPMD*, Pipeline e Divisão e Conquista. Por fim, na Parte 4, apresenta-se a metodologia PCAM (Particionamento, Comunicação, Aglomeração e Mapeamento), proposta por Foster [14]. Embora antiga, ainda é uma metodologia válida para o projeto de aplicações paralelas. Pretende-se para novas ofertas, incluir também conceitos de padrão de projetos voltados ao paralelismo [2], [15]. O material contendo esses novos conteúdos já foram elaborados e foram ministrados como minicurso na Escola Regional de Alto Desempenho da Região Sul (ERAD-RS) de 2023 [3].

⁶A legislação da Universidade não permite a exigência de pré-requisitos para disciplinas optativas

⁷<https://www.intel.com/content/www/us/en/developer/tools/oneapi/vtune-profiler.html>

Módulo IV - Análise de Desempenho - As métricas de desempenho desempenham um papel essencial na avaliação e otimização de programas paralelos. Assim, nesse módulo são abordadas métricas como tempo de execução, *speedup*, eficiência e escalabilidade, que são fundamentais para avaliar o quão bem um programa paralelo está se beneficiando da arquitetura subjacente. Ao se utilizar corretamente tais métricas pode-se obter *insights* valiosos sobre aspectos como escalabilidade, utilização de recursos, eficiência de comunicação e balanceamento de carga. Aborda-se também as leis relacionadas à escalabilidade das aplicações: Lei de Amdahl e Lei de Gustafson.

Módulo V - Modelos de Programação Paralela - Este módulo foi desenvolvido como uma oficina prática de programação. Foi adotado um modelo de programação para cada tipo de arquitetura paralela: OpenMP para sistemas de memória compartilhada, MPI para sistema de memória distribuída, e OpenACC para exploração do paralelismo em GPUs. Optouse por OpenMP e MPI por serem os padrões *de facto* para a exploração de paralelismo, enquanto OpenACC foi selecionado por ter seu modelo de programação semelhante ao oferecido pelo OpenMP, o que reduz a curva de aprendizagem dos alunos. Conteúdos apresentados:

1) OpenMP

- a) O modelo de programação OpenMP;
- b) Diretivas, funções e variáveis de ambiente;
- c) Diretivas de trabalho (for, sections, single e master);
- d) Cláusulas de controle ao acesso à memória (privada, compartilhada);
- e) Escalonamento
- f) Sincronização (barrier, atomic, critical);
- g) Funções OpenMP.

2) MPI

- a) Introdução ao modelo de troca de mensagens;
- b) Rotinas básicas (inicialização, finalização, rank, size);
- c) Comunicação ponto-a-ponto (send e recv);
- d) Comunicação coletiva (broadcast, gathers, scatters e reduces);
- e) Funções MPI;
- f) Sincronização;

3) OpenACC

- a) O modelo de programação OpenACC;
- b) Compilação de códigos OpenACC;
- c) Principais diretivas (parallel, loop, kernels);
- d) Gerenciamento de dados na memória.

D. Avaliações

A avaliação desempenha um papel importante, servindo como uma ferramenta fundamental para medir o progresso e o alcance dos objetivos da disciplina. Além de avaliar o conhecimento adquirido pelos alunos, ela também proporciona aos educadores um feedback valioso sobre a eficácia dos métodos de ensino e o design do currículo. A análise cuidadosa dos

resultados da avaliação permite ajustes contínuos no processo educacional.

Visando cobrir tanto a parte teórica, quanto a prática, a avaliação da disciplina foi dividida em dois momentos: uma prova (peso 30%) e um trabalho prático (peso 70%). Foi dado um valor maior ao trabalho por ser a oportunidade de demonstrar na prática os conhecimentos adquiridos. Nesse trabalho, os alunos selecionam problemas para paralelizar usando um ou mais modelos de programação paralela. Depois, realizam um conjunto de testes e fazem a avaliação de desempenho utilizando as métricas apresentadas no Módulo IV da disciplina. Em geral, tem-se adotado os problemas das Maratonas de Programação Paralela que ocorrem no WSCAD e nas ERADs. Por fim, os alunos redigem um relatório descrevendo o problema paralelizado, a metodologia empregada na paralelização, e a avaliação de desempenho e escalabilidade.

