

INF01209 - Fundamentos de Tolerância a Falhas

Você acessou como João Luiz Grave Gross (Sair)

Moodle do INF ► FTF 2012/2 ► Questionários ► Tolerância a falhas na aplicação ► Revisão da tentativa 1

Tolerância a falhas na aplicação

Revisão da tentativa 1

Terminar revisão

Iniciado em	sexta, 14 setembro 2012, 03:47
Completado em	segunda, 24 setembro 2012, 13:37
Tempo empregado	10 dias 9 horas
Notas	27.58/28
Nota	98.49 de um máximo de 100(98%)

- 1
Notas:
1
- No item 2.1 do artigo: V. D. Florio e C. Blondia, "A survey of linguistic structures for application-level fault tolerance", *ACM Comput. Surv.*, vol. 40, no. 2, p. 6:1–6:37, maio 2008, quando argumentam sobre a necessidade de tolerância a falhas de software, os autores comentam sobre tolerância a falhas de hardware. Assinale **verdadeiro** ou **falso** segundo as opiniões expressas no texto.

Durante muitos anos, a pesquisa na área de tolerância a falhas foi concentrada em desenvolver estruturas de hardware engenhosas e eficientes para lidar com falhas.

Verdadeiro ▼

Durante algum tempo, tolerância a falhas de software foi considerada a única necessária para alcançar a disponibilidade e a integridade de dados demandadas por computadores modernos.

Falso ▼

Componentes de hardware são apenas uma das fontes de não confiabilidade em sistemas computacionais, decrescendo em importância conforme a confiabilidade desses componentes aumenta.

Verdadeiro ▼

A prevalência de falhas de hardware está aumentando continuamente.

Falso ▼

Correto

Notas relativas a este envio: 1/1.

2

Notas:
1

Ainda no item 2.1 do artigo, os autores comentam sobre a complexidade dos sistemas computacionais. Assinale **verdadeiro** ou **falso** segundo as opiniões expressas no texto.

Quanto maior o nível de abstração, maior é a complexidade dos algoritmos que entram em jogo e a consequente propensão a erro das máquinas envolvidas (sejam máquinas reais ou abstratas).

Verdadeiro ▼

A maior parte da complexidade dos modernos sistemas computacionais reside no hardware.

Falso ▼

Dividindo a complexidade do sistema em camadas, falhas ficam confinadas a uma única camada, tornando mais fácil gerenciar a complexidade crescente dos sistemas.

Verdadeiro ▼

Uma forma conceitual eficiente e flexível de gerenciar complexidade é considerar um sistema como uma hierarquia de máquinas abstratas sofisticadas.

Verdadeiro ▼

Mesmo que a divisão em camadas torne a complexidade transparente, a complexidade ainda é parte do sistema global sendo desenvolvido e falhas se propagando por várias camadas podem ocorrer.

Verdadeiro ▼

Parcialmente correta

Notas relativas a este envio: 0.8/1.

3

Notas:
1

Florio e Blondia, no item 2.2, justificam prover tolerância a falhas de software no nível da aplicação baseados no fato de que a maioria dos defeitos experimentados por computadores modernos são devidos a:

Escolher
uma
resposta.

- ☒ a. falhas de software incluindo as falhas no nível de aplicação ✓
- ☐ b. falhas em device drivers e kernel de sistemas operacionais ✗
- ☐ c. quedas de serviço ✗
- ☐ d. falhas de hardware ✗
- ☐ e. falhas de interação com o usuário ✗

Correto

Notas relativas a este envio: 1/1.

4

Notas:
1

Um dos argumentos apresentados por Florio e Blondia para justificar tolerância a falhas de software no nível da aplicação é a adoção popular de componentes de software reusáveis. Muitas aplicações orientadas a objetos são construídas com componentes reusáveis. Considerando reuso de componentes, podemos afirmar, de acordo com a opinião dos autores, que:

Escolher
uma
resposta.

