informática

Universidade Federal de Santa Maria

Modelos de Consistência e Replicação de Dados

Prof. Raul Ceretta Nunes

Curso de Ciência da Computação

ELC1018 - Sistemas Distribuídos

informática
Universidade Federal de Santa Maria

Introdução

- SD é adequado para replicação de dados
- Replicas devem ser mantidas consistentes
- Questões:
  - O que significa consistência de dados replicados?
  - Quais as maneiras para obter consistência?
  - Como o desempenho é afetado por diferentes modelos de consistência?
  - Como a consistência é implementada?
    - Forma de distribuição dos updates para as réplicas
    - Forma de manter a consistência (forte ou fraca)

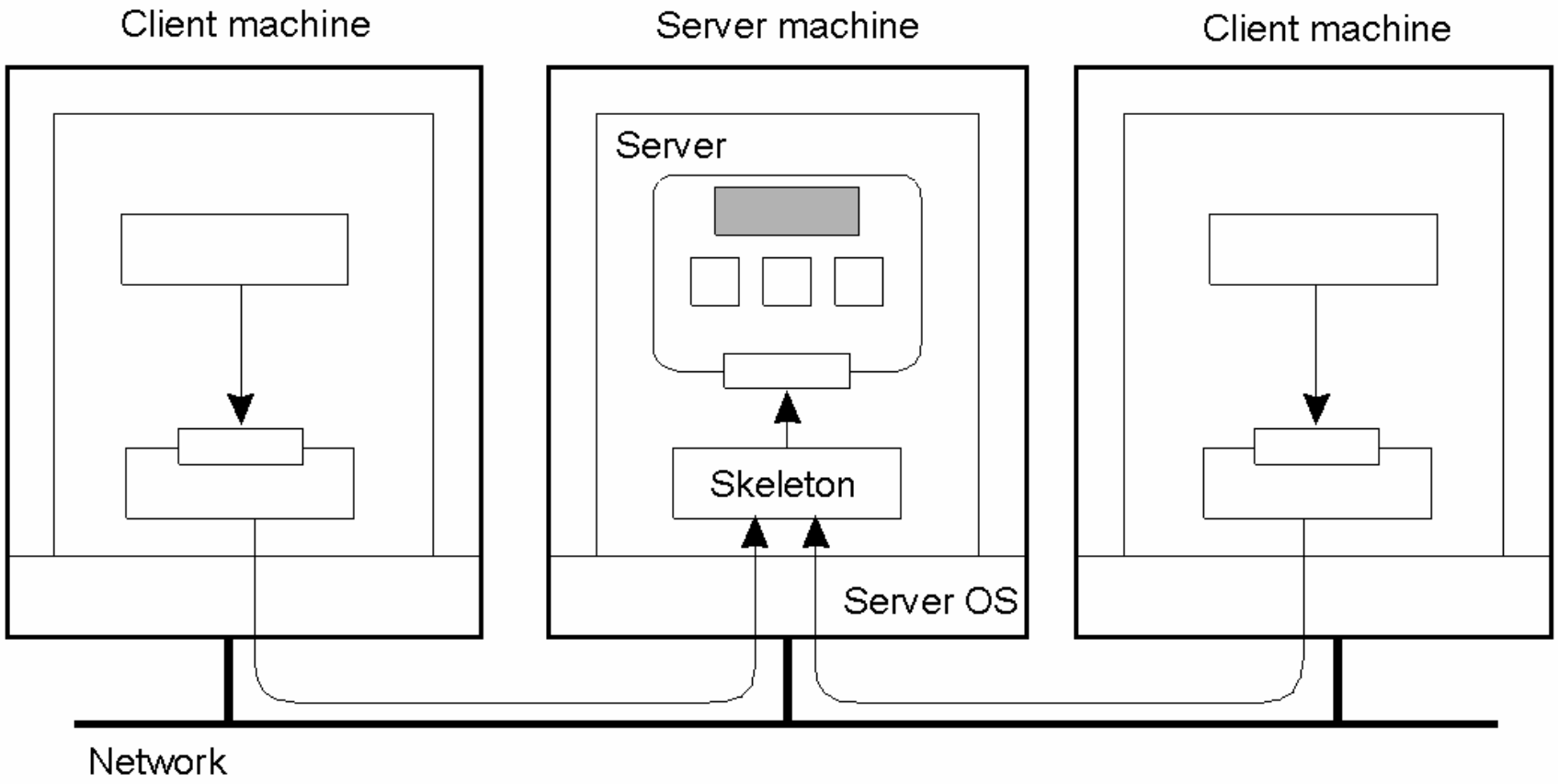
informática
Universidade Federal de Santa Maria

Razões para replicação

- Duas razões:
  - Confiabilidade
    - colapso de processo
    - dados inconsistentes (usar valor da maioria)
  - Desempenho
    - Escalabilidade através de serviços replicados
    - Redução de tráfego de rede através de réplicas mais próximas
- Preço a pagar: gerenciamento das réplicas para manter consistência

informática
Universidade Federal de Santa Maria

Objeto Remoto Compartilhado



- Organização composta por um objeto remoto distribuído que é compartilhado por dois clientes.