Aqui cabe uma observação sobre a dificuldade de encontrar problemas para paralelização. Embora muitos dos problemas das maratonas sejam bastante interessantes, geralmente tem uma complexidade limitada ou não disponibilizam programas para gerar entradas de dados mais desafiadoras. Uma contribuição da comunidade de alto desempenho da SBC poderia ser a criação de um repositório de problemas paralelizáveis para serem usados no ensino ou até mesmo em projetos de pesquisa onde se demandam estudos de caso mais consistentes.

E. Materiais e Bibliografia

Como apontado por Grossman et al. [6], não há um livro ou repositório atualizado que contemple todo o conteúdo previsto para uma disciplina de computação paralela. Portanto foi necessário coletar inúmeros materiais, incluindo livros, tutoriais, manuais e artigos para a montagem do material didático. A lista dos materiais empregados pode ser visualizada no Apêndice A.

IV. INFRAESTRUTURA UTILIZADA

Para a condução da disciplina, foi utilizado um laboratório com televisão ou projetor para a realização das aulas teóricas, e computadores para as atividades práticas, embora os alunos utilizassem seus notebooks pessoais. Além disso, nas duas edições da disciplina, foi utilizado o supercomputador SDumont (LNCC)⁸ como ambiente de execução das aplicações paralelas. O SDumont possui um total de 36.472 núcleos de CPU, distribuídos em 1.134 nós computacionais, compostos por CPUs multicore, e adicionalmente alguns deles com GPUs. Os nós são interligados por rede Infiniband. Os jobs são submetidos via sistema de gerenciamento Slurm.

O acesso ao supercomputador foi viabilizado via chamada pública para fins educacionais⁹, que prevê alocação de recursos computacionais, para realização de cursos de treinamento e para disciplinas de graduação e pós-graduação em instituições de ensino superior e de pesquisa. O processo de solicitação é feito via um formulário que é submetido via sistema JEMS da

⁸<https://sdumont.lncc.br>

⁹<https://sdumont.lncc.br/call.php>

SBC. Após o período de avaliação da proposta, caso seja aprovada, os documentos dos usuários (alunos e professores) são solicitados para cadastramento e disponibilização do acesso.

Por conta do uso do supercomputador, foi necessário incluir, no decorrer da disciplina, treinamento sobre o sistema de submissão de tarefas e noções de boas práticas de programação em arquiteturas paralelas.

V. DESENVOLVIMENTO DA DISCIPLINA

Esta seção tem como objetivo apresentar como a disciplina foi desenvolvida ao longo do semestre. O conteúdo foi dividido em duas partes, uma teórica, contemplando os Módulos I a IV, e uma prática, contemplando o Módulo V e elaboração do trabalho final.

Na fase teórica, a carga horária alocada para cada tópico é apresentada na Tabela I. Concluída esta etapa, procedeu-se a uma revisão abrangente, seguida da criação, em conjunto com os alunos, de um mapa mental destinado a facilitar a organização do material de estudo. O mapa gerado é exibido na Figura 1.

Tabela I
CONTEÚDOS TEÓRICOS E RESPECTIVAS CARGAS HORÁRIAS.

Conteúdo Programático	CH
Introdução à Computação Paralela	4
Conceitos básicos	2
Arquiteturas de Memória Compartilhada	8
Arquiteturas de Memória Distribuída	8
Projeto de Aplicações Paralelas	4
Total	26

No que se refere à parte prática, foram empregadas 26 horas/aula por meio de oficinas dinâmicas. Em cada sessão, iniciava-se com uma explanação dos conceitos pertinentes, seguida imediatamente por exercícios práticos de programação. Os detalhes das horas investidas em cada tópico prático encontram-se apresentados na Tabela II.

