



UNIVERSIDADE FEDERAL DE OURO PRETO
PRÓ-REITORIA DE GRADUAÇÃO
PLANO DE ENSINO



| | | |
|--|--|--|
| Nome do Componente Curricular em português: Programação Paralela | | Código: BCC 447 |
| Nome do Componente Curricular em inglês: Parallel Programming | | |
| Nome e sigla do departamento: Departamento de Computação – DECOM | | Unidade acadêmica: ICEB |
| Nome do docente: Joubert de Castro Lima | | |
| Carga horária semestral 60 horas | Carga horária semanal teórica 04 horas/aula | Carga horária semanal prática 04 horas/aula |
| Data de aprovação na assembleia departamental: 20/08/2021 | | |
| Ementa: Teoria do Paralelismo; Arquiteturas paralelas; Primitivas básicas de programação paralela: controle de tarefas, comunicação e sincronização; Conceitos básicos de avaliação de desempenho e complexidade de programas paralelos; Paralelização automática; Vetorização; Algoritmos clássicos de programação paralela; Medidas de desempenho das aplicações paralelas; exploração do paralelismo em programas: implícita e explícita; Paradigmas de programação paralela: trocas de mensagens e memória compartilhada; Escrita de programas paralelos. | | |
| Conteúdo programático: <ol style="list-style-type: none">1 Apresentação do curso: programa, objetivos, bibliografia;2 Divulgação dos trabalhos (http://hpclab.net.br, link ensino);3 Introdução a programação paralela4 Razões que influenciaram o desenvolvimento da computação paralela5 Divisão das áreas da computação paralela6 Áreas de abrangência do processamento paralelo Primitivas Básicas de Programação Paralela: controle de tarefas, comunicação e7 sincronização8 Modelos de arquiteturas paralelas9 Divisão em relação ao fluxo de dados | | |

- 10 Divisão quanto ao fluxo de instruções
- 11 Divisão em relação ao mecanismo de controle
- 12 Máquinas paralelas comerciais
- 13 Classificação do desempenho das máquinas paralelas
- 14 Redes de interconexão de processadores
- 15 Medidas de desempenho das aplicações paralelas
- 16 Obtenção de resultados de aplicações paralelas
- 17 Custo computacional
- 18 Speedup
- 19 Eficiência
- 20 Escalabilidade
- 21 Exploração do paralelismo em programas
- 22 Modelos de algoritmos paralelos
- 23 Exploração implícita
- 24 Exploração explícita
- 25 Paradigmas de programação paralela
- 26 Programação através de trocas de mensagens
- 27 Programação através de memória compartilhada
- 28 Escrita de programas paralelos

Objetivos:

Ensinar as diversas técnicas de programação paralela. Ensinar modelos de algoritmos paralelos. Resolver problemas oriundos de competições nacionais ou internacionais de programação paralela.

Metodologia:

Aulas expositivas usando Google Meet.

Aulas práticas usando Google Meet e plataformas para computação em nuvem, tais como Google Cloud Platform ou Amazon AWS.

Há um ou dois revisionais antes da entrega de cada Trabalho prático/teórico. Os revisionais são atividades assíncronas em que o aluno deve usar do conhecimento obtido nas atividades síncronas para construir o Trabalho a ser entregue. Há componentes nos revisionais, tais como listas de exercícios e os enunciados dos Trabalhos, como norteadores dos estudos. Cada revisional considera uma parte do conteúdo programático.

Trabalhos práticos/teóricos com entregas na forma de seminários e feitos em plataformas para computação em nuvem, tais como Google Cloud Platform ou Amazon AWS. Os alunos possuem acesso gratuito a tais plataformas. Todo o conteúdo dos trabalhos está em: https://docs.google.com/document/d/10offmDS8FaIQFxs1QPGXSgkMxW_RdNsUZyw2YInvIeU/edit?usp=sharing. As apresentações ocorrerão em grupos de até 3 alunos e via Google Meet

(aberto aos demais alunos). A IDE de programação e o Trabalho funcionando corretamente são imprescindíveis.

O exame especial será feito de forma oral, individual e usando Google Meet. Todo o conteúdo da disciplina fará parte das perguntas ao aluno.

Atividades avaliativas:

3 entregas de trabalhos práticos/teóricos (100% da nota). Haverá avaliação da teoria e dos aspectos de engenharia, ou seja, aspectos relacionados ao como construir as 3 entregas propostas.

Trabalho 1 – 33% da nota (3,3 pontos). Nesta tarefa, o grupo apresenta aspectos práticos/teóricos do Trabalho 1. Como são vários grupos numa turma, o formato adotado é de seminário entre os alunos, onde todos apresentam e debatem.