- ☐ a. o reuso de componentes aumenta os custos de desenvolvimento mas contribui para aumento de confiabilidade porque os componentes já foram exaustivamente testados ✖
- ☐ b. o desenvolvedor da aplicação não conhece o desenvolvedor do componente reusável e por essa razão não pode confiar na qualidade do componente ✖
- ☒ c. o reuso de componentes diminui os custos de desenvolvimento mas a confiabilidade desses componentes e o seu impacto na confiabilidade da aplicação são frequentemente desconhecidos ✔
- ☐ d. o reuso de componentes aumenta a complexidade do projeto de sistemas e, além disso, a confiabilidade desses componentes é geralmente inadequada ✖

Correto

Notas relativas a este envio: 1/1.

5

Notas:
1

Um dos argumentos apresentados por Florio e Blondia para justificar tolerância a falhas de software no nível da aplicação, que eles consideram como provavelmente o mais convincente, é chamado de argumento fim-a-fim. Considerando esse argumento e a opinião dos autores, assinale verdadeiro e falso.

frequentemente funções só podem ser completamente e corretamente implementadas com o conhecimento e auxílio da aplicação que se encontra nas extremidades do sistema de suporte

verdadeiro ▼

soluções de tolerância a falhas baseadas puramente em hardware ou no sistema operacional não são capazes de prover tolerância a falhas completa e fim-a-fim na aplicação do usuário

verdadeiro ▼

o argumento implica que todas as tarefas de tolerância a falhas podem ser realizadas no nível da aplicação

falso ▼

como exemplo é citado que um canal de comunicação extraordinariamente confiável livra a aplicação da obrigação de prover confiabilidade

falso ▼

Correto

Notas relativas a este envio: 1/1.

6

Notas:
1

No item 2.3, os autores listam problemas de projeto relacionados a tolerância a falhas no nível da aplicação. Quais foram os problemas mencionados?

Escolher
uma
resposta.

- ☐ a. como incorporar tolerância a falhas no nível da aplicação, como treinar a equipe de desenvolvedores para aplicar tolerância a falhas no nível da aplicação, e como gerenciar o código tolerante a falhas ✖
- ☐ b. como treinar a equipe de desenvolvedores para aplicar tolerância a falhas no nível da aplicação, quais mecanismos de tolerância a falhas devem ser suportados, e como gerenciar o código tolerante a falhas ✖
- ☒ c. como incorporar tolerância a falhas no nível da aplicação, desenvolvedores para quais recursos de tolerância a falhas devem ser suportados, e como gerenciar o código tolerante a falhas ✓
como treinar a equipe de aplicar tolerância a falhas no nível da aplicação, quais mecanismos de tolerância a falhas devem ser suportados, e como gerenciar o código tolerante a falhas
- ☐ d. como reduzir os custos de projeto redundante, como treinar a equipe de desenvolvedores para aplicar tolerância a falhas no nível da aplicação, e como gerenciar o código tolerante a falhas ✖

Correto

Notas relativas a este envio: 1/1.

7

Notas:
1

No item 2.3, os autores listam problemas de projeto relacionados a tolerância a falhas no nível da aplicação. Qual problema foi proposto primeiramente por Randell?

Escolher
uma

- ☐ a. como reduzir os custos de projeto ✖
- ☐ b. como treinar a equipe de desenvolvedores para aplicar

- resposta. tolerância a falhas no nível da aplicação ✖
- ☐ c. como gerenciar o código tolerante a falhas ✖
 - ☐ d. quais mecanismos de tolerância a falhas devem ser suportados ✖
 - ☒ e. como incorporar tolerância a falhas no nível da aplicação ✓

Correto

Notas relativas a este envio: 1/1.

- 8 Randell afirmou ser necessário o uso de técnicas de estruturação
Notas: apropriadas para a incorporação de tolerância a falhas no nível da aplicação.
1 Soluções inadequadas a esse problema, segundo o artigo de Florio e
Blondia, podem provocar .

Correto

Notas relativas a este envio: 1/1.

- 9 A falta de estruturação adequada, segundo o artigo de Florio e Blondia,
Notas: resulta na mistura do código funcional com o código não-funcional. Assinale
1 **verdadeiro** para as consequências ao desenvolvimento do projeto
resultantes dessa mistura, e **falso** quando a alternativa não estiver
relacionada a mistura dos códigos:

- | | |
|---|---|
| o custo e o tempo de desenvolvimento aumentam | <input type="text" value="verdadeiro"/> |
| o tamanho do código resultante é menor | <input type="text" value="falso"/> |
| o código resultante é mais fácil de ser mantido | <input type="text" value="falso"/> |
| a complexidade do sistema diminui | <input type="text" value="falso"/> |
| as tarefas do desenvolvedor ficam significativamente mais complicadas | <input type="text" value="verdadeiro"/> |

Correto

Notas relativas a este envio: 1/1.