Tabela II
CONTEÚDOS PRÁTICOS E RESPECTIVAS CARGAS HORÁRIAS.

Conteúdo Programático	CH
Programação em memória compartilhada: OpenMP	8
Programação em memória distribuída: MPI	8
Acesso ao Cluster SDumont	2
Programação em GPUs: OpenACC	8
Total	26

Das horas restantes alocadas para a disciplina, um total de 4 horas foi dedicado à realização da avaliação escrita, enquanto que os acadêmicos investiram 8 horas na concepção e exposição de seus projetos práticos.

Conforme discutido previamente, os projetos executados centram-se em desafios extraídos das competições de programação paralela. Dentre os problemas solucionados, destacam-se:criptografia de senhas por ataque de força bruta, cálculo do número Pi, criação de Fractais de Mandelbrot, determinação do par de pontos mais próximos, algoritmo

k-means, produção de histogramas, implementações de algoritmos de ordenação, simulação do Jogo da Vida (Conway), aplicação do método do Gradiente Conjugado, emprego de algoritmos genéticos, entre outros desafios afins. A maior parte das soluções foi construída utilizando OpenMP, embora em alguns trabalhos tenham empregado também MPI. No entanto, é pouco comum a elaboração de soluções voltadas para GPU por parte dos estudantes, embora haja hardware disponível para tal.

Um outro aspecto notável em relação à disciplina foi a promoção ativa da participação nas Escolas Regionais de Alto Desempenho (ERAD/RS) nos anos de 2022 e 2023¹⁰. No ano de 2022, conseguiu-se mobilizar aproximadamente 60 estudantes, entre aqueles inscritos na disciplina e outros acadêmicos do curso, para o evento realizado em Curitiba-PR. Já na edição de Porto Alegre-RS, 35 alunos da Unioeste participaram do evento.

VI. AVALIAÇÃO DA DISCIPLINA: COM A PALAVRA, OS ALUNOS

Ao fim da disciplina do ano letivo de 2022, foi solicitado aos alunos que dessem um parecer sobre a disciplina. Todos os pareceres enviados estão na íntegra (sem edição), a seguir.

- “Didática excelente; apesar de ser um conteúdo que tenho dificuldade a maneira de aplicar a prova contribuiu mais para o aprendizado pessoal do que uma prova comum, tornando a matéria mais leve e mais interessante.”
- “Gostei bastante da disciplina e acho que foi um bom overview da área. Me ajudou a usar de estratégias paralelas e a pensar em como usar isso para meus projetos.”
- “Devido a turma ter poucos alunos, a comunicação professor/aluno foi extremamente enriquecedora. Avaliações e trabalhos de acordo com a proposta inicial contando com o apoio fundamental do docente e dos materiais disponibilizados.”
- “A disciplina ocorreu bem conforme o que já era esperado, com a parte teórica no início, e períodos de prática mais adiante, com dinâmicas interessantes, e aulas em laboratório bastante interativas, o que torna mais divertido aprender sobre o conteúdo. Quanto a o que pode ser melhorado, só mudar as cores dos slides, evitar vermelho vibrante, de preferência.” (Nota: o aluno neste caso é dautônico).
- “As aulas de PPar foram muito interessantes e a primeira vez que tivemos um contato mais aprofundado com arquiteturas paralelas e seus usos durante o curso, com pinceladas anteriores em Organização e Arquitetura de Computadores e em Sistemas Operacionais. Entender as variedades de arquiteturas, mesmo nas aulas mais teóricas (que foram longas por causa dos horários disponíveis, mesmo com os intervalos) foi super instrutivo pra entender um pouco mais a fundo sobre

¹⁰Devido ao atraso no Calendário Acadêmico causado pela pandemia, os anos letivos de 2021 e 2022 só foram concluídos nos anos subsequentes.

