

Disciplina DCE540 - Computação Paralela e Distribuída	Método de realização Presencial	Data da prova 16/11/2022 às 14h00
Professor Iago Augusto de Carvalho (iago.carvalho@unifal-mg.edu.br)		

Prova 01

Questão 1 (25 %)

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 de acesso
- b) Transparência de localização
- c) Transparência a falhas
- d) Transparência de migração
- e) Transparência de concorrência

Questão 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

Questão 3 (10%)

O que é um *middleware*? Qual é a importância de *middlewares* para sistemas distribuídos?

Questão 4 (20%)

Dê exemplos de aplicações que utilizem cada um dos tipos de comunicação abaixo. Além disso, diga qual é a vantagem de tal tipo de comunicação para cada aplicação

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

Questão 5 (25%)

Considere um sistema computacional que possui 6 processadores independentes, cada uma com uma *thread*. Calcule o *speed-up* de algoritmos com as seguintes características:

- a) Algoritmo tem uma fração paralela de 75% (decomposição de domínio)
- b) Algoritmo tem uma fração paralela de 25% (decomposição funcional), sendo que a maior das quatro sub-tarefas corresponde a 30% 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 diferenças na representação de dados e no modo de acesso a um recurso, todos conseguem acessar independentemente de como representa os dados e acessa o recurso
- b) Oculta o lugar onde o recurso está localizado, ou seja, não precisa saber onde o recurso está para acessá-lo
- c) Oculta quando ocorre falha e a recuperação de um recurso é feita de forma que o usuário não percebe que processo falhou
- d) Oculta que um objeto pode (ou foi) movido para outra localização física
- e) 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

Middleware é o software distribuído que coordena o sistema distribuído enquanto fornece transparência ao sistema. *Middlewares* são as peças centrais de sistemas distribuídos pois possibilita que

- Programadores construam aplicações sem ser necessário criar interfaces personalizadas para acessar diferentes componentes (serviços ou microserviços), bancos de dados ou dispositivos de computação; e que
- Usuários consigam utilizar um sistema distribuído de maneira simples e transparente, sem se preocupar com a localização física dos aplicativos e recursos do sistema

Exercício 4

- a) RPC: receptor e remetente devem estar ativos durante a comunicação e ficam bloqueados até que a comunicação solicitada tenha sido realizada. **Vantagem:** meio de comunicação barato (computacionalmente falando) e rápido
- b) MPI assíncrona: receptor e remetente devem estar ativos durante a comunicação e não ficam bloqueados até que a comunicação solicitada tenha sido realizada. **Vantagem:** Não é necessário bloquear o processo emissor da mensagem
- c) Certas implementações de filas de mensagens: receptor e remetente não necessitam estar ativos durante a comunicação e ficam bloqueados até que a comunicação solicitada tenha sido realizada. **Vantagem:** Meio de comunicação rápido e eficiente onde as mensagens são persistidas no *middleware*
- d) Caixa de e-mails: receptor e remetente não necessitam estar ativos durante a comunicação e não ficam bloqueados até que a comunicação solicitada tenha sido realizada. **Vantagem:** A base da comunicação por texto moderna na *web*, sendo amplamente utilizada por diversas aplicações. Permite a troca de mensagens de forma fácil e descomplicada

Exercício 5

- a) Aplicação direta da Lei de Amdahl:

$$S(n) = \frac{1}{(1-B) + \frac{1}{n}B} = \frac{1}{(1-0,75) + \frac{1}{6}0,75} = 2,66666667$$

- 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 25% do algoritmo permite decomposição funcional e os outros 75% devem ser executados de forma serial. Além disto, a maior sub-tarefa corresponde a 30% do tempo da fração paralela, isto é, $\frac{25}{100} \frac{30}{100} = 7,5\%$ do tempo total do algoritmo. Seja $x = 1$ o tempo do programa serial. Temos que o speed-up é

$$S(n) = \frac{x}{(x-0,25) + 0,3 \times 0,25x} = \frac{1}{0,75 + 0,075} = \frac{1}{0,875} = 1,1212$$

- c) Neste exercício, temos que todas as sub-tarefas da decomposição funcional gastam exatamente o mesmo tempo para serem executadas. Desta forma, pode-se fazer o calculo como se a decomposição funcional fosse uma decomposição de domínio. Somando as duas frações paralelas, temos que 50% do algoritmo pode ser paralelizado. Deste modo, temos que

$$S(n) = \frac{1}{(1-B) + \frac{1}{n}B} = \frac{1}{(1-0,5) + \frac{1}{4}0,5} = 1,666667$$