- 10 Com relação ao problema de como incorporar tolerância a falhas no nível da
Notas: aplicação, Florio e Blondia concluem que:

1

- Escolher uma resposta.
- ☐ a. a incorporação neste nível é inadequada pois é impossível conciliar interesses ✖
 - ☐ b. os aspectos de tolerância a falhas devem prevalecer em relação aos aspectos funcionais no código da aplicação ✖
 - ☐ c. equipes diferentes de desenvolvedores devem ser encarregadas da parte funcional do código e da parte de

suporte a tolerância a falhas ✖

- ☒ d. uma estrutura ideal de sistema deve garantir uma adequada separação entre os interesses funcionais e os relativos a tolerância a falhas no código da aplicação ✔

Correto

Notas relativas a este envio: 1/1.

11

Notas:
1

Segundo Florio e Blondia, uma resposta inadequada para o problema de qual recurso de tolerância a falhas deve ser suportado pelo nível da aplicação pode apresentar duas características inconvenientes:

Escolher
uma
resposta.

- ☐ a. requerer algoritmos muito sofisticados e estratégias de verificação de alto custo ✖
- ☐ b. requerer desenvolvedores altamente especializados e ferramentas de teste sofisticadas ✖
- ☐ c. exigir equipamento especial e linguagens de programação específicas ✖
- ☒ d. requerer alto grau de redundância e rapidamente consumir a quantidade de redundância disponível ✔

Correto

Notas relativas a este envio: 1/1.

12

Notas:
1

Florio e Blondia definem 3 atributos estruturais para tolerância a falhas no nível da aplicação (ALTF) chamados de SC, SA e A. Associe:

separação de interesses funcionais e não funcionais

SC ▼

suporte direto a um grande conjunto de mecanismos de tolerância a falhas

SA ▼

facilidade do código tolerante a falhas se adaptar ao ambiente onde se encontra no momento

A ▼

Correto

Notas relativas a este envio: 1/1.

13

Notas:
1

Considerando o conceito de redundância como apresentado no item 2.3.1 do artigo de Florio e Blondia, é possível afirmar que:

Tolerância a falhas é, em geral, o resultado de alguma estratégia que efetivamente explora alguma forma de redundância.

Verdadeiro ▼

É impossível acrescentar mais confiabilidade a um canal não confiável apenas aumentando o grau de redundância de informação.

Falso ▼

Redundância pode consumir uma grande quantidade de recursos extras e rapidamente exaurir esses recursos.

Verdadeiro ▼

Redundância requer grande quantidade de recursos extras, mas isso não implica em necessariamente em aumento de custo.

Falso ▼

Correto

Notas relativas a este envio: 1/1.

14

Notas:
1

Florio e Blondia concluem o item 2.3.1 com algumas observações. Assinale verdadeiro para as alternativas que correspondem a conclusões dos autores e falso caso contrário.

Isolar a complexidade fora da aplicação do usuário provoca aumento de custos para a verificação, validação e teste.

Falso ▼

Para qualquer dado nível de redundância, balancear a complexidade do modo de defeito contra a quantidade e tipos de falhas que podem ser toleradas é uma característica importante para uma estrutura tolerante a falhas eficaz.

Verdadeiro ▼

A complexidade é um importante fonte de falhas de projeto.

Verdadeiro ▼

Correto

Notas relativas a este envio: 1/1.

15

Notas:
1

Florio e Blondia afirmam que para um sistema ser tolerante a falhas, recursos de tolerância a falhas devem também ser providos no nível da aplicação. Assinale com X as 6 classes de métodos, mencionadas no item 3 do artigo, que podem ser usadas para prover tolerância a falhas no nível de aplicação.

tolerância a falhas em software de versão única

X ▼

baseada em monitoração da tarefa de recuperação

- ▼

tolerância a falhas em software com múltiplas versões

X ▼

baseada em protocolos para metaobjetos

X ▼

programação orientada a aspectos

X ▼

redundância temporal

- ▼

tolerância a falhas embarcada

- ▼

novas linguagens de programação distribuídas

customizadas

redundância espacial

Parcialmente correta

Notas relativas a este envio: 0.78/1.

