

Plano de Ensino – 2025/2

Código DCC030/049

Disciplina Programação e Arquitetura de Computadores Paralelos

Horário 2a e 4a - 14:55/16:35

Professor George Luiz Medeiros Teodoro

EMENTA:

Fundamento, arquiteturas, modelos de programação e técnicas avançadas para o desenvolvimento de sistemas paralelos. São discutidos aspectos teóricos e práticos de paralelismo em diferentes níveis - de processadores multi-core a ambientes distribuídos, com ênfase na eficiência, escalabilidade e controle de concorrência. Além disso, são estudadas plataformas e linguagens modernas paralelas, além de suas aplicações em áreas como processamento de grafos, imagens e aprendizado profundo.

A - OBJETIVOS

Este curso tem como objetivo proporcionar uma compreensão abrangente dos princípios fundamentais e dos compromissos de engenharia envolvidos no projeto de sistemas modernos de computação paralela. Além disso, busca capacitar os alunos com as técnicas essenciais de programação paralela necessárias para explorar de forma eficiente as capacidades desses sistemas. Considerando que o desenvolvimento de software paralelo eficaz requer uma compreensão sólida das principais características de desempenho do hardware, o curso abordará tanto os aspectos arquiteturais quanto os de software no projeto de sistemas paralelos.

B - PROGRAMA

- Fundamentos de Paralelismo e Arquitetura.
- Programação Paralela e Modelos Computacionais
- Desempenho e Otimizações
- Controle de Concorrência e Consistência
- Aplicações, Projetos e Comunicação Científica
- Tendências em programação paralela.

C - CRONOGRAMA

Aula	Dia	Conteúdo
01	13/08	Apresentação da disciplina
02	18/08	Por que paralelismo e eficiência.

03	25/08	Um processador multicore moderno (Parte I).
04	27/08	Um processador multicore moderno (Parte II).
05	01/09	Introdução à Programação Paralela.
06	03/09	OpenMP
07	08/09	Otimizações de desempenho: distribuição de carga e escalonamento
08	10/09	Otimizações de desempenho: comunicação, localidade etc.
09	15/09	Arquitetura de GPUs e Programação em CUDA.
10	17/09	Processamento <i>data-parallel</i> .
11	22/09	Processamento <i>data-parallel</i> .
12	24/09	Distributed Data-Parallel Computing Using Spark.
13	29/09	The Message Passing Interface (MPI)
14	01/10	Consistência de cache.
15	06/10	Consistência de memória
16	08/10	Estratégias de sincronização grão-fino: Locks, Sincronização grão-fino e Lock-free prog.
17	11/10	Linguagens para Domínios Específicos.
18	13/10	Memória Transacional.
19	15/10	Memória Transacional.
20	20/10	Acompanhamento dos Projetos (~10 min).
21	22/10	Linguagens para Domínios Específicos.
22	29/10	Processadores especializados.
23	03/11	Sistemas para Processamento Paralelo de Grafos.
24	05/11	Seminários.
25	10/11	Seminários.
26	12/11	Seminários.
27	17/11	Programação para Hardware Especializado
28	19/11	Apresentação dos Projetos.
29	24/11	Apresentação dos Projetos.
30	26/11	Apresentação dos Projetos.

D - AVALIAÇÃO

A menção final será calculada com base nos exercícios de programação (EP), apresentação de seminário (S) e projeto final (PF). A nota final será calculada como:

$$NF = 0,3*EP + 0,2*S + 0,5*PF.$$

Os exercícios de programação (EP) não entregues na data combinada terão um decréscimo de 50% por dia de atraso. Instruções para compilação e execução devem acompanhar os códigos, assim como um **relatório** descrevendo as técnicas de programação empregadas e avaliação experimental adequada. A intenção dos trabalhos é exercitar os conceitos aprendidos em sala de aula. Todos os programas devem ser executados em plataforma Linux e utilizar bibliotecas padrão.

Os seminários (S) devem apresentar artigos sobre linguagens, frameworks ou técnicas de programação paralela. Artigos não apresentados nos dias agendados receberão nota zero. Além disso, a nota será baseada na apresentação, uma vez que ela não será acompanhada de um relatório. Algumas fontes recomendadas para pesquisa de trabalhos são:

- IEEE International Parallel & Distributed Processing Symposium (IPDPS)
- The International Conference for High Performance Computing, Networking, Storage, and Analysis (SC)
- IEEE Transactions on Parallel and Distributed Systems (TPDS)
- Parallel Computing (A Journal from Elsevier)

O **projeto final (PF)** consiste na implementação de uma aplicação não trivial, preferencialmente. Preferencialmente, os alunos devem procurar problemas que demandam paralelismo nas suas áreas de interesse. As notas do PF são divididas em proposta e acompanhamento (0,1) e relatório final (0,4).

- **Proposta:** deve-se entregar uma proposta de tópico/problema, descrevendo o problema a ser implementado, sua relevância, as dificuldades do ponto de vista de paralelização. Essa proposta deverá ter até 2 páginas.
- **O acompanhamento:** corresponde a apresentação em sala de aula apresentado o problema, passos já concluídos, as dificuldades e o cronograma para o restante do trabalho.
- **Relatório final e apresentação:** devem ser entregues um relatório (formato SBC) com até 8 páginas e uma apresentação em sala de aula.

E - BIBLIOGRAFIA

- An Introduction to Parallel Programming (First Edition). Peter S. Pacheco. Morgan Kaufman, 2011. [livro- texto]. Material do livro: <http://booksite.elsevier.com/9780123742605/?ISBN=9780123742605>
- Programming Massively Parallel Processors (Second Edition) : A Hands-on Approach, David B. Kirk and Wen-mei W. Hwu. Morgan Kaufman, 2012.
- The Art of Multiprocessor Programming, Revised Reprint, Maurice Herlihy and Nir Shavit. Morgan Kaufman, 2012.
- Patterns for Parallel Programming (First Edition). Timothy G. Mattson and Beverly A. Sanders. Addison- Wesley Professional, 2004
- Distributed Systems: Concepts and Design (5th Edition). George Coulouris, Jean Dollimore, Tim Kindberg. Addison-Wesley, 2011.
- MapReduce Design Patterns (First Edition): Building Effective Algorithms and Analytics for Hadoop and Other Systems. Donald Miner and Adam Shook. O'Reilly Media, 2012