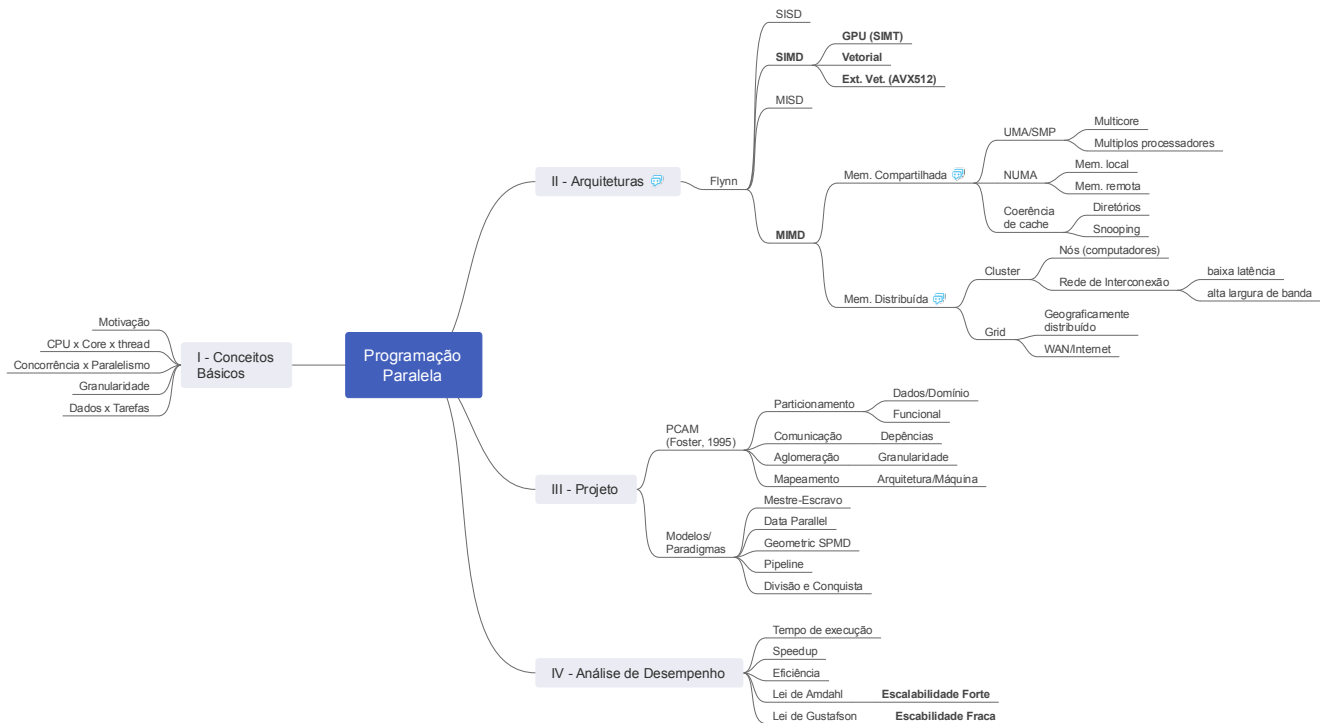


Figura 1. Mapa mental elaborado na disciplina (parte teórica).

como aplicações científicas e de alto processamento de dados podem ser escritas e executadas em diferentes ambientes e contextos de hardware. Especialmente SIMD e computação distribuída, as maiores "novidades" da disciplina pra mim, foram coisas divertidíssimas de usar e saber como funcionam. Gostei bastante dos frameworks utilizados. Acredito que os modelos de programação paralela passados foram muito pertinentes, embora acredite que talvez, para a disciplina, mais modelos poderiam ser abordados/pincelados para entender outros paradigmas/tecnologias, como Hadoop e Apache Spark."