16 Qual o requisito chave para o desenvolvimento de sistemas tolerantes a falhas (de acordo com Florio e Blondia, item 3.1)?

Notas:

1

- Escolher uma resposta.
- ☐ a. Replicação de dados ✖
 - ☒ b. Recursos replicados em hardware ou em software ✔
 - ☐ c. Recursos replicados em hardware ✖
 - ☐ d. Replicação temporal ✖
 - ☐ e. Recursos replicados em software ✖

Correto

Notas relativas a este envio: 1/1.

17 A replicação é usada como método fundamental para alcançar tolerância a falhas com múltiplas computações. Segundo Florio e Blondia, quais os seus domínios ?

Notas:

1

- Escolher uma resposta.
- ☐ a. Espaço, paralelismo e codificação de dados ✖
 - ☐ b. Tempo, concorrência e paralelismo ✖
 - ☒ c. Tempo, espaço e informação ✔
 - ☐ d. Codificação de dados, tempo e concorrência ✖

Correto

Notas relativas a este envio: 1/1.

18 Considerando os domínios de replicação para alcançar tolerância a falhas, segundo Florio e Blondia, associe a explicação ao domínio:

Notas:

1

Repetição da computação

Adoção de múltiplos canais de hardware

Adoção de múltiplas versões de software

Correto

Notas relativas a este envio: 1/1.

19 Considerando os domínios de replicação sugeridos por Florio e Blondia (item

Notas: 3.1 do artigo sob análise), associe o significado aos termos na notação
1 nT/mH/pS:

número de programas

p

número de canais de hardware

m

número de execuções

n

Correto

Notas relativas a este envio: 1/1.

20 Um sistema 1T/1H/1S, de acordo com Florio e Blondia, é chamado de:

Notas:

1

Escolher
uma
resposta.

- ☐ a. tolerante a falhas ✖
- ☐ b. intolerante a falhas ✖
- ☐ c. duplex ✖
- ☒ d. simplex ✔
- ☐ e. replicado ✖

Correto

Notas relativas a este envio: 1/1.

21 Tolerância a falhas em versão única de software implica em:

Notas:

1

Escolher
uma
resposta.

- ☒ a. Embutir em um sistema simplex (ou seja, sem réplicas) técnicas de detecção e recuperação de erros. ✔
- ☐ b. Construir programas que não possam ser copiados nem replicados. ✖
- ☐ c. Executar software em ambientes que inibam múltiplos processos e múltiplos usuários simultâneos. ✖
- ☐ d. Construir programas aplicativos sem erros. ✖
- ☐ e. Usar externamente aos programas aplicativos as técnicas de ações atômicas, pontos de recuperação e recuperação por retorno e tratamento de exceções. ✖

Correto

Notas relativas a este envio: 1/1.

22 Segundo Florio e Blondia, a adoção de tolerância a falhas em versão única de software apresenta como desvantagens:

Notas:

1

Escolher
uma
resposta.

- ☐ a. necessidade de optar entre o sistema EFTOS e o sistema SwIFT ✖
- ☐ b. necessidade de desenvolver bibliotecas e frameworks para suprir os recursos de tolerância a falhas ✖

- ☒ c. aumento do tamanho da aplicação com consequente perda de transparência, da facilidade de manutenção e da portabilidade, acompanhados do aumento do tempo de desenvolvimento e dos custos ✓
- ☐ d. obrigação do uso estrito de processos de engenharia de software ✗

Correto

Notas relativas a este envio: 1/1.

23 Considerando o conceito de SV como apresentado por Florio e Blondia, responda verdadeiro ou falso:

Notas:

1

A adoção do SV na camada de aplicação requer concentrar no código fonte da aplicação tanto a especificação do que fazer como a estratégia de tolerância a falhas.

Verdadeiro ▼

SV baseia-se na hipótese que todos os recursos de tolerância a falhas são providos transparentemente pelo hardware, o que gera um código leve, facilmente portátil, de baixo custo e baixa necessidade de manutenção.

