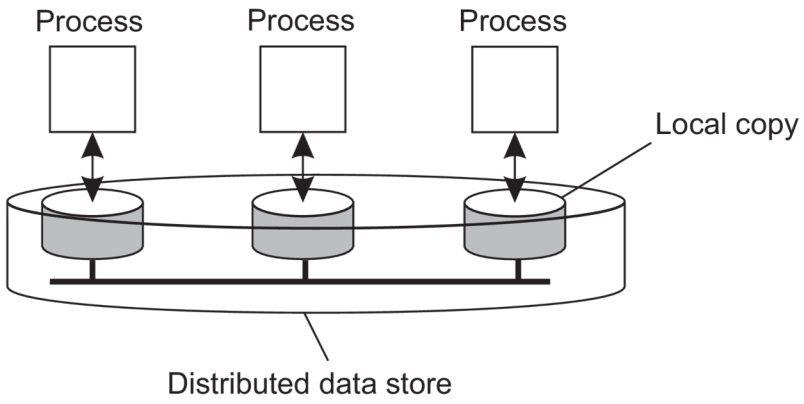
Consistência e Replicação

Performance e Escalabilidade

- Para manter consistência entre réplicas, é geralmente necessário garantir que as operação conflituantes são feitas pela mesma ordem.
- Operações que geram conflitos:
  - Read-write: operação de leitura e escrita concorrentes
  - Write-write: duas operações de escrita
- Garantir uma ordem global em operações conflituantes pode ser uma operação onerosa, o que baixa a escalabilidade.
  - Solução: diminuir requisitos de consistência para que a necessidade de sincronismo global seja evitada.

Modelos de consistência Data-centric

- Modelo de consistência
  - Um contrato entre uma data store (distribuída) e processos, em que a data store especifica precisamente qual o resultado de uma operação de leitura e escrita em concorrência.

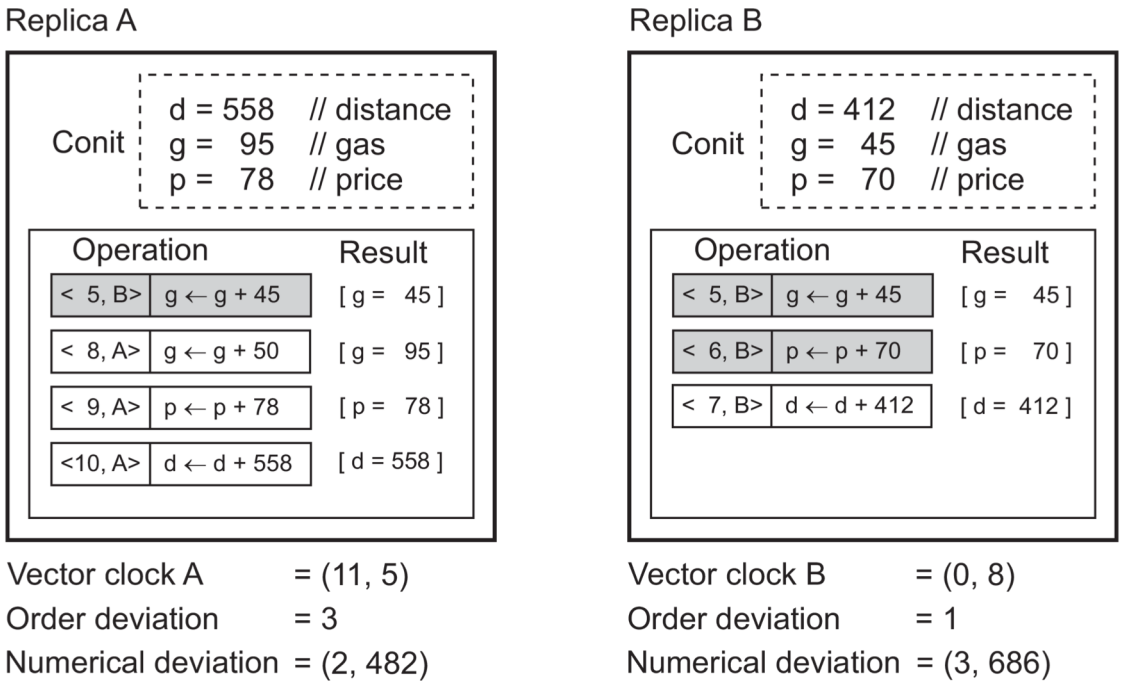


Consistência contínua

- Podemos falar do grau de consistência:
  - Replicas podem diferir no seu **valor numérico**
  - Replicas podem diferir na sua **imutabilidade** relativa
  - Podem existir diferenças no **número e ordem de operações de actualização (update)**
- Conit
  - Unidade de consistência => especifica a **unidade de dados** sobre a qual deve ser medida a consistencia

