

SISTEMAS DISTRIBUÍDOS

CONSISTÊNCIA E REPLICAÇÃO

CONSISTÊNCIA E REPLICAÇÃO

- Uma questão importante em sistemas distribuídos é a replicação de dados
- Dados são replicados para aumentar a confiabilidade ou melhorar o desempenho
- A replicação é útil, mas traz dificuldades
 - Gerenciamento de réplicas => Servidores de réplicas e distribuição de conteúdo
 - Consistência entre as réplicas => Modelos de consistência e implementação da consistência

RAZÕES PARA REPLICAÇÃO

Confiabilidade

- Dados são replicados para aumentar a confiabilidade do sistema
- Se um sistema de arquivos for replicado é possível continuar o trabalho mesmo que uma das réplicas caia
- Melhor proteção contra dados corrompidos
 - Um valor correto pode ser retornado por no mínimo duas cópias

RAZÕES PARA REPLICAÇÃO

Desempenho

- Replicação para aumentar desempenho é importante quando um SD precisa ser ampliado em quantidade e área geográfica
- Ampliação em quantidade
 - Um número cada vez maior de processos => para único servidor
 - Divide trabalho de servidor centralizado, diminuindo o esforço para cada servidor
- Ampliação geográfica
 - Criar uma cópia de dados próxima ao cliente => para diminuir tempo de acesso a dados, aumentando o desempenho dos clientes

REPLICAÇÃO

O problema da replicação

- A existência de múltiplas cópias pode levar a problemas de consistência
 - Sempre que uma cópia é modificada ela se torna diferente das restante
 - É necessário atualizar todas as réplicas, o que pode gerar, no mínimo, em maior largura de banda para manter consistência dos dados

REPLICAÇÃO - EXEMPLO

Melhorar tempo de acesso de páginas Web

- Browsers costumam armazenar no local uma cópia de uma página Web que já foi buscada anteriormente => cache
- Se um usuário requisitar aquela página mais uma vez, o browser automaticamente retorna a cópia local
- Tempo de acesso excelente, mas e se o usuário quiser ter a versão mais recente da página?
 - Proibir cópia local e deixar para o servidor a responsabilidade de replicação => Se não existir cópia próxima ao usuário?
 - Servidor Web responsável por monitorar e atualizar todas as caches => pode degradar desempenho global do servidor

REPLICAÇÃO COMO TÉCNICA DE CRESCIMENTO (1/4)

- Questões de escalabilidade aparecem sob a forma de problemas de desempenho
- Cópias de dados mais próximas dos processos que as estão usando pode melhorar o desempenho pela redução do tempo de acesso, resolvendo o problema de escalabilidade!
- Manter cópias atualizadas pode requerer mais largura de banda de rede!

DESEMPENHO X ESCALABILIDADE

REPLICAÇÃO COMO TÉCNICA DE CRESCIMENTO (2/4)

- Supondo que um processo P acessa uma réplica N vezes por segundo
- A réplica é atualizada M vezes por segundo
- Se $N \ll M$, muitas versões da réplica nem serão vistas pelo processo P
- Neste caso é importante avaliar:
 - Se há realmente a necessidade de instalar a réplica próxima ao processo P
 - A utilização de uma estratégia diferente para atualizar a réplica

REPLICAÇÃO COMO TÉCNICA DE CRESCIMENTO (3/4)

- Manter várias cópias consistentes pode estar sujeito a problemas de desempenho
- Um conjunto de cópias é consistente se todas forem sempre iguais
- Logo, quando uma operação de atualização é realizada sobre uma cópia, a atualização deve ser propagada para todas as outras cópias
-
- Manter a consistência entre todas as cópias em um sistema distribuído de larga escala => **Muito custoso!**

REPLICAÇÃO COMO TÉCNICA DE CRESCIMENTO (4/4)

- Dilema:
 - Problemas de escalabilidade podem ser amenizados pela aplicação de replicação e cache => resultando em melhor desempenho
 - Manter todas as cópias consistentes => requer sincronização global => muito custoso em termos de desempenho
- A cura saiu pior que a doença?
 - Não se pudermos relaxar as restrições de consistência!

MODELOS DE CONSISTÊNCIA

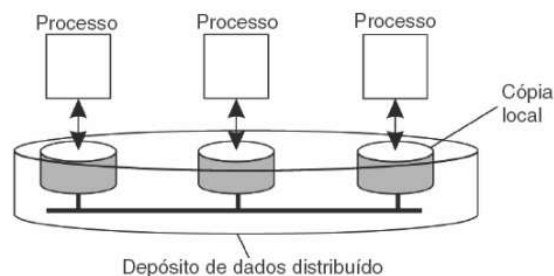
- Modelo de Consistência é um contrato entre processos e depósitos de dados
 - Se os processos concordarem em obedecer certas regras, o depósito promete funcionar de maneira correta
- Normalmente, um processo que executa uma leitura em um dado, espera como retorno um valor que mostre o resultado da última operação de escrita naquele dado
 - Sem um relógio global é difícil definir qual foi a última escrita
 - Como alternativa, são usadas outras definições/regras estabelecidas por um modelo de consistência
 - Quais inconsistências são toleráveis?

