

Replicação

Sistemas Distribuídos

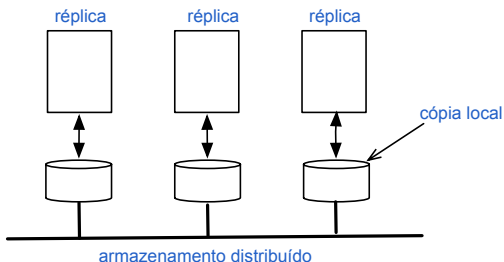
maio de 2017

- 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

Consistência Sequencial

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 |

- 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 |

- 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 |

Consistência FIFO

- 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 |

- garantias do ponto de vista de um cliente

alterações no modelo

- cliente não está amarrado a uma réplica específica

- ênfase em consistência: cliente precisa saber lidar com indisponibilidade para escritas
- ênfase em disponibilidade: aplicações que podem conviver com dados um pouco desatualizados
 - consistência *fraca* e janelas de inconsistência
 - DNS
 - LDAP

exemplos de aplicações

- Web
- DNS
- LDAP
- servidores de jogos

- cenário bastante comum em diversas aplicações
- *janela de inconsistência*

| | | | | | |
|-----|--------|----------|--------|----------|-----------------|
| 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 | | | |

- um determinado processo sempre enxerga as atualizações já feitas por ele mesmo

Leitura monotônicas

- leitura nunca retorna valores anteriores aos já vistos pelo mesmo processo

Escritas monotônicas

- sistema garante serialização das escritas de um mesmo processo

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

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

escritas feitas por um mesmo cliente em réplicas diferentes

tradeoffs

- consistência
- disponibilidade
- particionamento da rede

- Eric Brewer. Towards robust distributed systems (abstract). in *Proc. 19th ACM Symposium on Principles of Distributed Computing*. 2000.

Servidores: 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

grupo de servidores

- N réplicas
 - W réplicas precisam confirmar atualizações para completá-las
 - R réplicas contactadas na leitura
-
- se $W + R > N$ temos consistência forte
 - replicação com cópia primária e modo síncrono: $W = N$
 - replicação com cópia primária e modo assíncrono:
 $W = 1, R = 1$

- foco de sistema e *trade-offs*: disponibilidade, tolerância a falhas, consistência, tempo de acesso, ...
 - foco em tolerância a falhas com consistência:
 $N = 3, R = 2, W = 2$
 - foco em tolerância a falhas sem consistência:
 $N = \text{MUITOGRANDE}, R = 1, W = 1$
 - otimizações de leituras ($W = N$ e $R = 1$) ou de escritas ($W = 1$ e $R = N$)?

- 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

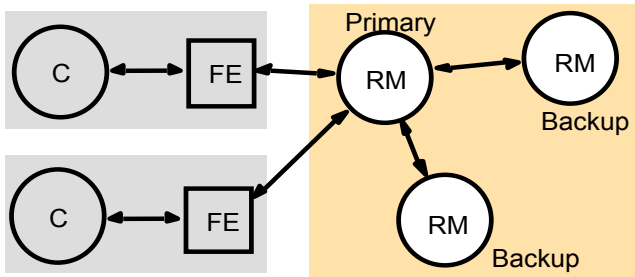
Localização de réplicas

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

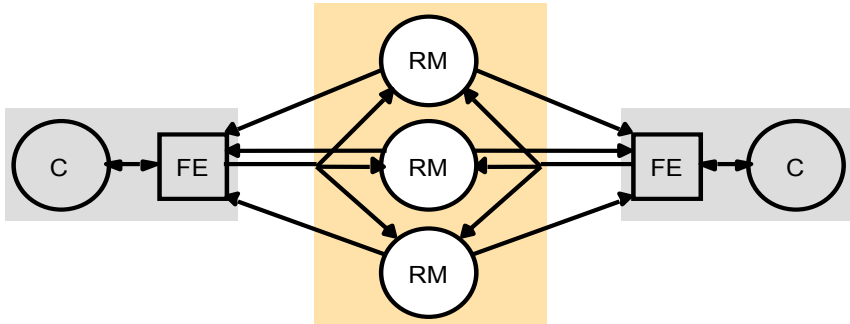
modelo mais comum:

- 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

- W. Vogels. Eventually Consistent. *Communications of the ACM*. 52(1). 2009.
- E. Brewer. CAP twelve years later: How the “rules” have changed. *Computer*. 45(2). 2012.