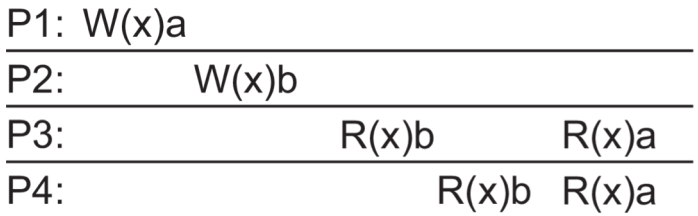
Exemplo: Conit

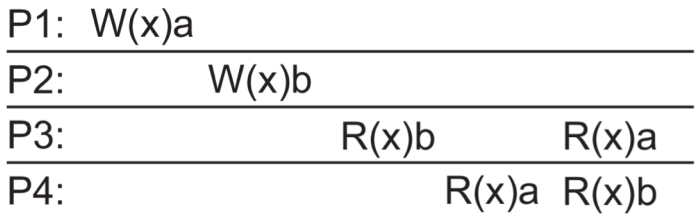
- Cada réplica tem um vector clock
- B envia a A operação a cinzento
  - A executou permanentemente essa operação
- Conit:
  - A tem 3 operações pendentes:
    - Ordem desvio = 3
  - A perdeu 2 operações de B:
    - Max diff é 70 + 412 = (2, 482)



Consistência Sequencial

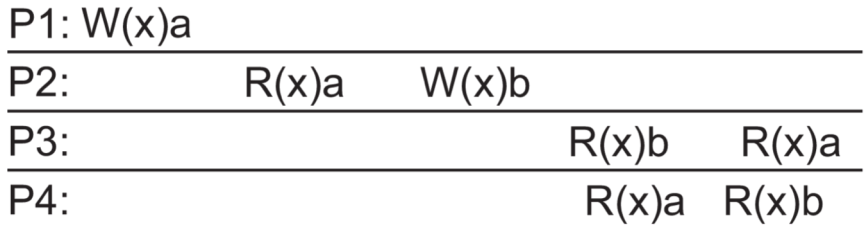
- Definição
  - O resultado de qualquer execução é o mesmo se todos operações em todos processos forem executadas na mesma ordem sequencial, e as operações de cada processo individual aparecerem na sequencia na ordem especificada pelo programa.
- (a) Armazenamento consistente sequencialmente. (b) Armazenamento não consistente sequencialmente

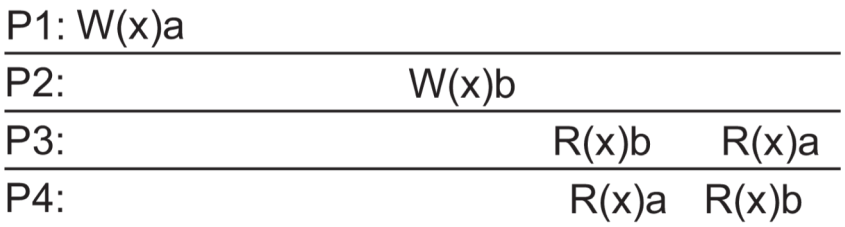




Consistência causal

- Escritas que estão relacionadas casuisticamente, têm que ser vistas por todos processos com a mesma ordem. Escritas concorrentes podem ser vistas com ordens diferentes por diferentes processos.
- (a) Um violação de armazenamento casuístico. (b) Sequencia correcta de eventos num armazenamento casuístico.





