

Guido Araújo

Início

MO644/MC970 Programação Paralela (1s15)

Notícias

- 26/2 Sejam Bem-vindos
- Slides cap. 1 e cap. 5 on-line
- **Slides cap. 5 atualizados**

Ementa

Neste curso nós iremos explorar técnicas de programação paralela e estudar algumas metodologias de programação para multithread e cloud: PThreads, OpenMP, MPI e CUDA. Em adição iremos introduzir paradigmas como MapReduce (usada pela Google em seu "search engine") e Memória Transacionais. O curso terá uma componente prática, cujo objetivo é fazer o aluno entender os conceitos estudados através de uma série de tarefas de programação. Ao final do curso espera-se que o aluno tenha uma boa noção das técnicas utilizadas e dos ambientes de programação comumente empregados nesta área.

Horários

- Aulas: Ter. e Qui.
- Sala: 352 (IC3.5)
- Horários: 10:00 - 12:00
- Instrutor: Prof. Guido Araújo
- Atendimento: agende via e-mail.

Monitor

- Leandro Zanotto
- Atendimento: Ter. e Qui. após as aulas.

Bibliografia

O curso será baseado nos livros abaixo. Adotaremos um livro-texto e todas as aulas, slides e exercícios serão baseados neste.

- An Introduction to Parallel Programming by Peter S. Pacheco. Morgan Kaufman. [\[livro-texto\]](#)
- Programming Massively Parallel Processors: A Hands-on Approach, David B. Kirk and Wen-mei W. Hwu. Morgan Kaufman.
- The Art of Multiprocessor Programming, Revised Reprint, Maurice Herlihy and Nir Shavit. Morgan Kaufman.

Slides

As aulas utilizarão um conjunto de slides, disponíveis abaixo nesta página. Se necessário, notas de aula adicionais, bem como artigos discutidos em classe serão disponibilizados.

Vídeos

Algumas aulas desta disciplina estão disponíveis em [vídeo](#).

Grupo

O grupo desta disciplina pode ser encontrado [aqui](#).

Avaliação

Esta disciplina utilizará uma avaliação, majoritariamente, baseada na implementação de programas. O objetivo é fazer com que o conhecimento do aluno seja solidificado através de pequenos exemplos práticos. As atividades desta disciplina serão compostas de tarefas de programação (Tis), projeto (P) e exame (E), sendo que a média final será calculada da seguinte forma:

(a) **MC970**: Para alunos de graduação, a média final será calculada como $M = 0.7 * \text{Média (Ti)} + 0.3 * P$, onde Média (Ti) é a média das notas das tarefas Ti ($i = 1 \dots 10$) e P a nota do projeto. Exame Final (F) será aplicado se $M < 5.0$. O aluno com frequência $< 75\%$ não terá direito ao Exame Final.

(b) **MO644**: para alunos de pós-graduação, a média final será calculada como $M = 0,6 * \text{Média (Ti)} + 0.2 * P + 0.2 * E$, onde Média (Ti) é a média das notas das tarefas Ti ($i = 1 \dots 10$), P a nota do projeto e E a nota do único exame. A matéria do exame cobrirá apenas o material apresentado entre dias 09/6 e 30/6 (inclusive). O conceito final será atribuído de acordo com as faixas abaixo, normalizada em relação à turma.

- A: $M \geq 8.5$
- B: $7 \leq M < 8.5$
- C: $5.0 \leq M < 7.0$
- D: $M < 5.0$
- E: se frequência $< 75\%$

(c) Alunos matriculados em ambas as disciplinas, estão optando por fazer a disciplina como pós-graduação, e serão avaliados usando o item (b) acima.

(d) As datas de entrega de todas as tarefas de do projeto estão marcadas no calendário abaixo em azul.

(e) As notas das disciplinas estarão aqui.

Lista de Exercícios

A lista abaixo contém exercícios de preparação para o Exame. Sugerimos fortemente fazer todos eles.

- Cap.1: 1.1, 1.2, 1.3, 1.4 e 1.6
- Cap.2: 2.3, 2.8, 2.10, 2.15 e 2.19
- Cap.3: 3.6, 3.8, 3.11, 3.12, 3.25a
- Cap.4: 4.8, 4.11, 4.12, 4.16 e 4.17
- Cap.5: 5.4, 5.5, 5.8, 5.10 e 5.11
- Cap.6: 6.1, 6.6, 6.10, 6.13 e 6.18

Tarefas

Ao longo do curso serão solicitadas tarefas práticas de programação. Os exercícios deverão ser entregues, através do sistema Susy, nos prazos especificado no calendário abaixo. A página com as tarefas e instruções sobre a submissão das tarefas e do projeto podem ser encontradas [aqui](#).

Cronograma

- 26/2 Visão geral
- 03/3 Cap. 5 - OpenMP
- 05/3 OpenMP - Cláusula schedule e diretivas static, dynamic, runtime
- 10/3 OpenMP - critical, atomic, lock
- 12/3 Perfilamento de código - VTune e perf. **(T1)**
- **17/3 Cap. 2 - Hardware e software paralelo**
- 19/3 Hardware e software paralelo (cont.)
- **24/3 Cap. 4 - Pthreads (T2)**
- 26/3 Pthreads - Seção crítica, mutexes e semáforos
- 25/3 Pthreads - Read-write locks
- 31/3 Pthreads - Read-write locks (cont.) **(T3)**
- **02/4 Feriado**
- **14/4 GPU**
- 16/4 CUDA - Arquitetura e Memória **(T4)**
- **21/4 Feriado**
- 23/4 CUDA - Computação Numérica em CUDA
- 28/4 CUDA - Perfilamento e Otimização **(T5)**
- 30/4 CUDA - Stencil e Simulação de Monte Carlo
- **5/5 Apresentação Lista de projetos**
- **7/5 Cap.2 - MPI (T6)**
- 12/5 Programação básica em MPI
- 14/5 Sistemas de I/O e Comunicação coletiva em MPI
- 19/5 Datatypes e Avaliação de Desempenho em MPI **(T7)**
- 21/5 Algoritmos de ordenação em MPI
- **26/5 Cap. 6 - Projeto de programa paralelos**
- 28/5 Problema N corpos **(T8)**
- 02/6 Problemas de N corpos em MPI (cont.)
- **04/6 Feriado**
- 09/6 Arquiteturas multicore e coerência **(T9)**
- 11/6 Arquiteturas multicore e coerência (cont.)
- 16/6 Modelos de sincronização
- 18/6 Modelos de sincronização (cont.) **(T10)**
- 23/6 Memórias Transacionais
- 25/6 Memórias Transacionais (cont.)
- **30/6 Exame (MO644) (P)**
- 2/7 Não haverá aula.
- 7/7 Divulgação notas
- **14/7 Exame Final (MC970)**

Anexo	Tamanho
ch1.pdf	3.52 MB
ch5.pdf	2.23 MB