informática
Universidade Federal de Santa Maria

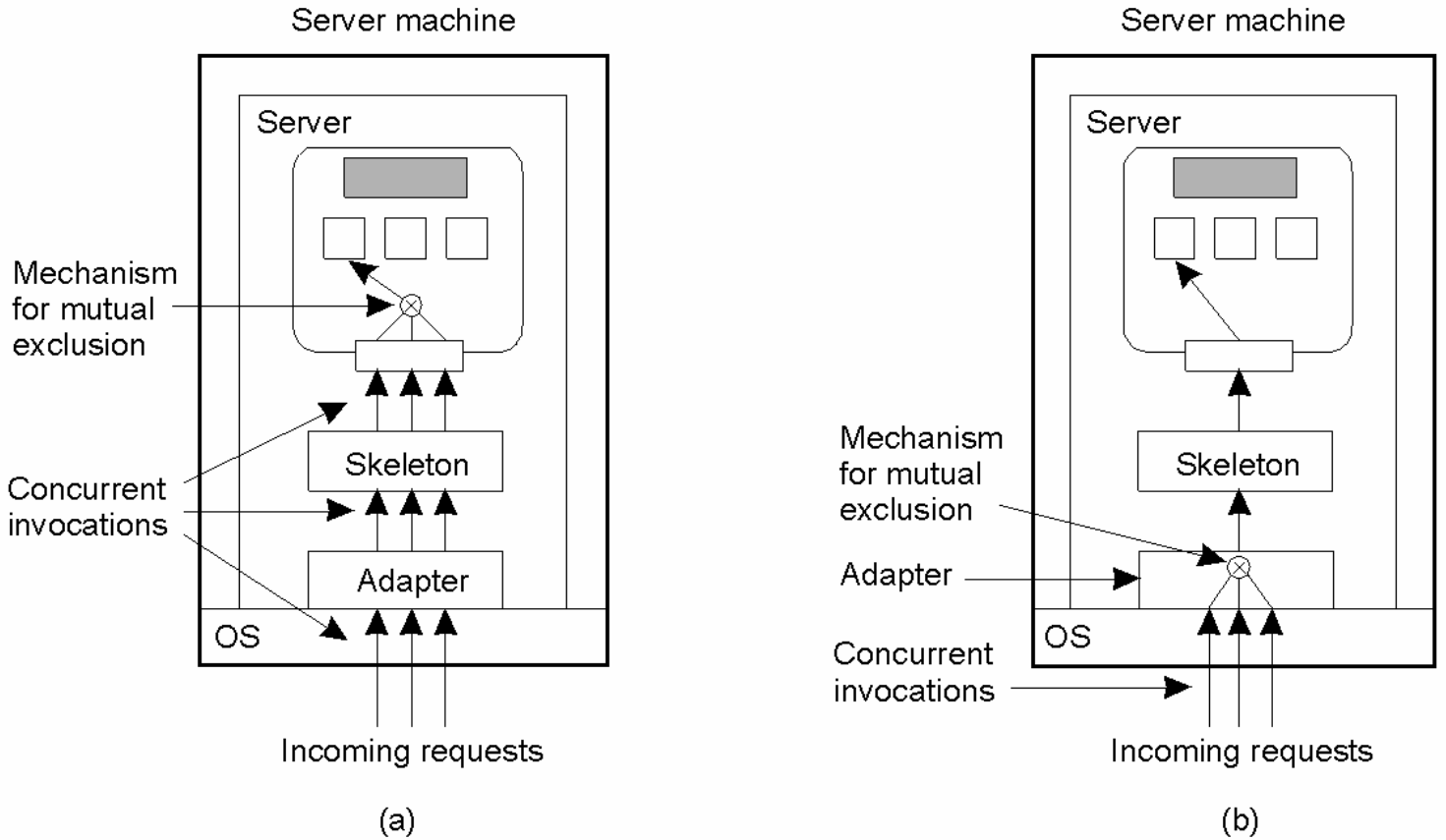
Objeto Remoto Compartilhado

- Objetos encapsulam dados e operações sobre dados
- Middleware gerencia distribuição de objetos remotos (operações são independentes dos dados)

- Como proteger o objeto remoto contra acessos simultâneos realizados por múltiplos clientes?
  - Controlar no próprio objeto (ex: `synchronized` do Java)
  - Controlar no middleware (ex: adaptador de objetos)

informática
Universidade Federal de Santa Maria

Objeto Remoto Compartilhado



a) Objeto remoto capaz de controlar invocações concorrentes.

b) Objeto remoto que requer um adaptador de objeto para controlar invocações concorrentes.

