

Campus: São José dos Campos		
Curso (s): Engenharia de Computação e Ciência da Computação		
Unidade Curricular (UC): Programação Paralela e Processamento de Alto Desempenho		
Unidade Curricular (UC): <i>Parallel Programming and High Performance Computing</i>		
Código da UC: 4663		
Docentes Responsáveis: Álvaro Luiz Fazenda		Contato (e-mail): <i>alvaro.fazenda@unifesp.br</i>
Docente (s) Colaborador/a (es/as):		Contato (e-mail): [opcional]
Ano letivo: 2022	Termo: 8	Turma (s): I
Nome do Grupo/Módulo/Eixo da UC (se houver):		Idioma predominante em que a UC será oferecida: (x) Português () English () Español () Français () Libras () Outro:
UC: () Fixa (X) Eletiva () Optativa	Oferecida como: (x) Disciplina () Módulo () Estágio () Outro:	Oferta da UC: (x) Semestral () Anual
Ambiente Virtual de Aprendizagem: () Moodle (x) Classroom () Outro: () Não se aplica		
Pré-Requisito (s) - Indicar Código e Nome (s) da (s) UC: 3580 - Programação Concorrente e Distribuída		
Carga horária total (em horas): 72h		
Carga horária teórica (em horas): 36h	Carga horária prática (em horas): 36h	Carga horária de extensão (em horas, se houver):
Se houver atividades de extensão, indicar código e nome do projeto ou programa vinculado na Pró-Reitoria de Extensão e Cultura (ProEC):		
Ementa: Revisão: arquiteturas computacionais paralelas de memória compartilhada e distribuída e métricas de desempenho. Pacotes paralelos para funções matemáticas. Afinidade de dados em memória cachê. Influência da memória cache no desempenho. Programação com OpenMP avançada. Programação com o padrão MPI-1 e MPI-2. Programação em C/C++-CUDA para GPUs (Graphics Processing Units). Introdução ao modelo PGAS (Partitioned Global Address Space). Introdução a Computação em Grade		
Conteúdo programático: <ol style="list-style-type: none"> 1. Revisão: arquiteturas computacionais paralelas de memória compartilhada e distribuída; 2. Arquiteturas <i>Multicores / Manycores</i>; métricas de desempenho: Speedup, eficiência, escalabilidade, granularidade; investigação sobre razões que impedem paralelismo eficiente: fração serial do código, custo de comunicação e desbalanceamento de carga. 3. Pacotes paralelos para funções matemáticas (<i>BLAS, LAPACK, ATLAS, MKL, BLACS, scaLAPACK</i>). 4. Afinidade de dados em memória cachê. Influência da memória cachê no desempenho. 5. Programação com <i>OpenMP</i> avançada: revisão, novas funcionalidades e aplicações. 6. Programação com o padrão <i>MPI-1</i> e <i>MPI-2</i>: Revisão de comunicações ponto-a-ponto. Comunicações coletivas. Criação de tipos derivados de dados. Criação e manipulação de Grupos e Comunicadores. Topologias de processos. Avaliações de desempenho. Padrão <i>MPI-2: One-sided-communication</i>, Entrada/saída paralela. 7. Programação em <i>C/C++ - CUDA</i> para GPUs (<i>Graphics Processing Units</i>): Arquiteturas de uma GPU, Introdução a Linguagem CUDA, Organização de <i>threads</i> em <i>CUDA</i>, Acesso a memória. 8. Introdução ao modelo <i>PGAS (Partitioned Global Address Space)</i>. 9. Introdução a Computação em Grade: 10. modelo <i>Bag of Tasks, OurGrid</i>. 		

Objetivos:

Geral: Apresentar aos alunos os fundamentos da programação concorrente para arquiteturas paralelas e/ou distribuídas.

Específicos: Ao final do curso os alunos deverão ser capazes de compreender os princípios da programação concorrente para arquiteturas paralelas e distribuídas, bem como projetar algoritmos segundo estes princípios.

Metodologia de ensino:

- Aulas expositivas;
- Atividades monitoradas em grupos de trabalho;
- Laboratório de programação;
- Atividades complementares a distância;
- Listas de exercícios.

Avaliação:

Metodologia de Avaliação:

MF = (soma das melhores N atividades práticas de programação, com descarte das duas piores notas)/N

O **Exame** consiste em uma prova prática de programação que abrange o conteúdo do semestre.

Bibliografia:

Básica:

1. BEN-ARI, M. Principles of concurrent and distributed programming. 2nd ed. Harlow: Addison-Wesley, 2006. 361 p ISBN 9780321312839.
2. GRAMA, Ananth; KUMAR, Vipin; KARYPIS, George; GUPTA, Anshul. Introduction to parallel computing. 2nd. ed. London: Person Addison Wesley, c2003. 636 p. ISBN 9780201648652.
3. PACHECO, Peter S. An introduction to parallel programming. Burlington, MA: Morgan Kaufmann, 2011. 370 p. ISBN 9780123742605.

Complementar:

1. HERLIHY, Maurice; SHAVIT, Nir. The art of multiprocessor programming. Burlington: Elsevier, c2008. 508 p. ISBN 97801237005914.
2. GHOSH, Sukumar. Distributed systems: an algorithmic approach. Boca Raton: Chapman & Hall/CRC, 2006. 402 p. ISBN 1584885645.
3. RAUBER, Thomas; RÜNGER, Gudula. Parallel programming: for multicore and cluster systems. New York: Springer, c2010. 455 p. ISBN 9783642048173.
4. KIRK, David B.; HWU, Wen-Mei W. Programando para processadores paralelos: uma abordagem prática à programação de GPU. Rio de Janeiro: Elsevier, 2011. 212 p. ISBN 9788535241884.
5. HEROUX, Michael A.; RAGHAVAN, Padma; SIMON, Horst D. (ed.). Parallel processing for scientific computing. Philadelphia: SIAM, 2006. 397 p. ISBN 9780898716191.
6. BRESHEARS, Clay. The art of concurrency: a thread monkey's guide to writing parallel applications. Sebastopol, CA: O'Reilly, 2009. 285 p. ISBN 9780596521530.