

Unidade: Instituto de Informática - UFRGS
Departamento de Informática Aplicada (INA)

Dados de identificação

Disciplina: Programação Distribuída e Paralela			
Período Letivo: 2021/2			
Professor Responsável: Cláudio Fernando Resin Geyer			
Sigla: INF01008	Créditos: 4		
Carga Horária: 60h	CH Autônoma: 10h	CH Coletiva: 50h	CH Individual: 0 h

Súmula

Conceitos básicos de sistemas distribuídos; aplicações distribuídas e paralelas; características básicas de PDP; modelo de processos e troca de mensagens; conceitos básicos de programação paralela; algoritmos paralelos e distribuídos; conceitos de Big Data; conceitos de sistemas distribuídos em larga escala (Cloud, IoT e Big Data); introdução a frameworks Big Data como Hadoop e Spark; programação de aplicações com Big Data (batch/streaming); introdução à programação móvel para IoT.

Currículos

Currículos	Etapa	Pré-Requisitos	Natureza
Engenharia de Computação	7	(INF01120) Técnicas de Construção de Programas e (INF01151) Sistemas Operacionais II	Eletiva
Bacharelado em Ciência da Computação	7	(INF01120) Técnicas de Construção de Programas e (INF01151) Sistemas Operacionais II	Eletiva
Bacharelado em Matemática - Ênfase matemática aplicada e Computacional	8	90 créditos obrigatórios	Alternativa

Objetivos

O primeiro objetivo da disciplina é a apresentação dos diferentes mecanismos para PDP encontrados em linguagens e bibliotecas (ambientes) de programação distribuída e paralela (PDP), de modo que o aluno possa avaliar as vantagens e desvantagens de cada um com relação ao desenvolvimento de programas (tipos de), incluindo aqueles em sistemas de larga escala como Cloud, IoT e Big Data. Algoritmos básicos de sistemas distribuídos são apresentados, seja para exemplificar o uso dos modelos como pela sua utilidade em PDP. Conceitos básicos de programação paralela são apresentados, seguidos de exemplos clássicos de algoritmos paralelos acompanhados de uma introdução à sua complexidade. Alguns dos ambientes de PDP mais usados, em especial Cloud, IoT e Big Data, são apresentados com ênfase sobre o modelo de sistemas largamente distribuídos.

Conteúdo Programático

Semana	Título	Conteúdo
--------	--------	----------

1	Apresentação da disciplina	Introdução a PDP a) áreas; b) tópicos e abordagem; c) casos; d) bibliografia; e) avaliação
1,2	Programação Concorrente (Revisão)	Programação Orientada a Objetos (POO): gerência de threads, monitores, opcionais: extensões de locks e semáforos, barreiras, variáveis atômicas e interrupção.
3,4	Algoritmos Distribuídos; um (1) encontro síncrono presencial para atividades previstas na metodologia.	Conceitos de AD. Algoritmos de Coleta, Snapshot Distribuído; opcionais - Terminação, Semáforo Distribuído.
5,6	Big Data; um (1) encontro síncrono presencial para atividades previstas na metodologia.	Conceitos de Big Data; Modelo MapReduce; Conceitos de streaming. Framework para Big Data streaming. Introdução à programação MapReduce. Ferramenta para gerência de recursos e jobs. Introdução às aplicações Machine Learning distribuídas.
7,8	Algoritmos Paralelos; um (1) encontro síncrono presencial para atividades previstas na metodologia.	Introdução à programação paralela; conceitos de algoritmos paralelos (AP); requisitos de um bom algoritmo; máquinas abstratas para algoritmos paralelos; máquinas DAG e PRAM; APs em PRAM: produto vetorial; soma de n elementos; multiplicação de matrizes; com análise da complexidade e variações.
8	Aula Demo; síncrona presencial	Experiência com ambientes de PD com tarefas e memória compartilhada; experiência com troca de mensagens
9,10	Algoritmos Paralelos; um (1) encontro síncrono presencial para atividades previstas na metodologia.	Introdução à programação paralela; conceitos de algoritmos paralelos (AP); requisitos de um bom algoritmo; máquinas abstratas para algoritmos paralelos; máquinas DAG e PRAM; APs em PRAM: produto vetorial; soma de n elementos; multip
11,12	Programação Paralela; um (1) encontro síncrono presencial para atividades previstas na metodologia.	Introdução à metodologia de desenvolvimento para programas paralelos; MPI: introdução; API de controle; API de comunicação ponto-a-ponto; API de comunicação coletiva; exemplos de códigos; APIs especiais. OpenMP: introdução; diretivas de paralelização e sincronização; exemplos
12	Aula Demo; síncrona presencial	Experiência em ambientes para programação distribuída corporativa; Experiência com ambientes para mobilidade, computação ubíqua, ...
13, 14	Tópicos Especiais em Computação Distribuída; um (1) encontro síncrono presencial para atividades previstas na metodologia.	Conceitos e exemplos de frameworks em: IoT, Cloud, Fog/Edge. Introdução ao Processamento Móvel. Mecanismos de sincronização e comunicação indireta.

