

Disciplina DCE540 - Computação Paralela e Distribuída	Método de entrega Moodle da disciplina	Data de entrega 26/01/2022 às 22h00
Professor Iago Augusto de Carvalho (iago.carvalho@unifal-mg.edu.br)		

Prova 01

Cada aluno deverá submeter um único arquivo .pdf com a resolução da prova.

A prova pode ser realizada de duas maneiras:

- Com papel e caneta, sendo posteriormente escaneada e enviada
- Digitada em algum editor de texto, e.g., Word ou LaTeX

A prova deverá ser entregue no Moodle da disciplina até a data limite.

- Atrasos não serão tolerados

Exercício 1 (15 %)

Uma das características básicas de sistemas distribuídos é a transparência. Em um sistema distribuído transparente, espera-se que seus recursos, embora fisicamente distantes entre si, sejam exibidos de uma forma coesa e clara aos usuários e desenvolvedores, de tal forma que estes não precisem se preocupar com a localização espacial dos recursos que estão utilizando. Existem alguns importantes tipos de transparência que devem ser garantidos em sistemas distribuídos. Explique, com suas próprias palavras, as definições de

- a) Transparência a falhas
- b) Transparência de migração
- c) Transparência de concorrência

Exercício 2 (20%)

Existem diversos tipos de arquitetura de software em sistemas distribuídos. Apresente a definição das quatro principais delas, a saber

- a) Arquitetura baseada em camadas
- b) Arquitetura baseada em objetos
- c) Arquitetura baseada em recursos
- d) Arquitetura baseada em eventos

Exercício 3 (10%)

Quais são as definições de processos e threads? Quais são as principais diferenças entre os dois?

Exercício 4 (15%)

Em suas palavras, explique cada um dos tipos de sistema distribuídos abaixo. Além disso, dê um exemplo real de cada um destes sistemas

- a) Sistemas de computação distribuída de alta performance
- b) Sistemas de informação distribuídos
- c) Sistemas pervasivos

Exercício 5 (20%)

Apresente a definição dos seguintes esquemas de comunicação

- a) Comunicação transiente e síncrona
- b) Comunicação transiente e assíncrona
- c) Comunicação persistente e síncrona
- d) Comunicação persistente e assíncrona

Exercício 6 (20%)

Considere um sistema computacional que possui 8 processadores independentes, cada uma com uma *thread*. Calcule o *speed-up* de algoritmos com as seguintes características:

- a) Algoritmo tem uma fração paralela de 25% (decomposição de domínio)
- b) Algoritmo tem uma fração paralela de 80% (decomposição funcional), sendo que a maior das quatro sub-tarefas corresponde a 50% da fração paralela
- c) Algoritmo possui duas frações paralelas. A primeira é uma decomposição de domínio e corresponde a uma fração de 30% do algoritmo. A segunda é uma decomposição funcional e corresponde a 20% do algoritmo, sendo que as quatro sub-tarefas levam o mesmo tempo para serem executadas

Gabarito

Exercício 1

- a) Oculta quando ocorre falha e a recuperação de um recurso é feita de forma que o usuário não percebe que processo falhou
- b) Oculta que um objeto pode (ou foi) movido para outra localização física
- c) Oculta que um recurso é compartilhado por dois ou mais usuários

Exercício 2

- a) Componentes são organizados em camadas. Além disso, as chamadas de componentes são realizadas seguindo estas camadas, no geral, de uma camada anterior para uma posterior, cabendo ao componente anterior somente esperar uma resposta. Em alguns casos, pode acontecer chamadas no sentido inverso, isto é, de um componente de uma camada posterior para uma anterior. Um exemplo real que implementa esta arquitetura é o sistema de comunicação TCP/IP, onde os componentes de cada camada enviam dados e fazem requisições para as camadas posteriores a eles.
- a) Cada componente representa um objeto. Nesta arquitetura, as chamadas de componentes podem ser realizadas sem uma ordem predefinida, isto é, com origem e destino em qualquer componente. Esta arquitetura tem um forte paralelo com o paradigma de programação orientada a objetos. Diversos serviços distribuídos se encaixam nesta categoria, sendo que lojas online são claros exemplos de sistemas distribuídos desenvolvidos utilizando-se componentes como objetos.
- a) Nesta arquitetura existem diversos componentes isolados que fornecem recursos ou serviços para os usuários. Na maioria dos casos, existe uma interface de usuário que simplifica o acesso a estes recursos e serviços disponibilizados. Um exemplo desta arquitetura são sistemas web baseados em serviços REST (Representational state transfer).
- a) Este é um caso especial de arquiteturas *publisher-subscriber*. Nesta arquitetura, assim como nas arquiteturas baseadas em recursos, existem diversos componentes isolados que oferecem recursos ou serviços, sendo estes denotados como *publishers*. Outros componentes, denotados como *subscribers*, podem escolher utilizar ou não os recursos dos componentes *publishers*. Sempre que um *publisher* disponibiliza novos dados, ele notifica todos os seus *subscribers* ativos no momento e envia os dados a eles. Um exemplo prático de um sistema construído utilizando esta arquitetura são os espaços de dados compartilhados, isto é, serviços de armazenamento em nuvem onde é permitido compartilhar dados entre diversos usuários, como o Dropbox, o Google Drive ou o One Drive.

Exercício 3

Todos os arquivos executáveis chamados serão rodados como um processo. Processos são controlados pelo sistema operacional, sendo definidas suas permissões de acesso, espaço de memória e tempo de processador. Processos podem ter uma linha de execução única ou então serem divididos em múltiplas linhas de execução utilizando threads.

