

7.31.- Computação de Alto Desempenho

PROGRAMA DA DISCIPLINA

1.1 Área Científica:

1.2 Tipo(Duração): Semestral

1.3 Ano/Semestre: 4ºA / 7ºS

1.4 Tempo de Trabalho (horas):

Horas Semanais (1)					Total Horas (1)		
T	TP	P	TC	OT	TH	S	A
3	3				6,0	90,0	

(1)T-Aulas Teóricas;TP-Teórico-prático;P-aulas práticas / laboratório; TC- aula de campo;OT- orientação tutorial;TH- total de horas;S- horas semestrais;A- horas anuais.

1.5 Créditos:

2.Requisitos e Precedências

3.Competências

A Lei de Moore prevê uma duplicação do número de transístores dos circuitos integrados aproximadamente a todos os 18 meses. No entanto, na última década, o aumento de desempenho sequencial tem sido modesto. O futuro aponta para que os ganhos de desempenho sejam cada vez mais provenientes da paralelização dos programas. Nesta disciplina serão abordadas as tecnologias necessárias à criação de sistemas de elevado desempenho, não só a nível de hardware, mas especialmente de software. Os alunos deverão ser capazes de programar sistemas paralelos de memória partilhada e de troca de mensagens.

É também objectivo a aquisição das seguintes 5 competências principais (segundo os descritores de Dublin):

- trabalho em grupo, resolução de problemas, raciocínio crítico, aprendizagem autónoma e aplicação prática de conhecimentos teóricos;

e das seguintes 5 competências secundárias:

- análise e síntese, criatividade, iniciativa e espírito empreendedor, preocupação com a qualidade e planear e agir.

4.Conteúdos

Aulas Teórico/Práticas

- Taxionomia de Flynn. Arquitecturas e hardware de sistemas de elevado desempenho (shared memory [UMA e NUMA], distributed memory, híbridas, etc., e forma como são implementados)
- Arquitectura Single Instruction Multiple Data em GPUs. OpenCL.
- Arquitecturas Paralelas Modernas.
- Aspectos quantitativos do desenvolvimento de aplicações de elevado desempenho (Lei de Amdal e de Gustafson-Barsis)
- Modelos de programação de sistemas de elevado desempenho (programação paramétrica, sistemas multi-threaded, passagem de mensagens, linguagens data-parallel)

- Estratégias básicas para paralelização de aplicações (decomposição trivial, funcional e de dados)
- Programação shared-memory multi-threaded: Cilk, OpenMP e Pthreads
- Bibliotecas de suporte à concorrência em Java e Python
- Standard para programação de memória distribuída: MPI
- Sistemas de ficheiros distribuídos e paralelos
- Algoritmos paralelos essenciais
- Grid & Cloud

5. Metodologia de Ensino/Aprendizagem

Aulas teóricas, recorrendo a meios audiovisuais, com exposição detalhada dos conceitos, princípios e teorias fundamentais e com a resolução de exercícios práticos elementares que concretizem o interesse prático da matéria e exemplifiquem a sua aplicação a situações reais.

Aulas práticas-laboratoriais, onde o docente deverá apoiar os alunos na realização dos trabalhos práticos das disciplinas.

6. Modo de Avaliação

A avaliação consiste num exame final (6/20), em dois mini-testes (3/20 + 3/20) e em trabalhos práticos (8/20). No exame e nos mini-testes são colocadas questões de natureza teórica, em que os alunos demonstram a compreensão das matérias cobertas na disciplina. Há também lugar a um menor número de questões práticas. Nos trabalhos práticos, os alunos terão de utilizar de forma adequada as tecnologias que aprendem na disciplina. Na parte teórica de 12 valores (6+3+3) e na parte prática de 8, os alunos terão de ter um mínimo de 47.5% para aprovação.

7. Bibliografia:

- Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers. Barry Wilkinson, Michael Allen, C. Michael Allen, Prentice Hall, March 2004.
- MPI, The Complete Reference, Vol. 1: The MPI Core. Marc Snir et. al., 2nd ed., September 1998
- (<http://www.mpi-forum.org/docs/docs.html>)
- Patterns for Parallel Programming [Hardcover] Addison-Wesley Professional; 1 edition (September 25, 2004) Timothy G. Mattson, Beverly A. Sanders, Berna L. Massingill
- From the Lawrence Livermore National Laboratory (LLNL): (<http://www.llnl.gov/computing/hpc/training/>)
 - Introduction to Parallel Computing
 - Message Passing Interface (MPI)
 - OpenMP

O Regente

O Docente