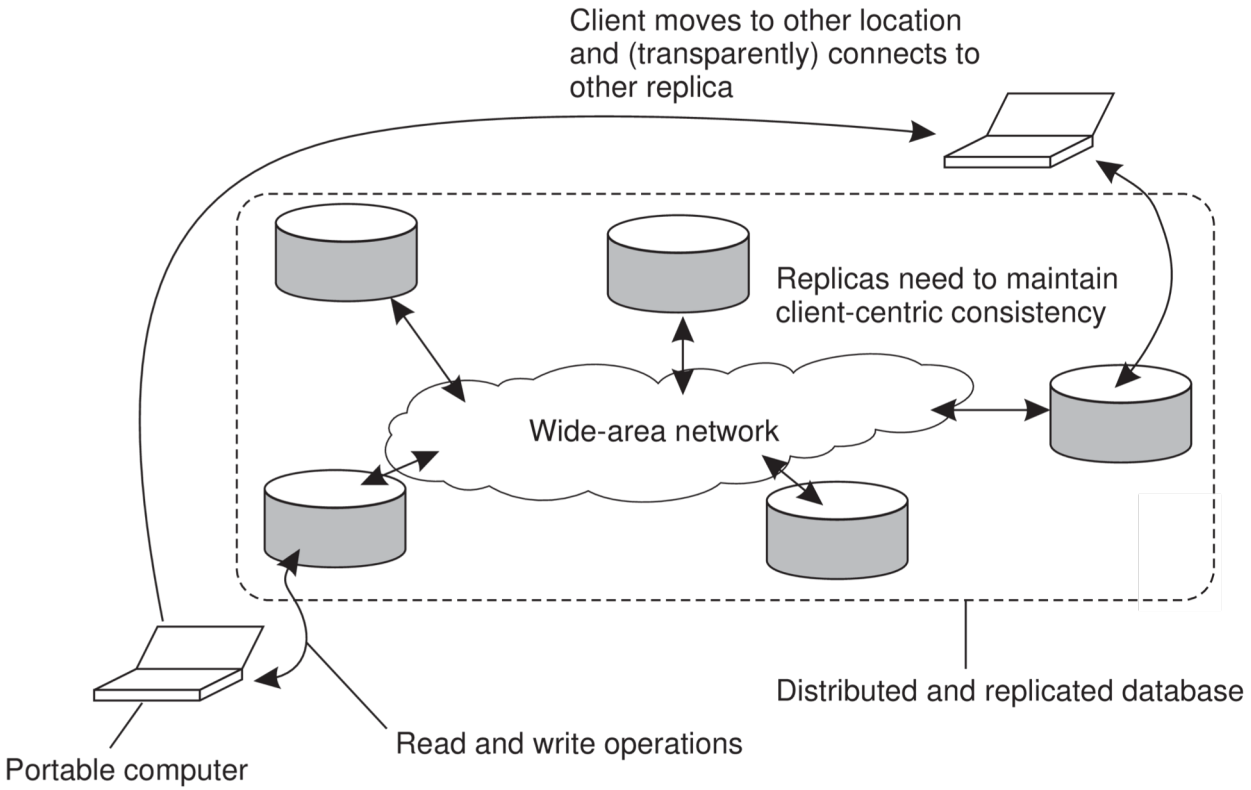
Agrupar Operações

- Definição:
  - Acesso a **locks** são consistentes sequencialmente
  - Não são permitidos acessos a **locks**, antes que todas escritas prévias tenham sido completadas
  - Nenhum acesso a dados é permitido até que todos acessos anteriores a **locks** tenham sido feitos.
- Ideia base
  - Não interessa que as leituras e escritas de uma **série** de operações seja imediatamente conhecida por outros processos. Apenas queremos que o **efeito** da serie seja conhecido.

Agrupar Operações (2)

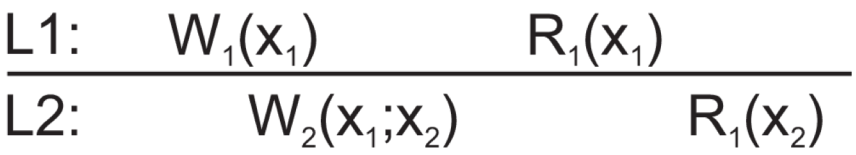
P1:	L(x)	W(x)a	L(y)	W(y)b	U(x)	U(y)					
P2:							L(x)	R(x)a		R(y)	NIL
P3:									L(y)	R(y)b	

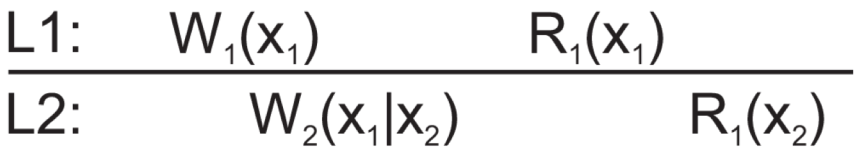
Modelos Client centric - Consistência Eventual