- "Durante o desenvolvimento da disciplina houve a construção da ideia do porquê desenvolver aplicações com paralelismo, além disso a abordagem de diferentes métodos e práticas para a criação de software no modelo. Foi apresentado o supercomputador brasileiro, LNCC Santos Dumont, sendo possível durante a disciplina a utilização do mesmo para a execução das práticas propostas. Em suma, abordagem teórica e prática do paradigma da computação paralela."
- "O estudo da disciplina de Programação Paralela é fundamental para explorar o potencial máximo de desempenho dos sistemas computacionais modernos. As práticas de programação Multi-processos e Computação Distribuída expandiram o escopo previamente estudado no

Curso e permitiram o contato com tecnologias as quais normalmente não poderíamos explorar, como o Supercomputador Santos Dumont (SDumont), fornecendo uma base teórica para lidar com arquiteturas distribuídas, que estão se popularizando cada vez mais."

- "A disciplina de Programação Paralela decorreu de maneira bastante dinâmica, com a maioria de suas aulas ocorrendo em laboratório, com ótima interação entre discentes e docente. No âmbito de conteúdo teórico, foram abrangidas as estruturas que utilizam métodos de paralelismo, como MIMD e clusters, já durante o período prático, que constituiu grande parte das aulas ministradas, foram vistos os conceitos de OpenMP, OpenACC e MPI, que foram apresentados de maneira concisa e elaborada."
- "O desenvolvimento da Disciplina de Programação Paralela foi bem organizado, primeiramente, com um enfoque na parte teórica da disciplina, tratando sobre arquiteturas paralelas, classificação de Flynn, máquinas multicore, entre outros assuntos. Já com um embasamento da área, optou-se por aulas no formato apresentação da teoria e aplicação na teoria na prática (formato utilizado nas aulas de OpenMP e MPI), o que facilitou imensamente adquirir os conceitos, visto que eles ainda estavam 'frescos' na memória. Além disso, especialmente pelo

tamanho reduzido da turma, foi possível realizar as aulas em um formato mais dinâmico, com uma relação aluno-professor maior. Quanto às avaliações, foi utilizado uma avaliação escrita abordando a parte teórica, enquanto que na parte prática, realizou-se a paralelização de problemas de maratonas de programação paralela, divisão que permitiu avaliar os alunos de ambas as formas, sem sobrecarregá-los.”

Analisando os comentários, pode-se concluir que, de modo geral, a disciplina teve boa aprovação por conta dos alunos. O principal ponto positivo apontado foi o modelo mais dinâmico das aulas práticas e o uso do SDumont como ambiente de experimentação. Também teve avaliação positiva a forma como as avaliações foram realizadas. Como melhorias para a disciplina, apontou-se a inclusão de novas tecnologias, como Hadoop e Apache Spark, e o cuidado na elaboração dos slides, evitando certos padrões de cores.

VII. CONSIDERAÇÕES FINAIS

Este trabalho tem como objetivo abordar a criação da disciplina optativa de Programação Paralela, integrante do currículo do curso de Ciência da Computação da Universidade Estadual do Oeste do Paraná, campus Cascavel, além de compartilhar as experiências vivenciadas e os resultados obtidos.

No transcurso do período, a disciplina desempenhou com êxito sua função de fornecer os conhecimentos fundamentais no campo da computação paralela para as turmas de 2021 e 2022, obtendo uma recepção positiva por parte dos estudantes, conforme documentado na Seção IV. Embora desafios tenham surgido e haja aspectos a serem aprimorados, conforme ressaltado ao longo deste artigo, medidas estão sendo adotadas para aprimorar ainda mais a disciplina nas futuras edições.

Outro aspecto importante foi a ampliação do conhecimento dos alunos sobre a área, estimulando-os a se engajarem em projetos de iniciação científica e a desenvolverem trabalhos de conclusão de curso abordando temas ligados à computação paralela, bem como despertou o interesse na participação de eventos científicos.

