



UNIVERSIDADE ESTADUAL DO OESTE DO PARANÁ

Curso de Bacharelado em Ciência da Computação

UNIOESTE - Campus de Cascavel

## MODELO PARA ELABORAÇÃO DE PROJETOS DE TRABALHO DE CONCLUSÃO DE CURSO

### 1. IDENTIFICAÇÃO DA DISCIPLINA

- 1.1. Colegiado: Ciência da Computação
- 1.2. Nome da disciplina: Trabalho de Conclusão de Curso
- 1.3. Ano de realização: 2025

### 2. IDENTIFICAÇÃO DO ALUNO

- 2.1. Nome: Caio Hideki Gomes Shimohiro
- 2.2. Nº de matrícula: 156242
- 2.3. Professor orientador: Guilherme Galante

### 3. PROJETO

- 3.1. Título: Eficiência e Escalabilidade em Algoritmos Paralelos Presentes em Benchmarks Clássicos

#### 3.2. Introdução:

A computação soluciona diversos problemas presentes em múltiplas áreas do conhecimento. Dentro desses problemas, há aqueles que trabalham com enorme volume de dados, grande complexidade, exigindo grande capacidade computacional para serem resolvidos. Tendo isso em vista, sempre houve a preocupação em aprimorar e desenvolver tecnologias, sobretudo com melhoramentos nos componentes computacionais. Uma dessas tecnologias desenvolvidas é a de computação paralela.

O paralelismo se trata de dividir os problemas para serem executados simultaneamente em múltiplos núcleos de processamento [1, 2, 7]. Assim, além de mais unidades trabalhando para os cálculos também é possível, em arquiteturas que o permitem, se aproveitar de uma maior capacidade de memória de alto nível, aumentando a velocidade do acesso aos dados.

Há, porém, problemas envolvendo a computação paralela que começam com alguns conceitos primários dessa área. O primeiro conceito é o de speedup, definido pela razão 2 entre o tempo de execução do algoritmo em apenas um processador para o tempo de execução paralelo utilizando N processadores para execução. A partir desse conceito, é derivada a eficiência, que é a razão entre o speedup da aplicação e N. Esse conceito dita portanto o custo benefício da versão paralela do algoritmo com N processadores. A tendência para algoritmos paralelos, de acordo com a chamada Lei de Amdahl, é de que o speedup tenda a diminuir conforme o número de processadores aumenta por conta do tempo de execução não fracionável, a parte serial do código [3], além do fator de overhead, que são os comandos necessários para administrar a parte paralelizável do programa. Como o problema diminui conforme N aumenta, com as tarefas sendo divididas e distribuídas entre os múltiplos

processadores, surge o conceito de granularidade, que define com que frequência o algoritmo realiza essa divisão, criando e administrando novas threads. Algoritmos com granularidade alta tendem a depender de maior administração e comunicação, refletindo e mais tempo gasto com overhead [9].

Posteriormente, com a Lei de Gustafson, o impacto da parte serial do código de acordo com Ahmdal foi contestado seguindo a ideia de que os custos com a execução da porção serial do programa não aumenta com base no tamanho do problema. Por conta disso, foi concluído que para se aproveitar mais dos recursos disponíveis (número de processadores), deve ser aumentada a dimensão do problema [4].

Dentro do processo de identificar até que ponto vale a pena aumentar o número de recursos alocados a uma solução, há o conceito da isoeficiência, que é a proporção em que o tamanho do problema deve ser aumentado com relação ao número de processadores trabalhando para que a eficiência do algoritmo seja constante. Essa proporção se torna importante para medir a escalabilidade de um sistema, já que uma isoeficiência alta mostra que é preciso um aumento muito maior na dimensão do problema para justificar mais recursos. Do contrário, ela representa que uma expansão proporcionalmente menor dos recursos pode resultar em um aumento maior de performance [5, 6].

É proposto nesse trabalho que sejam utilizadas as métricas de eficiência e isoeficiência para compreender como os recursos podem ser mais otimamente utilizados e alocados em diferentes casos de uso, tal qual analisar como diferentes atributos dos algoritmos afetam essas métricas. Para isso, serão utilizados algoritmos já conhecidos utilizados para benchmarking em ambientes paralelos.

É possível analisar o aproveitamento de hardware considerando a escalabilidade da aplicação que será executada, visto que pode não haver ganhos significativos apesar de melhorias consideráveis no equipamento. Assim sendo, pode-se obter uma visão de qual nível de hardware seria ideal do ponto de vista do custo-benefício.

### 3.3. Objetivos:

#### - Geral.

Encontrar, por meio do uso das métricas de isoeficiência e eficiência, o grau de escalabilidade de determinados algoritmos de benchmarking paralelos e estudar suas características afim de compreender como elas se relacionam aos resultados empíricos coletados.