Uma thread é chamada de um processo leve. Threads normalmente são criadas e gerenciadas por processos e não existem *per-se*. Diferentemente de processos, uma thread não é gerenciada pelo sistema operacional, sendo que o processo pai é responsável por gerenciar o tempo de processador, as permissões e o espaço de memória de cada thread.

As principais diferenças entre processos e threads são representadas na Tabela 1.

Exercício 4

- a) São aqueles sistemas de computação distribuídos utilizados para processar um grande volume de dados ou para paralelizar e acelerar a execução de algoritmos. Estes sistemas oferecem grandes recursos de

Tabela 1: Diferenças entre processos e threads

Feature	Processo	Thread
Execução	Linha própria	Linha própria
Memória global (heap e data segment)	Própria	Compartilhada
Memória local (stack, registers, PCounter)	Sim	Geralmente
Consumo de memória	Normal	Ligeiramente menor
Manipuladores de recursos externos	Proprietário	Empresta do processo
Tempo de criação	Relativamente longo	Relativamente curto
Tempo de troca de contexto	Relativamente longo	Relativamente curto
Instâncias	Múltiplas	Múltiplas
Associação	Um programa (executável)	Um processo
Paralelismo	Limitado	Sim
Comunicação entre seus pares	Só com um mecanismo de IPC	Sim, dentro do processo
Eficiência de comunicação	Não	Sim
Comunicação direta com o OS	Sim	Não
Controle de exceções	Próprio	Próprio
Confiabilidade e segurança	Sim	Depende do código

hardware (poder de processamento e espaço de armazenamento) como serviços. Como exemplos, podemos citar *clusters* ou *grids* computacionais, muitas vezes conhecidos como supercomputadores.

- b) Um sistema de informação pode ser visto como um *software* que abrange pessoas, dispositivos, máquinas e métodos que, coletivamente, colecionam, armazenam, processam e disponibilizam dados para seus usuários. Um sistema de informação distribuído é um sistema de informação cujos componentes são distribuídos entre diversos dispositivos computacionais. Um exemplo de um sistemas de informação distribuído são os sistemas da UNIFAL disponíveis a todos os alunos, professores e demais funcionários da universidade. Outro exemplo simples são sistemas bancários.
- c) A definição de sistemas pervasivos está intimamente ligada a definição de sistemas ubíquos. Estes são sistemas de computação que estão distribuídos no nosso ambiente e conectados utilizando redes sem fio, como o wi-fi ou o bluetooth. Normalmente, as pessoas interagem com tais sistemas durante o dia-a-dia. Estes sistemas atuam de forma transparente, autônoma, inteligente e sensíveis ao contexto dos usuários. Um exemplo de sistema pervasivo é uma casa inteligente.

Exercício 5

São características de cada um dos tipos de comunicação:

- **Comunicação transiente:** Esquema de comunicação no qual mensagens só existem no sistema enquanto os componentes (ou aplicações) envolvidos estejam ativos
- **Comunicação persistente:** Esquema de comunicação no qual mensagens *persistem* no sistema, mesmo quando todos os componentes envolvidos na comunicação estejam inativos
- **Comunicação síncrona:** Comunicação na qual tanto o remetente quanto o(s) destinatário(s) ficam bloqueados até que a comunicação tenha sido finalizada
- **Comunicação assíncrona:** Comunicação na qual nem o remetente nem o(s) destinatário(s) ficam bloqueados durante o processo

Desta forma, as respostas para questões a), b), c) e d) são combinações dos itens acima.

Exercício 6

- a) Aplicação direta da Lei de Amdhal:

$$S(n) = \frac{1}{(1 - B) + \frac{1}{n} B} = \frac{1}{(1 - 0,25) + \frac{1}{8} 0,25} = 1,28$$

- b) Na decomposição funcional, existe um ponto de sincronia ao fim da realização de todas as sub-tarefas. Assim, devemos considerar somente o tempo da maior das sub-tarefas. Neste exercício, temos que 80% do algoritmo permite decomposição funcional e os outros 20% devem ser executados de forma serial. Além disto, a maior sub-tarefa corresponde a 50% do tempo da fração paralela, isto é, 40% do tempo total do algoritmo. Então, o speed-up é igual a 50% do tempo da fração paralela. Temos que o speed-up é

$$S(n) = \frac{1}{(1 - B) + 0,5B} = \frac{1}{(1 - 0,8) + 0,5 \times 0,8} = \frac{1}{0,2 + 0,4} = 1,6666667$$

- c) Neste exercício devemos levar em consideração o *speed-up* das duas frações paralelas, o que nos leva a um calculo um pouco mais complexo. Devemos observar que, na decomposição funcional, todas as sub-tarefas gastam exatamente o mesmo tempo para serem executadas. Entretanto, existem somente quatro tarefas, sendo que o computador possui oito núcleos de processamento. Desta forma, pode-se fazer o calculo como se a decomposição funcional como uma decomposição de domínio com quatro processadores. Somando as duas frações paralelas, temos que 50% do algoritmo pode ser paralelizado. Deste modo, seja C a fração do algoritmo onde é possível aplicar a decomposição de domínio e D a fração onde é possível utilizar decomposição funcional. Temos que o speed-up pode ser computado adaptando-se a Lei de Amdhal:

$$S(n) = \frac{1}{(1 - (C + D)) + \frac{1}{8} C + \frac{1}{4} D} = \frac{1}{(1 - (0,3 + 0,2)) + \frac{1}{8} 0,3 + \frac{1}{4} 0,2} = \frac{1}{0,5 + 0,0375 + 0,05} = 1,70$$