Leituras Monotónicas

- Definição
  - Se um processo ler o valor do dado x, qualquer operação de leitura sucessiva sobre x pelo mesmo processo retorna o mesmo valor ou mais recente.
- (a) leitura monotónica (b) leitura não monotónica





Escritas Monotónicas

- Definição
  - Uma operação de escrita por um processo num dado x é completada antes de qualquer operação de escrita sucessiva em x pelo mesmo processo.
- (a) escrita monotónica (b) data store que não disponibiliza consistência por escrita monotónica (c) sem consistência por escrita monotónica (d) Consistente pois WS(x_1; x_3) apesar de x_1 aparentemente reescrever x_2.

L1:	$W_1(x_1)$	
L2:	$W_2(x_1;x_2)$	$W_1(x_2;x_3)$

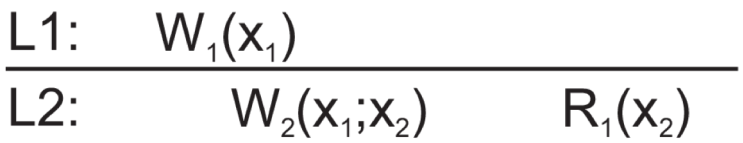
L1:	$W_1(x_1)$	
L2:	$W_2(x_1\|x_2)$	$W_1(x_1\|x_3)$

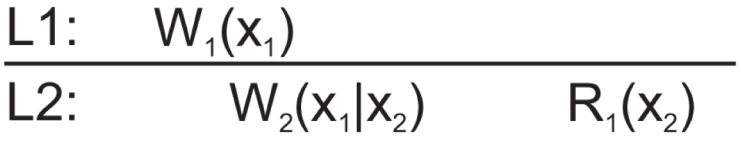
L1:	$W_1(x_1)$	
L2:	$W_2(x_1\|x_2)$	$W_1(x_2;x_3)$

L1:	$W_1(x_1)$	
L2:	$W_2(x_1\|x_2)$	$W_1(x_1;x_3)$

Leitura após escrita

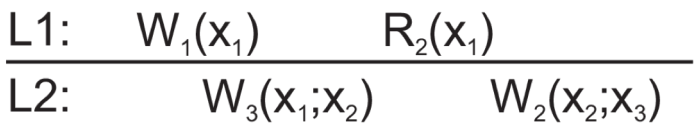
- Definição
  - O efeito da operação de escrita por um processo nos dados x, será sempre visto pelas operações de leitura sucessivas em x pelo mesmo processo.
- (a) Leitura após escrita consistente (b) Não consistente

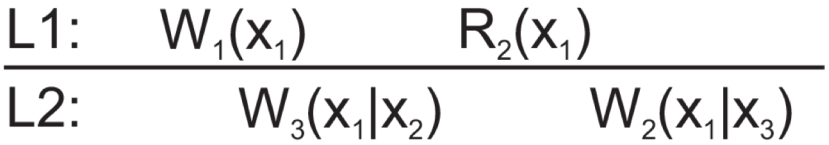




Escrita após leitura

- Definição
  - Numa operação de escrita por um processo sobre os dados x após uma operação de leitura prévia sobre x pelo mesmo processo, é garantido que ocorre sobre o mesmo valor de x que foi lido ou mais recente.
- (a) Escrita após leitura consistente (b) não consistente