		Plataformas de desenvolvimento e introdução à programação para dispositivos móveis (Opcional). Ambientes Corporativos como JEE, AWS, Google Cloud, Google Core IoT e outros (opcional)
15	Apresentações de Trabalhos Finais	

Metodologia

A disciplina será apresentada na forma de aulas teórico-práticas e pela execução de trabalhos e exercícios extraclasse. **As atividades ocorrerão de forma remota síncrona ou assíncrona, e presencial restrita conforme previsto no cronograma. As atividades presenciais restritas ocorrerão em alguns encontros (ver cronograma) nos dias e horários regulares da disciplina.**

Durante os encontros presenciais será dada preferência a: a) apresentação resumida dos conceitos e técnicas; b) tratamento de dúvidas dos alunos; c) exercícios de apoio; d) apresentação e demo do enunciado de atividades práticas.

Para os alunos que não puderem participar das atividades presenciais, com será disponibilizado alguns encontros síncronos com gravação para a realização das mesmas atividades, em modo reduzido.

A parte teórica será realizada através da apresentação de conceitos via vídeo aulas (formato assíncrono preferencialmente) e por leituras e exercícios recomendados. Os conceitos abordados na parte teórica serão complementados e reforçados por atividades e exercícios práticos, também em modo assíncrono. Tarefas de programação serão realizadas em linguagens de programação com compiladores e ambientes de desenvolvimento e execução paralela e/ou distribuída.

Todas as atividades serão propostas, entregues e avaliadas usando preferencialmente o Moodle da disciplina (em especial, isto será indicado pelo Moodle). Em caso de dúvidas, os alunos poderão contar com atendimento do professor, em momentos síncronos no horário regular da disciplina, a fim de evitar conflitos com outras atividades dos discentes. Todos os encontros síncronos serão gravados pelo AVA escolhido e disponibilizados no AVA e/ou no Moodle. Conforme semestres anteriores, o TEAMS será a ferramenta AVA preferencial.

Informações sobre Direitos Autorais e de Imagem:

Todos os materiais disponibilizados são exclusivamente para fins didáticos, sendo vedada a sua utilização para qualquer outra finalidade, sob as penas legais.

Todos os materiais de terceiros que venham a ser utilizados devem ser referenciados, indicando a autoria, sob pena de plágio.

A liberdade de escolha de exposição da imagem e da voz não isenta o aluno de realizar as atividades originalmente propostas ou alternativas;

Todas as gravações de atividades síncronas devem ser previamente informadas por parte dos professores.

Somente poderão ser gravadas pelos alunos as atividades síncronas propostas mediante concordância prévia dos professores e colegas, sob as penas legais.

É proibido disponibilizar, por quaisquer meios digitais ou físicos, os dados, a imagem e a voz de colegas e do professor, sem autorização específica para a finalidade pretendida.

Os materiais disponibilizados no ambiente virtual possuem licença de uso e distribuição específica, sendo vedada a distribuição do material cuja a licença não permita ou sem a autorização prévia dos professores para o material de sua autoria.

Carga Horária

Teórica: 50 horas

Prática: 10 horas

Experiências de Aprendizagem

Os discentes serão estimulados a realizar as seguintes atividades de aprendizagem:

- 1) Visualização de videoaulas (modo assíncrono) seguido de leituras recomendadas que complementem e reforcem os conceitos apresentados nas videoaulas;
- 2) Implementação de trabalhos de programação;
- 3) Execução de listas de exercícios de fixação de conteúdo recomendadas;
- 4) Realização de atividades de verificação de aprendizagem dos conteúdos vistos até a data das mesmas.

Critérios de Avaliação

A avaliação será realizada na forma de (i) exercícios teóricos, (ii) trabalhos práticos e (iii) exercícios de fixação de conceitos. **Em relação a atividades de verificação de aprendizagem, esses serão realizados na forma de 2 atividades teóricas (AT) síncronas presenciais referentes aos conteúdos desenvolvidos durante as várias etapas do semestre. Essas atividades devem ser realizadas de forma presencial, sem consulta, devido o estilo de verificação usado: provas com questões sobre conceitos exigindo respostas textuais, e questões de programação e algoritmos também com respostas textuais.** Em relação a trabalhos práticos, esses consistirão na execução de 2 atividades de programação guiada (PG) por roteiros experimentais e de um trabalho de programação (TF). **No início de cada PG haverá um (1) encontro presencial síncrono, onde o enunciado será apresentado e uma demo básica das ferramentas poderá ser realizada.** As apresentações dos PGs e dos TFs acontecerão em seminários síncronos online. Os exercícios de fixação de conceitos (EX) serão realizados periodicamente após determinado tópico do programa, de forma assíncrona remota. A média final (M) será calculada da seguinte forma: $M = 0,4 * ((AT1 + AT2) / 2) + 0,3 * (TF) + 0,2 * ((PG1 + PG2) / 2) + 0,1 * (\text{média dos EX})$, considerando notas na escala 1 - 10. A conversão da média numérica para conceito será feita da seguinte maneira:

$M \geq 9,0$: Conceito A

$9,0 > M \geq 7,5$: Conceito B

$7,5 > M \geq 6,0$: Conceito C

$M < 6,0$: ver Atividades de Recuperação

Será considerado aprovado o aluno que obtiver média final (MF) maior ou igual a 6,0.