MODELOS DE CONSISTÊNCIA CENTRADOS EM DADOS

- A consistência tem sido discutida no contexto de operações de leitura e escrita em dados compartilhados por meio de:
 - Memória compartilhada (distribuída) >>
 - Banco de dados (distribuído) >> Depósito de Dados
 - Sistema de arquivos (distribuído) >>
- Cada processo que pode acessar dados do depósito tem uma cópia local (ou próxima) disponível referente ao depósito inteiro

MODELOS DE CONSISTÊNCIA CENTRADOS EM DADOS

- Operações de escrita são propagadas para outras cópias
 - Se a operação alterou dados => Escrita
 - Se a operação não alterou dados => Leitura



Organização geral de um depósito de dados lógico, fisicamente distribuído e replicado por vários processos.

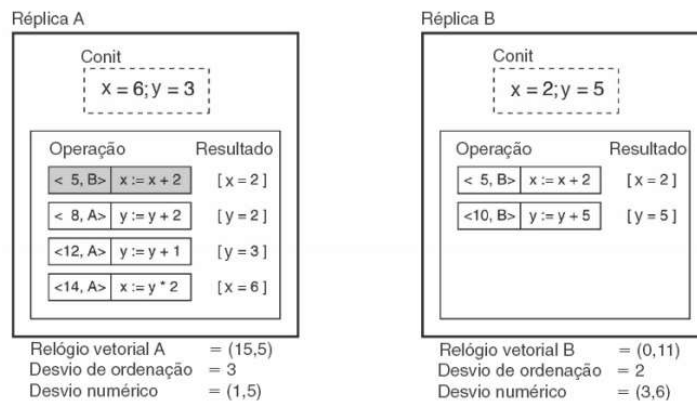
CONSISTÊNCIA CONTÍNUA

Definidas a partir de três eixos:

1. Desvio em valores numéricos entre réplicas
 - Cópias não podem variar por mais que valor absoluto ou relativo
 - Número de atualizações não vistas por outras réplicas
2. Desvio em idade entre réplicas
 - Última vez que a réplica foi atualizada
3. Desvio em relação às operações de atualização
 - Tolerância ordenação diferente de atualizações nas réplicas
 - As atualizações podem ser aplicadas temporariamente em uma cópia local a espera de um acordo global, podendo serem refeitas na ordem correta antes de se tornarem permanentes

CONSISTÊNCIA CONTÍNUA

Conit: especifica a unidade segundo o qual a consistência deve ser medida (exemplo: número de atualizações da previsão de tempo).

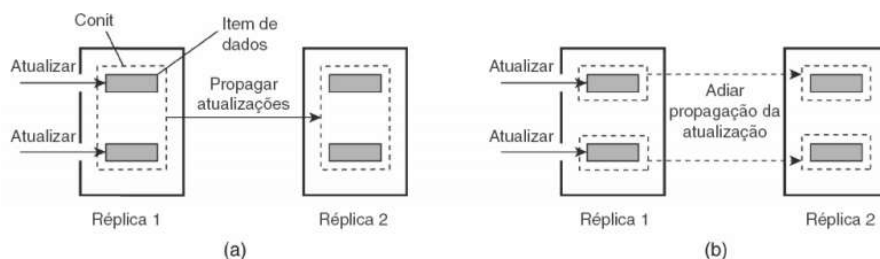


Exemplo de monitoração de desvios de consistência (adaptado de Yu e Vahdat, 2002).

CONSISTÊNCIA CONTÍNUA

Necessário compromisso para manter **conits** de granularidade grossa e **conits** de granularidade fina

- Exemplo: diferença de duas réplicas não pode ter mais que uma atualização pendente



Escolha da granularidade adequada para uma conit. (a) Duas atualizações resultam em propagação da atualização. (b) Nenhuma propagação de atualização é necessária (ainda).

