



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: __/__/____		
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: 1 Apresentação do curso: programa, objetivos, bibliografia; 2 Divulgação dos trabalhos (http://hpclab.net.br, link ensino); 3 Introdução a programação paralela 4 Razões que influenciaram o desenvolvimento da computação paralela 5 Divisão das áreas da computação paralela 6 Áreas de abrangência do processamento paralelo Primitivas Básicas de Programação Paralela: controle de tarefas, comunicação e 7 sincronização 8 Modelos de arquiteturas paralelas 9 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.

Trabalhos práticos 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_RdNsUZyw2YlnvIeU/edit?usp=sharing. As apresentações ocorrerão em grupo e via Google Meet (aberto aos demais alunos).

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 seminários 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.

Seminário 1 – 33% da nota (3,3 pontos)

Seminário 2 – 33% da nota (3,3 pontos)

Seminário 3 – 34% da nota (3,4 pontos)

1 exame como substituição de 100% da nota obtida no semestre (10 pontos), conforme norma da UFOP.

Cronograma:

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

Data	Atividade
24/08/2020	1;2;3;4;5;6
25/08/2020	7;8;
26/08/2020	9;10;11
27/08/2020	12;13;14
31/08/2020	15;16;17;
01/09/2020	18;19;20
02/09/2020	21;22
03/09/2020	23;24
07/09/2020	Feriado
08/09/2020	Seminário 1
09/09/2020	Seminário 1
10/09/2020	Seminário 1
14/09/2020	25;26;27;28
15/09/2020	25;26;27;28
16/09/2020	25;26;27;28
17/09/2020	25;26;27;28
21/09/2020	Seminário 2
22/09/2020	Seminário 2
23/09/2020	Seminário 2
24/09/2020	Seminário 2
28/09/2020	25;26;27;28
29/09/2020	25;26;27;28
30/09/2020	25;26;27;28
01/10/2020	25;26;27;28
05/10/2020	25;26;27;28
06/10/2020	25;26;27;28
07/10/2020	25;26;27;28
08/10/2020	25;26;27;28
12/10/2020	Feriado
13/10/2020	Seminário 3

	14/10/2020	Seminário 3	
	15/10/2020	Seminário 3	
	19/10/2020	exame especial	

Bibliografia básica:

GRAMA, Ananth. Introduction to parallel computing. Tutorial disponível gratuitamente em: https://computing.llnl.gov/tutorials/parallel_comp/

Material sobre programação paralela gratuito – Profa. Simone de Lima Martins - UFF - <http://www.ic.uff.br/~simone/labprogparal/contaulas/>

Material sobre programação paralela gratuito – Prof. Luiz Gustavo Leão Fernandes – PUC RS - https://www.inf.pucrs.br/~gustavo/disciplinas/ppd/material/Prog_Paralela.pdf

Bibliografia complementar:

Material do Instituto de matemática. Universidade de Minnesota - <https://www.ima.umn.edu/materials/2010-2011/T11.28-29.10/10287/IMA-PPTutorial.pdf>

POSIX Thread Programming - <https://computing.llnl.gov/tutorials/pthreads/>

Thread programming in Python - <https://www.datacamp.com/community/tutorials/threading-in-python> (advanced) - <https://docs.python.org/3/library/threading.html> (parallel processing in Python) - <https://www.machinelearningplus.com/python/parallel-processing-python/>

Java Parallel Processing tutorial - <https://docs.oracle.com/javase/tutorial/collections/streams/index.html>

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