De acordo com a Resolução do CEPE sobre o ERE, durante o período em que perdurar o ERE, fica inaplicável a atribuição de conceito FF, prevista no §2º, do Art. 44, da Resolução nº 11/2013 do CEPE.

Para os estudantes matriculados até o final do período e que deixaram de participar da Atividade de Ensino, deverá ser atribuído o registro NI (Não Informado) no campo de conceito do sistema acadêmico.

Para os casos previstos no §1º, a justificativa do registro NI deverá conter a referência ao período de excepcionalidade.

Os casos de não informação de conceito durante o ERE, deverão ser resolvidos até o fim do segundo período letivo, após o fim da situação emergencial de saúde.

Atividades de Recuperação Previstas

Os discentes que não alcançarem média final para aprovação (ou seja $M < 6,0$) poderão realizar uma atividade de recuperação teórica (RT) e/ou uma atividade de recuperação prática (RP), sobre qualquer dos conteúdos apresentados na disciplina. **A atividade de recuperação teórica será realizada de forma presencial síncrona, nos mesmos moldes das atividades de verificação de aprendizagem executadas no decorrer do semestre. A atividade de recuperação prática será realizada de forma remota síncrona com apresentação do aluno.**

A nota da atividade de recuperação teórica (RT) substituirá 50% da nota M, correspondente aos exercícios teóricos e de fixação de conceitos. A nota da atividade de recuperação prática (RP) substituirá 50% da nota M, correspondente aos trabalhos práticos.

Prazo para Divulgação dos Resultados das Avaliações

Os resultados de cada atividade serão divulgados em até quatro semanas após sua data de entrega. Os resultados da prova serão disponibilizados em até duas semanas após a avaliação, respeitando-se o prazo de 72 horas antes da recuperação.

Bibliografia

A Bibliografia Básica Essencial deve estar disponível de forma digital.

Básica Essencial
1. Andrews, G.R.. Foundations of Multithreaded, Parallel, and Distributed Programming. Addison-Wesley, ISBN 978-0201357523.
2. Jaja, J.. An Introduction to Parallel Algorithms. Addison Wesley, ISBN 978-0201548563.
3. White, Tom - Hadoop – The Definitive Guide, Ed. O’ Reilly, 4ª ed., 2015.

Básica
1. Bodoff, Stephanie. Tutorial J2EE. Rio de Janeiro: Ciência Moderna, 2005. ISBN 8573933836.
2. Lea, Douglas. Concurrent programming in java :design principles and patterns. Addison-Wesley, 2009. ISBN 9780321256171.
3. - LECHETA, Ricardo R. Google android: aprenda a criar aplicações para dispositivos móveis com o android SDK. Editora Novatec, 2010.
4. Lynch, Nancy A.. Distributed algorithms. San Francisco: Morgan Kaufmann, c1996. ISBN 1558603484.
5. Taubenfeld, Gadi. Synchronization algorithms and concurrent programming. Harlow: Pearson Prentice Hall, 2006. ISBN 9780131972599;
6. Tel, Gerard. Introduction to distributed algorithms. Cambridge: Cambridge University, 2000. ISBN 0521794838.
7. Pacheco, P.. 978-0201548563. Morgan Kaufmann, ISBN 978-1558603394.

Complementar

1. Tanmay Deshpande - Learning Apache Flink, Ed. Packt Publish, 2017.
2. Holden Karau, Andy Konwinski, Patrick Wendell, Matei Zaharia - Learning Spark: Lightning-Fast Big Data Analysis, Ed. O’ Reilly, 1ª ed., 2015.
- Diversos. - Tutoriais na Web sobre Ambientes de Programação: Java Distribuído, J2EE, MPI, OpenMP, Hadoop, Spark, Android, iOS, Google G1.

Outras Referências

Observações

Conforme descrito no item Metodologia:

- As 60 horas previstas para atividades teóricas e práticas indicadas neste Plano de Ensino incluem 30 encontros de 100 minutos de duração (2 períodos de 50 minutos por encontro, 2 encontros por semana, durante 15 semanas), num total de 3.000 minutos, e mais 10 horas (600 minutos) de atividades autônomas, realizadas sem contato direto com o professor, correspondentes a exercícios e trabalhos extraclasse.

Prof. Cláudio Geyer