Trabalho 2 – 33% da nota (3,3 pontos). Nesta tarefa, o grupo apresenta aspectos práticos/teóricos do Trabalho 2. Como são vários grupos numa turma, o formato adotado é de seminário entre os alunos, onde todos apresentam e debatem.

Trabalho 3 – 34% da nota (3,4 pontos). Nesta tarefa, o grupo apresenta aspectos práticos/teóricos do Trabalho 3. Como são vários grupos numa turma, o formato adotado é de seminário entre os alunos, onde todos apresentam e debatem.

1 exame como substituição de 100% da nota obtida no semestre, conforme norma da UFOP.

Cronograma:

Cada aula representa duas aulas consecutivas e o “número da atividade” representa o conteúdo discriminado no item “Conteúdo Programático”

| Data | Atividade |
|------------|-------------------------------|
| 20/09/2021 | 1;2;3;4;5;6 (síncrona) |
| 22/09/2021 | 7;8; (síncrona) |
| 27/09/2021 | 9;10;11 (síncrona) |
| 29/09/2021 | 12;13;14 (síncrona) |
| 04/10/2021 | 15;16;17; 18;19;20 (síncrona) |
| 06/10/2021 | 21;22; 23;24 (síncrona) |
| 13/10/2021 | Revisional (assíncrona) |
| 18/10/2020 | Trabalho 1 (síncrona) |
| 20/10/2021 | Trabalho 1 (síncrona) |
| 25/10/2021 | 25;26;27;28 (síncrona) |

| | | | |
|--|------------|---------------------------|--|
| | 27/10/2021 | 25;26;27;28 (síncrona) | |
| | 03/11/2021 | Revisional (assíncrona) | |
| | 08/11/2021 | 25;26;27;28 (síncrona) | |
| | 10/11/2021 | 25;26;27;28 (síncrona) | |
| | 17/11/2021 | 25;26;27;28 (síncrona) | |
| | 14/07/2021 | Revisional (assíncrona) | |
| | 22/11/2021 | Trabalho 2 (síncrona) | |
| | 24/11/2021 | Trabalho 2 (síncrona) | |
| | 29/11/2021 | 25;26;27;28 (síncrona) | |
| | 01/12/2021 | 25;26;27;28 (síncrona) | |
| | 06/12/2021 | Revisional (assíncrona) | |
| | 08/12/2021 | 25;26;27;28 (síncrona) | |
| | 13/12/2021 | 25;26;27;28 (síncrona) | |
| | 15/12/2021 | 25;26;27;28 (síncrona) | |
| | 03/01/2022 | Revisional (síncrona) | |
| | 05/01/2022 | Trabalho 3 (síncrona) | |
| | 07/01/2022 | Trabalho 3 (síncrona) | |
| | 10/01/2022 | exame especial (síncrona) | |

Bibliografia básica:

GRAMA, Ananth. Introduction to parallel computing, 2020. Resumo do livro texto Introduction to Parallel Computing. Disponível em: https://computing.llnl.gov/tutorials/parallel_comp/. Acesso em: 07/12/2020.

Tutorials Point. Parallel Algorithm Tutorial, 2020. Disponível em: https://www.tutorialspoint.com/parallel_algorithm/. Acesso em: 07/12/2020.

KIRK, David B.; HWU, Wen-mei W. Programming Massively Parallel Processors: A Hands-on Approach, Third Edition. ISBN 978-0-12-811986-0. Morgan Kaufmann. 2017. Disponível em: <https://www.sciencedirect.com/book/9780128119860/programming-massively-parallel-processors>

Bibliografia complementar:

YANG, Ulrike Meier. IMA, 2020. Material do Instituto de matemática. Universidade de Minnesota. Disponível em: <https://www.ima.umn.edu/materials/2010-2011/T11.28-29.10/10287/IMA-PPtTutorial.pdf> Acesso em: 07/12/2020.

BARNEY, Blaise. Lawrence Livermore National Laboratory, 2020. POSIX Thread Programming. Disponível em: <https://computing.llnl.gov/tutorials/pthreads/>. Acesso em: 07/12/2020.

Python Organization. Python Language Documentation, 2020. Thread programming in Python. Disponível em: <https://docs.python.org/3/library/threading.html> . Acesso em: 07/12/2020.

Oracle. Java Language Documentation, 2020. Java Parallel Processing tutorial. Disponível em: <https://docs.oracle.com/javase/tutorial/collections/streams/index.html> . Acesso em: 07/12/2020.

KLEPPMANN, Martin. Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems. 1. ed. Nova York: O'Reilly & Assoc, 2015.