



1. Dada a seguinte fórmula da Lei de Amhdal:

$$Speedup = \frac{1}{(1-P) + \frac{P}{N}}$$

tal que P é a proporção de código paralelizável e N é o número de processadores. Avalie as alternativas a seguir:

(F) Caso 10% do código seja paralelizável, o ganho estimado de desempenho com 8 processadores será de 9%

Na verdade o ganho é de 9.5%

(F) Caso 10% do código seja paralelizável, o ganho estimado de desempenho com 4 processadores será de 8%

Na verdade o ganho é de 8.1%

(V) Caso 90% do código seja paralelizável, o speedup estimado com 8 processadores será de até 5 vezes. (4,7)

(F) Caso 90% do código seja paralelizável, o speedup estimado com 4 processadores será inferior a 3 vezes.

Na verdade o speedup estimado é de 3,076 sendo superior a 3 vezes

2. O que é um sistema distribuído?

- I. **É aquele no qual existe concorrência entre os componentes.**
- II. **É aquele no qual os componentes existem em um sistema fracamente acoplado com independência de falhas.**
- III. **É aquele no qual os componentes de hardware e software em uma rede de computadores se comunicam e coordenam apenas por passagem de mensagens.**
- IV. **É aquele no qual os componentes existem em um sistema fortemente acoplado com dependência de falhas.**

Na verdade é exatamente o oposto, ou seja, os sistemas distribuídos são fracamente acoplados com independência de falhas, para que em caso de falhas o sistema continue operando.

- V. **É aquele no qual a concorrência é tratada apenas no middleware.**

pode ser gerenciada em diferentes níveis, pois é uma das características fundamentais dos sistemas distribuídos

Estão corretas as alternativas:

- a) I, II, III
- b) I, II, III, V
- c) I, II
- d) Apenas a III
- e) Nenhuma das alternativas.

3. São características de sistemas distribuídos:

- I. **Visão de um sistema único, baseada em uma rede de computadores.**
- II. Visão de um sistema diferente em cada host, baseado em uma rede de computadores.

Na verdade é o oposto, os sistemas distribuídos buscam dar ,apesar de não ser efetivamente ,a visão de um sistema único.

- III. Alteração da implementação de serviços precisa ser notificada aos usuários, uma vez que estes acessam os recursos dos serviços por meio de interfaces

Não, pois os sistemas distribuídos buscam ser transparentes, logo não precisam notificar, caso seja acessado por interfaces e desde que essas permanecem as mesmas.

- IV. Alteração da implementação de serviços não precisa ser notificada aos usuários, uma vez que estes acessam os recursos dos serviços por meio de links na Web.

De fato as alterações não precisam ser notificadas, porém quando é acessado via links web é necessário notificar, por exemplo, os avisos de manutenção que de vez em quando é possível ver em sites.

- V. **Compartilhamento de recursos com transparência de localização física.**

Está(ão) correta(s) a(s) alternativa(s):

- a) I, III, V
- b) II, III, V
- c) V

d) I, V

e) Nenhuma das anteriores.

5. As opções que definem uma arquitetura de múltiplas camadas são:

I. **Utiliza um ou mais níveis de interação difundidos através de uma hierarquia de processos que lidam com serviços específicos em camadas distintas.**

II. O modelo Model-View-Control é um exemplo porque recebe as requisições na camada de modelagem; a camada de visualização exibe os dados processados, e a camada de controle faz o processamento e controle da interação com as demais camadas.

Na verdade a camada que recebe as requisições é a camada de Control, a camada Model apenas processa os dados, e a Vision apenas exibe os dados processados

III. O modelo Model-View-Control é um exemplo porque recebe as requisições na camada View; a camada Control valida os dados de entrada, e a camada Model prepara os dados para interação com o banco de dados.

Está(ão) correta(s) a(s) alternativa(s):

a) I, III

b) I, II

c) II

d) III

e) Nenhuma das anteriores.

6. Marque as alternativas que melhor definem um middleware:

I. **Camada de software que mascara a heterogeneidade de hardware e software da base.**

II. **Um conjunto de bibliotecas utilizadas por aplicações distribuídas para abstrair os detalhes de comunicação em rede.**

III. Uma máquina virtual que depende do hardware ou software de suporte.

Não, middleware não é uma máquina virtual, é um software que atua como uma camada intermediária entre o S.O. e aplicações que nele rodam

IV. Um framework que simplifica a criação de novos programas sem a necessidade de reinventar os seus métodos e procedimentos.

Na verdade, middlewares não são frameworks.

Está(ão) correta(s) a(s) alternativa(s):

- a) I, II
- b) III, IV
- c) I, II, III
- d) I, III
- e) Nenhuma das anteriores.

7. Avalie as questões a seguir:

I. Um servidor de Proxy fornece um cache compartilhado com o objetivo de aumentar a disponibilidade e desempenho de serviços Web.

II. HTML5 é um exemplo de código-móvel no qual o código executa no navegador do cliente, e interage com o browser do servidor.

III. No paradigma de Passagem de Mensagem (IPC - InterProcess Communication) um processo envia uma mensagem que é uma requisição, a mensagem é entregue para um receptor, que processa a requisição, e retorna uma mensagem em resposta.

Está(ão) correta(s) a(s) alternativa(s):

- a) I, II
- b) II, III
- c) I, III**
- d) Apenas a II
- e) Nenhuma das alternativas.