CONSISTÊNCIA CONTÍNUA

- Necessário tratar duas questões para se por as **conits** em prática:
 - Para impor consistência é necessário ter protocolos
 - Programadores devem especificar os requisitos de consistência para aplicações
- Os programadores não estão acostumados com replicação, muito menos em fornecer informações sobre consistência
- Para implementar **conits** é obrigatório que haja interfaces de programação simples e fáceis de entender
 - Pode ser implementada como um conjunto de ferramentas parecido com bibliotecas
 - AffectsConit, DependsOnConit

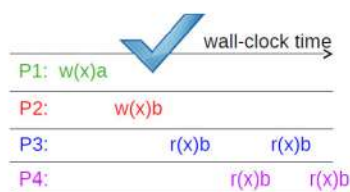
CONSISTÊNCIA SEQUENCIAL

- Um depósito de dados é sequencialmente consistente quando satisfaz a seguinte condição:
 - O resultado de qualquer execução é o mesmo que seria se as operações (de leitura e escrita) realizadas por todos os processos no depósito de dados fossem executadas na mesma ordem sequencial
- Todos os processos executam todas as operações de leitura e escrita em uma mesma ordem global, e as operações em um processo aparecem na ordem que são emitidas pelo programa que está sendo executado pelo processo

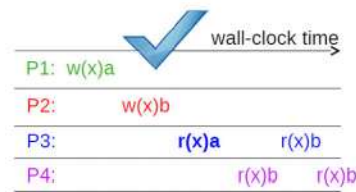
CONSISTÊNCIA SEQUENCIAL

Portanto:

- Leituras podem ficar obsoletas em relação ao tempo real, mas não em relação à hora lógica
- As operações de escrita em todas as réplicas estão totalmente ordenadas na mesma ordem e acordo com a hora lógica



Qual ordem global pode explicar o resultados acima?
 Na ordem do relógio real. Consistência sequencial e restrita



Qual ordem global pode explicar o resultados acima?
 w(x)a, r(x)a, w(x)b, r(x)b, ...
 Consistência sequencial mas não restrita

CONSISTÊNCIA CAUSAL

A Consistência Causal representa um enfraquecimento da consistência sequencial no sentido que faz distinção entre eventos que são potencialmente relacionados por causalidade e os que não são:

- Se o evento b é causado ou influenciado por um evento anterior a, então é necessário que todos vejam primeiro a e, depois, b
- Se o processo P1 escreve x, em seguida P2 lê x e escreve y, os dois primeiros são causais, porém as duas escritas são concorrentes.

CONSISTÊNCIA CAUSAL

A Consistência Causal representa um enfraquecimento da consistência sequencial no sentido que faz distinção entre eventos que são potencialmente relacionados por causalidade e os que não são:

- Se o evento b é causado ou influenciado por um evento anterior a, então é necessário que todos vejam primeiro a e, depois, b
- Se o processo P1 escreve x, em seguida P2 lê x e escreve y, os dois primeiros são causais, porém as duas escritas são concorrentes.

CONSISTÊNCIA CAUSAL

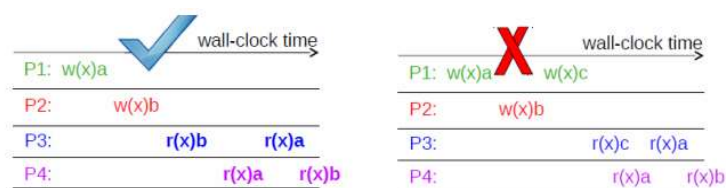
Para um depósito de dados ser considerado consistente por causalidade, é necessário que obedeça à seguinte condição:

- Escritas que são potencialmente relacionadas por causalidade devem ser vistas por todos os processos na mesma ordem
- Escritas concorrentes podem ser vistas em ordem diferente em máquinas diferentes

CONSISTÊNCIA CAUSAL

- Qualquer execução ocorre como se todas as operações de leitura/escrita que tem relação de causalidade entre elas tivessem sido executadas em uma ordem que reflete a causalidade entre elas
- As operações concorrentes, ou sejam, que não tem relação de causalidade, podem ser executadas em ordens diferentes nas diferentes réplicas

CONSISTÊNCIA CAUSAL



Só são obedecidas as restrições de ordem por cada processo:
 $w(x)b < r(x)b$; $r(x)b < r(x)a$; ...
 $w(x)a \parallel w(x)b$, então estas operações podem ser vistas por diferentes processos em ordens diferentes
 Não é consistente sequencialmente.

Como P3 leu c ($r(x)c$), ele tem que continuar a ler c ou algum valor mais novo (talvez b), mas não pode voltar para o valor a, pq $w(x)c$ está condicionado a $w(x)a$ ter terminado

Curso: ENGENHARIA DE COMPUTAÇÃO
Disciplina: SISTEMAS COMPUTACIONAIS DISTRIBUÍDOS
Professor: KLAUSNER VIEIRA GONÇALVES

COMPLEMENTO

Sistemas Distribuídos - Aula 13 – Consistência e Replicação

<https://www.youtube.com/watch?v=YuXcsNNO8CU>

Sistemas Distribuídos - Aula 14 - Abordagens de Consistência e Replicação

https://www.youtube.com/watch?v=R45qIXz_sI4