VIII. AGRADECIMENTOS

Agradecimentos especiais aos alunos que contribuíram com os comentários e pareceres sobre a disciplina.

O autor também agradece ao Laboratório Nacional de Computação Científica (LNCC/MCTI, Brasil) por fornecer recursos HPC do supercomputador SDumont, que contribuíram para os resultados relatados neste artigo.

REFERÊNCIAS

- [1] B. Wilkinson and M. Allen, *Parallel programming - techniques and applications using networked workstations and parallel computers* (2. ed.). Pearson Education, 2005.
- [2] M. D. McCool, A. D. Robison, and J. Reinders, *Structured parallel programming patterns for efficient computation*. Waltham, MA: Elsevier/Morgan Kaufmann, 2012.
- [3] G. Galante, “Projeto de aplicações paralelas,” in *Minicursos da XXIII Escola Regional de Alto Desempenho da Região Sul*. SBC, May 2023, pp. 21–37. [Online]. Available: <https://doi.org/10.5753/sbc.11938.7.2>
- [4] D. Johnson, D. Kotz, and F. Makedon, “Teaching parallel computing to freshmen,” 1994.
- [5] L. B. A. Vasconcelos, F. A. L. Soares, P. H. M. M. Penna, M. V. Machado, L. F. W. Góes, C. A. P. S. Martins, and H. C. Freitas, “Teaching parallel programming to freshmen in an undergraduate computer science program,” in *2019 IEEE Frontiers in Education Conference (FIE)*, 2019, pp. 1–8.
- [6] M. Grossman, M. Aziz, H. Chi, A. Tibrewal, S. Imam, and V. Sarkar, “Pedagogy and tools for teaching parallel computing at the sophomore undergraduate level,” *Journal of Parallel and Distributed Computing*, vol. 105, pp. 18–30, 2017. [Online]. Available: <https://doi.org/10.1016/j.jpdc.2016.12.026>
- [7] E. Saule, “Experiences on teaching parallel and distributed computing for undergraduates,” in *2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, 2018, pp. 361–368. [Online]. Available: <https://doi.org/10.1109/IPDPSW.2018.00068>
- [8] P. Chitra and S. K. Ghafoor, “Activity based approach for teaching parallel computing: An indian experience,” in *2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, 2019, pp. 290–295. [Online]. Available: <https://doi.org/10.1109/IPDPSW.2019.00057>
- [9] G. Martins, “Avaliação do uso de desafios no aprendizado de programação paralela,” Dissertação de Mestrado, ICMC-USP, São Carlos, 2020.
- [10] A. A. Younis, R. Sunderraman, M. Metzler, and A. G. Bourgeois, “Developing parallel programming and soft skills: A project based learning approach,” *Journal of Parallel and Distributed Computing*, vol. 158, pp. 151–163, 2021. [Online]. Available: <https://doi.org/10.1016/j.jpdc.2021.07.015>
- [11] T. Newhall, K. C. Webb, V. Chaganti, and A. Danner, “Introducing parallel computing in a second cs course,” in *2022 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, 2022, pp. 321–329. [Online]. Available: <https://doi.org/10.1109/IPDPSW55747.2022.00064>
- [12] T. de Jesus Oliveira Duraes, P. Sergio Lopes de Souza, G. Martins, D. Jose Conte, N. Garcia Bachiega, and S. Mazzini Bruschi, “Research on parallel computing teaching: state of the art and future directions,” in *2020 IEEE Frontiers in Education Conference (FIE)*, 2020, pp. 1–9. [Online]. Available: <https://doi.org/10.1109/FIE44824.2020.9273914>
- [13] N. G. Bachiega, P. S. L. d. Souza, and S. d. R. S. d. Souza, “The world teaching of parallel and distributed programming,” *International Journal of Computer Architecture Education - IJCAE*, 2019. [Online]. Available: http://www2.sbc.org.br/ceacpad/ijcae/v8_n1_dec_2019/IJCAE_v8_n1_dez_2019_paper_1_vf.pdf
- [14] I. T. Foster, *Designing and building parallel programs - concepts and tools for parallel software engineering*. Addison-Wesley, 1995.
- [15] T. Mattson, B. Sanders, and B. Massingill, *Patterns for Parallel Programming*, 1st ed. Addison-Wesley Professional, 2004.