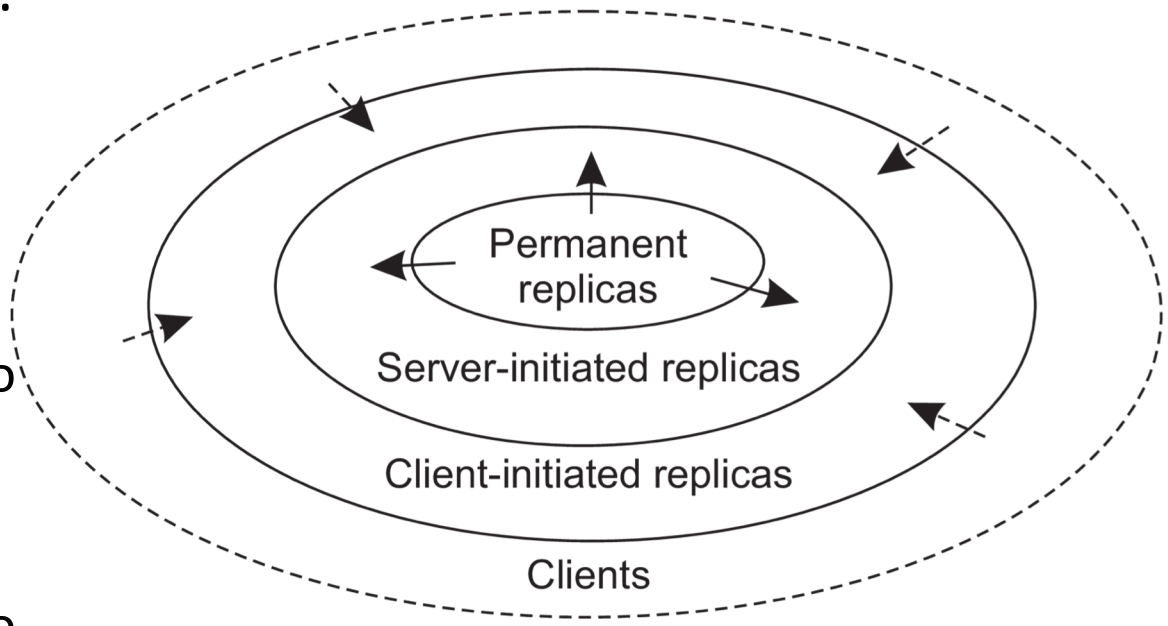


Localização de Replicas

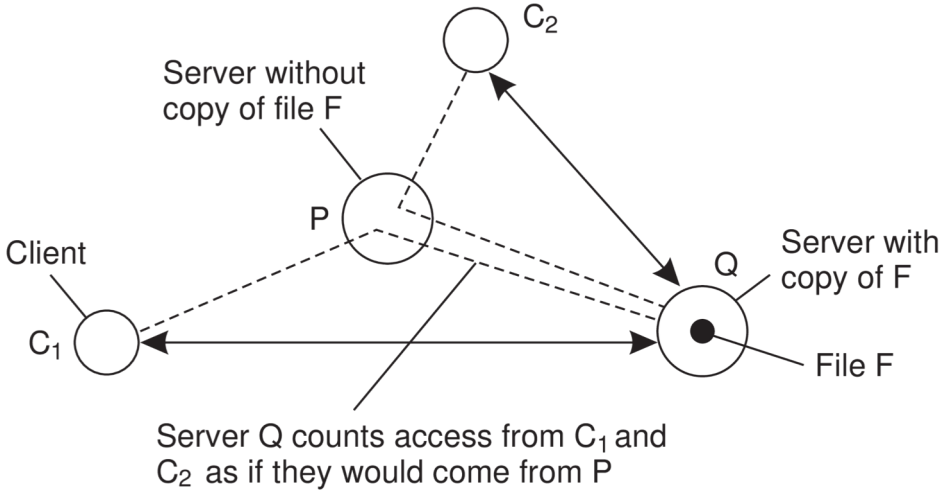
- Descobrir quais os melhores K locais de entre N possíveis localizações
  - Seleccionar a melhor localização de entre N – K para a qual a média da distancia aos clientes é mínima. De seguida escolher o melhor servidor. (caro)
  - Seleccionar o Kº maior Sistema Autonomo (AS) e colocar um servidor no computador melhor ligado. (caro)
  - Posicionar nós num espaço d-dimensional, onde a distancia reflete a latência. Identificar as K regiões com densidade mais elevada e colocar um servidor em cada uma. (barato)

Replicação de conteúdo

- Distinguir os diversos processos:
  - Replicas permanentes: Processo/máquina tem sempre uma réplica
  - Replicas iniciadas pelo servidor: Processo pode dinamicamente hospedar uma réplica a pedido do servidor.
  - Replicas iniciada pelo cliente: Processo pode dinamicamente hospedar uma réplica a pedido do cliente (client cache)



Replicas iniciadas pelo servidor



- Guardar numero de acessos por ficheiro, agregar por servidor
- Numero de acessos cai abaixo do limite D -> apagar ficheiro
- Numero de acessos acima do limite R -> replicar ficheiro
- Numero entre D e R -> migrar ficheiro