Falso ▼

A principal vantagem de incluir em um único código tanto a funcionalidade do sistema como as estratégias de tolerância a falhas é que essa abordagem aumenta a portabilidade do código pois o mesmo carrega consigo todos os recursos necessários.

Falso ▼

Como exemplos de técnicas de detecção e recuperação, que podem ser embutidas em software de versão única, os autores citam ações atômicas, pontos de recuperação e rollback e tratamento de exceções.

Verdadeiro ▼

Correto

Notas relativas a este envio: 1/1.


24 Uma solução parcial, segundo os autores, para a perda de portabilidade e aumento dos custos associados a embutir estratégias de tolerância a falhas em software de versão única seria:

Notas:

1

Escolher uma resposta.

- ☐ a. obrigação do uso estrito de processos de engenharia de software ✗
- ☐ b. o uso do sistema EFTOS e do sistema SwiFT ✗
- ☒ c. o uso de bibliotecas e frameworks criados seguindo criteriosamente processos de engenharia de software ✓
- ☐ d. uso eficiente de linguagens de baixo nível ✗

- ☐ e. o uso de estratégias como ações atômicas, recuperação por retorno e tratamento de exceções 

Correto

Notas relativas a este envio: 1/1.

25 Considerando o sistema SwiFT, assinale as sentenças verdadeiras e falsas.

Notas:

1

SwiFT inclui uma série de componentes reutilizáveis de software adequados para incluir técnicas de tolerância a falhas em software de versão única

Verdadeiro 

SwiFT é o nome de um computador tolerante a falhas que foi popular na década de 90

Falso 

SwiFT introduziu apenas um pequeno overhead em custo na maioria dos casos

Verdadeiro 

SwiFT foi usado com sucesso para aumentar a tolerância a falhas em sistemas de software onde falhas residuais estavam presentes

Verdadeiro 

SwiFT é o nome de um projeto que visava construir componentes de hardware para facilitar a execução de software tolerante a falhas

Falso 

Correto

Notas relativas a este envio: 1/1.

26 Florio e Blondia definem no item 2.3 os conceitos de SC (separação de interesses funcionais e de tolerância a falhas), SA (adequação sintática) e A (adaptabilidade). Em relação a software de versão única, SV, como os autores avaliam esses atributos? Indique suficiente ou insuficiente.

Notas:

1

separação de interesses

insuficiente 

adequação sintática

insuficiente 

adaptabilidade

insuficiente 

Correto

Notas relativas a este envio: 1/1.


27 Na conclusão do item 3.1.1, Florio e Blondia, após reforçarem uma série de desvantagens da abordagem de versão única, alertam para uma característica importante dessa abordagem. Qual é ela?

Notas:

1

Escolher uma resposta.

☐ a. apresenta suficiente adaptabilidade 

☐ b. apresenta um alto grau de portabilidade e facilidade de manutenção 

- ☒ c. não apresenta qualquer restrição quanto a classe de aplicação que pode se beneficiar da abordagem ✓
- ☐ d. não apresenta impacto nos custos de desenvolvimento e manutenção ✗

Correto

Notas relativas a este envio: 1/1.

28

Notas:
1

Na conclusão do item 3.1.1, Florio e Blondia, após reforçarem uma série de desvantagens, falam das vantagens de utilizar, em software de versão única, ferramentas que permitem ao usuário tratar com átomos de tolerância a falhas sem se preocupar com sua implementação. Em relação a essas ferramentas podemos afirmar que:

- Escolher uma resposta.
- ☐ a. só podem ser usadas em programas escritos em C ou C++ ✗
 - ☐ b. introduzem grande overhead no desempenho para apenas um pequeno aumento nos atributos de dependabilidade ✗
 - ☐ c. não apresentam qualquer impacto no custo ou no desempenho da aplicação ✗
 - ☐ d. prejudicam a dependabilidade do sistema ✗
 - ☒ e. permitem ao projetista reusar peças de software sofisticadas e amplamente testadas ✓

Correto

Notas relativas a este envio: 1/1.

Terminar revisão

Você acessou como [João Luiz Grave Gross \(Sair\)](#)

FTF 2012/2