8. Faça a associação das alternativas:

(1)	Permitir que aplicações e processos executem concorrentemente com distribuição do tráfego para acessar os mesmos recursos.	(2)	Vazão
(2)	A taxa na qual os usuários do sistema distribuído são atendidos	(3)	Interfaces

(3)	Especificam um conjunto de funções disponíveis para invocação (consulta) em um servidor.	(1)	Balanceamento de Carga
(4)	Processos interconectados realizam operações de entrada e saída, similar a operações de E/S em um arquivo.	(5)	Arquitetura Peer-to-Peer
(5)	Cada participante pode fazer uma consulta com o seu par e receber uma resposta.	(4)	Comunicação entre Processos com passagem de mensagens
(6)	Parte do processo que é compartilhado por todas as threads.	(10)	Socket
(7)	Parte de pilha que é exclusivo da execução da thread, e não é compartilhado por outras threads.	(6)	Heap
(8)	Geralmente utiliza uma arquitetura fortemente acoplada para processamento paralelo de alto desempenho.	(7)	Stack
(9)	Geralmente utiliza uma arquitetura fracamente acoplada para processamento paralelo de grande escala.	(9)	Cluster
(10)	É um canal lógico de comunicação com outro processo que utiliza um endereço IP e porta.	(8)	Grade

9. Faça a associação das alternativas:

(1)	Dados locais e remotos são acessados sem necessidade de conhecer a sua localização.	(6)	URL
(2)	Múltiplos clientes podem acessar o mesmo recurso compartilhado, e ao mesmo tempo.	(1)	Transparência

(3)	Aumento de programas, serviços e recursos sem mudança da estrutura do sistema.	(5)	HTTP
(4)	Encapsulados e acessíveis através de interfaces.	(2)	Concorrência
(5)	É um protocolo utilizado por browsers para acessar recursos em servidores Web.	(3)	Escalabilidade
(6)	Browsers utilizam para buscar o recurso em servidores Web	(7)	Servidor
(7)	É um processo que aceita requisições de múltiplos clientes.	(8)	Serviços
(8)	Uma interação completa entre cliente e servidor.	(4)	Invocação Remota
(9)	O principal papel desse servidor é encaminhar dados de uma LAN para outra. Também desempenha papel de firewall.	(9)	Servidor de Proxy
(10)	O principal papel desse servidor é armazenar uma cópia de dados mais próxima dos usuários, com dados recentemente acessados.	(10)	Servidor de Cache

10. Marque V (Verdadeiro) ou F (Falso). Se F, justifique.

(V) Comunicação entre Processos fornece comunicação através de datagramas e fluxo contínuo de bits (stream).

(F) As primitivas de comunicação entre processos suportam comunicação ponto-a-ponto, mas não é possível enviar uma mensagem para um grupo de receptores.

Além de processos ponto-a-ponto é sim possível enviar mensagens para um grupo de receptores por meio de broadcasts e multicasts

(F) Um programa com interface TCP fornece abstração para passagem de mensagem, para que um processo emissor envie um datagrama para o processo receptor usando um socket.

Na verdade quando é usado o protocolo TCP é enviado um fluxo de dados (stream) e no protocolo UDP que se baseia no envio de datagramas.

(F) Um programa Java UDP fornece uma abstração de transporte de fluxo de bytes entre um par de processos.

Na verdade, quando é usado o protocolo UDP é feito o transporte de datagramas, o protocolo que se baseia em transporte de fluxos de dados é o TCP.

(V) Um programa Java TCP com um objeto ServerSocket estabelece um fluxo de bytes entre um par de processos.

11. Observe a sequência de execução das threads a seguir. Marque Verdadeiro ou Falso:

```

run:

pool-1-thread-1 escreveu 1 na posicao 0
Proximo indice: 1
pool-1-thread-1 escreveu 2 na posicao 1
Proximo indice: 2
pool-1-thread-2 escreveu 11 na posicao 0
Proximo indice: 3
pool-1-thread-2 escreveu 12 na posicao 3
Proximo indice: 4
pool-1-thread-2 escreveu 13 na posicao 4
Proximo indice: 5
pool-1-thread-1 escreveu 3 na posicao 2
Proximo indice: 6
Vetor:
[11, 2, 3, 12, 13, 0]
CONSTRUIDO COM SUCESSO (tempo total: 1 segundo)

```

t1 [1

t1 [1 2

t2 [11 2

t2 [11 2 0 12

t2 [11 2 0 12 13

t1 [11 2 3 12 13

(F) A thread2 não sobrescreveu o valor da thread1 ao escrever o valor 11 no vetor.

Só olhar bem q tu vê meu nobre.

(F) Ao final do processo o valor 1 escrito pela thread1 não foi perdido.

Só olhar bem q tu vê meu nobre.

(V) Ao final do processo apenas o valor 1 escrito pela thread1 foi perdido.

(V) Supondo que o índice no vetor comece em 0 (zero), ao final do processo não houve escrita no índice 5 do vetor.

(F) É suficiente atribuir uma maior prioridade para a thread1 para resolver o problema de sobrescrita no vetor.

Pode ajudar a garantir, porém não resolve o problema para resolver é necessário implementar mecanismos de sincronização, para evitar condições de corrida e garantir que os dados sejam acessados de maneira segura.