

Programação Paralela e Distribuída (11485)

(2º ciclo em Engenharia Informática, 2019/20)

Objetivos:

Estudar modelos e técnicas de programação paralela e distribuída.

Desenhar e implementar aplicações tirando partido das atuais arquiteturas multi-core (CPU e GPU).

Estudar e utilizar ambientes para computação em grelha e em cloud.

No final o aluno deverá ser capaz de analisar artigos científicos na área da Unidade Curricular.

Programa

Princípios base da execução paralela.

- Paralelismo versus concorrência.
- Paralelismo, comunicação e coordenação.
- Erros de programação que não ocorrem na programação sequencial.

Decomposição para o paralelismo

- Decomposição baseada em tarefas
- Decomposição baseada nos dados

Arquiteturas paralelas

- Processadores multi-core
- Memória distribuída versus memória partilhada
- GPU e coprocessamento.

Técnicas de programação paralela

- Comunicação por mensagens
- Paralelismo de dados
- Estratégias de particionamento (dividir para conquistar)
- Pipelining
- Computação síncrona
- Balanceamento de carga e deteção de terminação.

Computação em grelha

Computação na cloud

Prática:

- Python multithreading
- Python multiprocessing
- Message Passing Interface (mpi4py)

- GPU Computing (pycuda, pyopencl)
- Cloud Computing (Apache Spark)

Bibliografia:

- Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers (2nd Edition). B. Wilkinson, M. Alle. Prentice-Hall, 2004
- Programming Massively Parallel Processors, A Hands-on Approach. David B. Kirk and Wenmei W. Hwu. Elsevier, 2010.
- Heterogeneous Computing with OpenCL. B. Gaster, L. Howes, D. Kaeli and P. Mistry. Morgan Kaufman, 2011.
- Mastering Cloud Computing, Rajkumar Buyya, Christian Vecchiola, and Thamarai Selvi. Morgan Kaufmann, ISBN: 978-0-12-411454-8, 2013.
- The Python Language Reference:
<https://docs.python.org/3/reference/index.html>
- Message Passing Interface (MPI)
<http://mpi-forum.org/docs/mpi-3.1/mpi31-report.pdf>
- MPI for Python:
<https://mpi4py.readthedocs.io/en/stable/>
- PyOpencl (<https://mathematician.de/software/pyopencl/>)
- PyCuda (<https://developer.nvidia.com/pycuda>)
- Apache Spark: (<https://spark.apache.org/>)

Avaliação:

Frequência (14 valores): 22 de Maio (componente T e P)

Projeto (6 valores), grupos de 2 alunos;

Apresentação teórica: 15/5; Apresentação prática: 29/5

Obtém frequência à disciplina quem assistir a um mínimo de 10 aulas práticas, obtiver nota mínima de 4.0 na frequência e de 2.0 no projeto.

Está dispensado de exame quem obtiver classificação de frequência e tiver nota ≥ 9.5 na média das notas de frequência e projeto.

O projecto é válido para exame, sendo este cotado para 14 valores tal como a frequência.

Trabalhadores estudantes que não possam assistir às aulas práticas devem falar com a docente.

Horário de atendimento: 4º feira 16-18h