

# Replicação

## Sistemas Distribuídos

abril de 2015

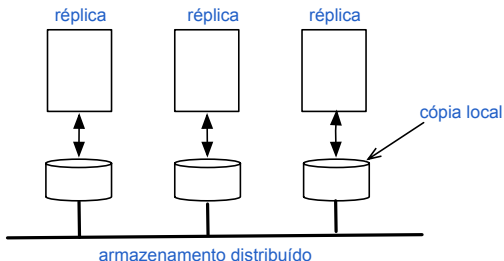


- tolerância a falhas
  - disponibilidade
- desempenho
  - proximidade
  - divisão da carga de trabalho: escalabilidade

- Se várias cópias de um serviço estão disponíveis, como ficam seus dados?
- intuitivamente, gostaríamos que o resultado de uma operação fosse independente da réplica utilizada...
  - grau de acoplamento muito forte: custo alto!
    - clientes com caches
    - replicação de servidores
  - uso de *modelos de consistência* menos rígidos

# Modelos centrados em dados

- modelos de consistência tradicionalmente discutem resultado de *leituras e escritas*
- podemos pensar em termos de serviços de *armazenamento de dados*
  - um cliente trabalha sobre uma réplica dos dados



## definição:

- qualquer leitura de item de dado  $x$  retorna o valor correspondente à escrita mais recente em  $x$
- dificuldades até na definição de “mais recente”
- dependência em tempo global

## definição:

- O resultado de qualquer execução é compatível com alguma execução sequencial (intercalada) das operações (de leitura e escrita) realizadas pelos processos do grupo.

P1:	W(x) a		
P2:		W(x) b	
P3:		R(x) a	R(x) b
P4:		R(x) b	R(x) a

P1:	W(x) a		
P2:		W(x) b	
P3:		R(x) b	R(x) a
P4:		R(x) b	R(x) a

- preocupação apenas com consistência de leituras e escritas causalmente relacionadas

causalmente inconsistente:

P1:	W(x) a		
P2:	R(x) a	W(x) b	
P3:			R(x) a R(x) b
P4:			R(x) b R(x) a

- garantia de que operações são observadas na ordem de possível precedência causal

P1:	W(x) a	
P2:	W(x) b	
P3:		R(x) a R(x) b
P4:		R(x) b R(x) a



- as operações de escrita feitas por um processo devem ser vistas na ordem correta por todos os demais, mas não há imposições sobre a ordenação das escritas realizadas por diferentes processos

P1:	W(x) a				
P2:	R(x) a	W(x) b	W(x) c		
P3:			R(x) b	R(x) a	R(x) c
P4:			R(x) a	R(x) b	R(x) c

# Modelos centrados no cliente

- assumem que clientes fazem muito mais leituras do que escritas e que há poucas escritas concorrentes
- clientes aceitam alguns atrasos em suas visões da informação
- modelos centrados no cliente provêm garantias do ponto de vista de um cliente

## alterações no modelo

- cliente agora pode “migrar” entre servidores
- conceito de *write set*: conjunto de escritas que levou a determinado valor para dado em determinada réplica (WS)



## exemplos de aplicações

- Web
  - DNS
  - LDAP
- cenário bastante comum em diversas aplicações

P1:	W(x) a	W(y) b			
P2:		R(x) nil	R(x) a		
P3:			R(y) nil	R(x) nil	R(x) a
P4:		R(y) b			

## Escritas monotônicas

- escritas em determinado item são sempre completadas antes da próxima escrita no mesmo item de dados (em qualquer réplica, pelo mesmo cliente)

R1:	WS(x1)
R2:	WS(x1;x2)

R1:	WS(x1)
R2:	WS(lx1;x2)

escritas feitas por um mesmo cliente em réplicas diferentes

# Implementação: Arquiteturas de Replicação

- cópia primária
- replicação ativa

# Replicação baseada em cópias primárias

- cada item de dados (ou todos) estão associados a um servidor primário
- leituras ocorrem em todas as cópias
- escritas:
  - 1 sempre no servidor primário
    - atualização em réplicas dispara atualização remota em servidor primário
    - atualização em servidor primário pode ou não bloquear até atualização ser propagada a todas as cópias
  - 2 local-write: cópia do dado primário transferido para local da atualização

- todas as réplicas realizam escritas “simultaneamente”
- necessidade de um protocolo que garanta uma determinada ordem na realização das operações
  - *broadcast* total ou causal
- ou consenso sobre estado/atualizações
  - algoritmo de consenso

- réplicas de dados
- réplicas de servidores



- clusters — replicação sem distribuição geográfica
- criação de réplicas em resposta a padrões de acesso
- modelos de previsão de acessos
- dinamismo: sistemas p2p

- réplicas permanentes
  - réplicas iniciadas pelo servidor
  - réplicas iniciadas pelo cliente
- 
- dificuldade de integrar nas aplicações a localização da melhor réplica
    - relação com discussão sobre sistemas p2p
  - dinamismo

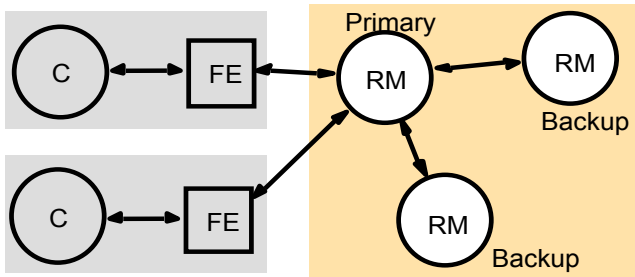
## organização de processos em grupos

- gerência de participantes
- broadcast e garantias
  - consistência centrada em dados X centrada em mensagens

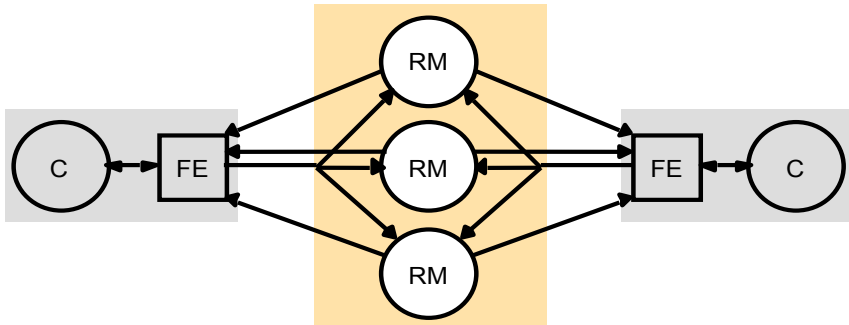
## modelo mais comum:

- tolerância a falhas *fail-stop* com uso de uma réplica primária e n secundárias

# Replicação de processos com cópia primária



# Replicação de processos com cópia primária



# tolerância a falhas e checkpoints