APÊNDICE A

Livros e materiais diversos utilizados na disciplina de Programação Paralela.

[Livro] **Programação Paralela e Distribuída com MPI, OpenMP e OpenACC**

Autores: Gabriel P. Silva, Calebe P. Bianchini, Evaldo B. Costa.
Editora: Casa do Código – Alura.

[Livro] **Parallel Programming: Techniques and Applications using Networked Workstations and Parallel Computers (2ª. Ed.)**

Autores: Barry Wilkinson e Michael Allen
Editora: Pearson

[Livro] **Parallel Programming for Modern High Performance Computing Systems**

Autor: Paweł Czarnul
Editora: CRC Press

[Livro] **Parallel Programming in OpenMP**

Autores: Rohit Chandra, Leonardo Dagum, Dave Kohr, Dror Maydan, Jeff McDonald, Ramesh Menon
Editora: Morgan Kaufman

[Livro] **Parallel Programming in C with MPI and OpenMP**

Autor: Michael J. Quinn
Editora: McGraw Hill

[Livro] **Parallel Computing Architectures and APIs: IoT Big Data Stream Processing**

Autor: Vivek Kale
Editora: CRC Press

[Livro] **Introduction to Parallel Computing (2ª. Ed.)**

Autores: Peter Pacheco, Matthew Malensek
Editora: Morgan Kaufman

[Livro] **Designing and Building Parallel Programs**

Autor: Ian Foster
URL: <http://www-unix.mcs.anl.gov/dbpp/text/book.html>

[Minicurso] **Projeto de Aplicações Paralelas**

Autor: Guilherme Galante
<https://sol.sbc.org.br/livros/index.php/sbc/catalog/view/119/527/804-1>

[Minicurso] **Projetando e Construindo Programas Paralelos**

Autores: Lucas Mello Schnorr, Lucas Leandro Nesi
<https://www.setrem.com.br/erad2019/data/pdf/minicursos/mc02.pdf>

[Minicurso] **Introdução à Programação com OpenACC**

Autores: Evaldo B. Costa, Gabriel P. Silva

[Apostila] **Introdução ao MPI**

URL: https://www.cenapad.unicamp.br/treinamentos/apostilas/apostila_MPI.pdf

[Slides] **Conservative Interconnect of Large-scale HPC systems**

URL: <http://research.nii.ac.jp/graphgolf/2019/candar19/graphgolf2019-nakashima.pdf>

[Slides] **OpenACC - Online Course**

URL: <https://www.openacc.org/events/openacc-online-course-2018>

[Software] **Intel Vtune Profiler**

URL: <https://www.intel.com/content/www/us/en/developer/tools/oneapi/vtune-profiler.html>

[Software] **NVIDIA HPC SDK (Compilador para OpenACC)**

<https://developer.nvidia.com/hpc-sdk>

[Misc] **Maratona de Programação Paralela - WSCAD**

URL: <http://lspd.mackenzie.br/marathon/old.html>

[Misc] **Maratona de Programação Paralela - ERAD/RS**

<https://github.com/mpp-eradrs/>

[Misc] **Supercomputador SDumont – Descrição e manuais**

URL: <https://sdumont.lncc.br/>

[Misc] **SX-Aurora TSUBASA Generation 2 (Brochure)**

https://www.nec.com/en/global/solutions/hpc/sx/docs/SX_Aurora_TSUBASA_brochure_2020_oct.pdf

[Misc] **Intel® Intrinsic Guide**

URL: <https://www.intel.com/content/www/us/en/docs/intrinsic-guide/index.html#>