#### - Específicos.

- Estudar e/ou implementar diferentes algoritmos de benchmarking paralelos comuns utilizando as APIs OpenMP em linguagem C;
- Testar os algoritmos em diversas configurações de número de processadores e dimensões de problema;
- Coletar os dados cronológicos das execuções e calcular os parâmetros de eficiência e isoeficiência;
- A partir dos parâmetros, desenvolver uma fórmula que será utilizada para determinar o grau de escalabilidade tendo
- Outros dados também coletados, como número de operações, podem ser utilizados para realizar um estudo comparativo entre os algoritmos;

### 3.4. Metodologia:

Inicialmente a proposta da pesquisa é utilizar material já existente na área de estudo de benchmarkings de algoritmos paralelos para seleção e possivelmente coleta de algoritmos já prontos.

Uma vez que os algoritmos selecionados já estejam implementados, eles serão testados em diversas configurações distintas, variando tanto o número de processadores quanto a dimensão do problema.

Assim, serão coletadas principalmente métricas de tempo de execução. Porém, também é possível que sejam coletadas outras métricas para tentar compreender outros aspectos dos algoritmos afim de realizar um estudo comparativo entre eles. Assim que forem extraídos os parâmetros de eficiência e, posteriormente, isoeficiência, estes serão inseridos em uma fórmula que retornará o grau de escalabilidade do algoritmo.

A fórmula em questão também deverá ser desenvolvida por meio da análise das métricas brutas dos algoritmos e possivelmente estudos externos a eles como comparação.

3.5. Cronograma de Atividades (Nesse item, deve ser confeccionado o quadro demonstrativo para o desenvolvimento, discriminado, do trabalho de TCC ao longo do ano corrente):

ETAPAS	2025						
	MAI	JUN	JUL	AGO	SET	OUT	NOV
Leitura de material de apoio	X	X					
Implementação dos algoritmos		X	X				
Testes			X	X			
Análise de resultados				X			
Escrita da prévia				X			
Entrega parcial				X			
Revisão de feedback					X		
Ajustes no trabalho					X	X	X
Entrega final							

3.6. Referências Bibliográficas (conforme modelo de referências disponível no modelo do TCC):

1. HELLER, Don. **A Survey of Parallel Algorithms in Numerical Linear Algebra**. Siam Review, [S.L.], v. 20, n. 4, p. 740-777, out. 1978. Society for Industrial & Applied Mathematics (SIAM).
2. IL'IN, V. P.. **Problems of Parallel Solution of Large Systems of Linear Algebraic Equations**. Journal Of Mathematical Sciences, [S.L.], v. 216, n. 6, p. 795-804, 24 jun. 2016. Springer Science and Business Media LLC.
3. AMDAHL, Gene M.. **Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities, Reprinted from the AFIPS Conference Proceedings**, Vol. 30 (Atlantic City, N.J., Apr. 18–20), AFIPS Press, Reston, Va., 1967, pp. 483–485, when Dr. Amdahl was at International Business Machines Corporation, Sunnyvale, California. Ieee Solid-State Circuits Society Newsletter, [S.L.], v. 12, n. 3, p. 19-20, 2007. Institute of Electrical and Electronics Engineers (IEEE).
4. GUSTAFSON, John L.. **Reevaluating Amdahl's law**. Communications Of The Acm, [S.L.], v. 31, n. 5, p. 532-533, maio 1988. Association for Computing Machinery (ACM).
5. GRAMA, A.y.; GUPTA, A.; KUMAR, V.. **Isoefficiency: measuring the scalability of parallel algorithms and architectures**. Ieee Parallel & Distributed Technology: Systems &

Applications, [S.L.], v. 1, n. 3, p. 12-21, ago. 1993. Institute of Electrical and Electronics Engineers (IEEE). 4

6. SHUDLER, Sergei; CALOTOIU, Alexandru; HOEFLER, Torsten; WOLF, Felix. **Isoefficiency in Practice**. Acm Sigplan Notices, [S.L.], v. 52, n. 8, p. 131-143, 26 jan. 2017. Association for Computing Machinery (ACM).

7. BRENT, Richard P.. **Parallel algorithms in linear algebra**. Arxiv, Canberra, p. 54-72, 30 abr. 2010. ArXiv.

8. PERLIN, N.; ZYSMAN, J. P.; KIRTMAN, B. P. **Practical scalability assesment for parallel scientific numerical applications**. arXiv (Cornell University), 1 jan. 2016.

9. MOREIRA, José E.; SCHOUTEN, Dale; POLYCHRONOPOULOS, Constantine. **The performance impact of granularity control and functional parallelism**. Lecture Notes In Computer Science, [S.L.], p. 581-597, 1996. Springer Berlin Heidelberg.

#### **4. COMPOSIÇÃO DA BANCA\***

Membro 1: Edmar André Bellorini – IFPR (Cascavel)

Membro 2: Gabriela Stein

\* para os membros constituintes da banca que não pertencem ao colegiado de Ciência da Computação, acrescentar a procedência.

#### **5. ASSINATURAS**

---

Assinatura do Aluno

---

Assinatura do Professor Orientador

#### **6. PARECERES/SUGESTÕES DA BANCA:**

(Composta por 3 professores do Curso, sendo 1 deles o próprio Professor-Orientador)