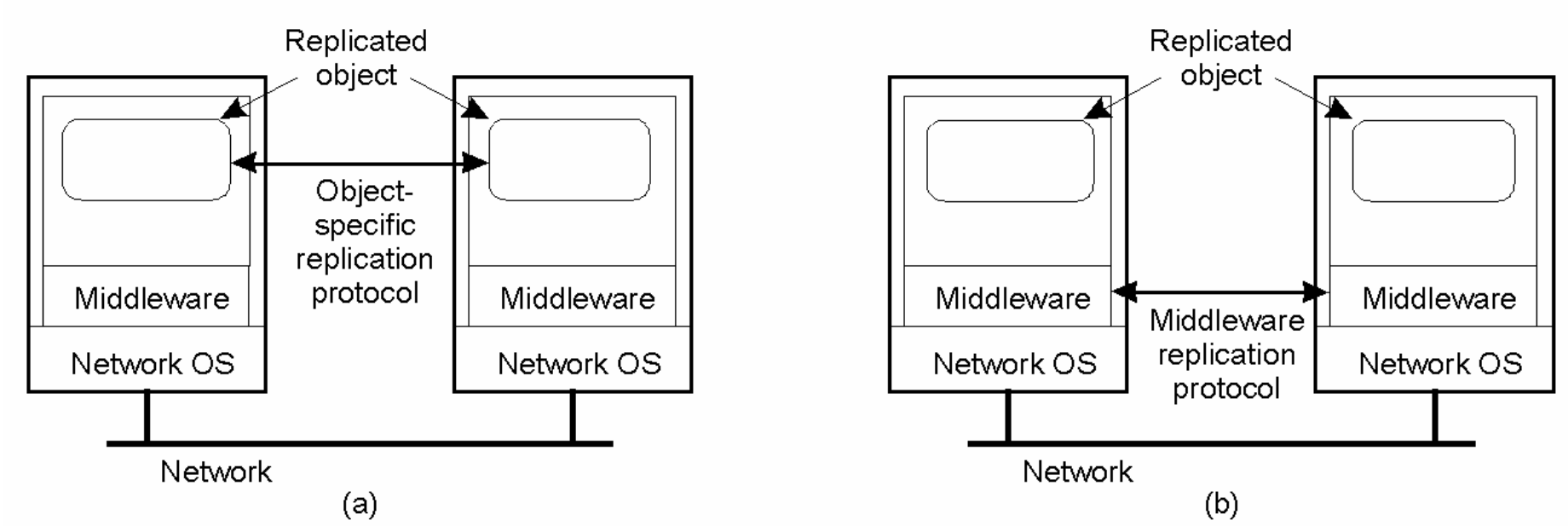
informática
Universidade Federal de Santa Maria

Replicação de Objeto Remoto

- Objeto remoto compartilhado e replicado necessita de sincronização adicional para garantir que invocações concorrentes sejam resolvidas na ordem correta em cada réplica.
- 2 abordagens:
  - Controlar sincronização no objeto replicado.
    - Vantagem: uso de protocolo de replicação específico.
  - Controlar sincronização no middleware
    - Vantagem: simplifica tarefa do desenvolvedor de aplicações

informática
Universidade Federal de Santa Maria

Replicação de Objeto Remoto



a) Um sistema distribuído para objetos cientes da replicação, que controlam sua replicação.

b) Um sistema distribuído responsável pelo gerenciamento de replicação.

informática
Universidade Federal de Santa Maria

Modelos de Consistência

Centrado nos dados
Hip: dados compartilhados p/ múltiplos processos

- **Forte** p/ *read*/*write*
  - Estrita - *write* p/ ordem total temporal
  - Seqüencial - *write* p/ ordem total
  - Linear - *write* p/ ordem total baseada em relógio global
  - Causal - *write* p/ ordem causal
  - FIFO - *write* p/ ordem causal
- **Fraca** p/ série de *read*/*write* via variáveis de sync
  - Fraca - *update* p/ operação
  - Release - *update* na liberação
  - Entry - *update* na entrada

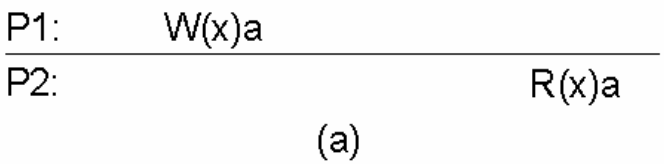
Centrado no cliente
Hip: dados atualizados p/ um único processo

- Leituras monotônicas - *reads* em ordem temporal
- Escritas monotônicas - *writes* em ordem temporal
- Leituras vêem escritas
- Escritas seguem leituras

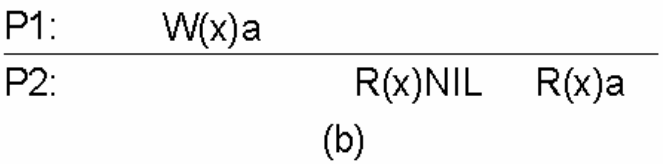
informática
Universidade Federal de Santa Maria

Consistência Estrita

- Operação tem efeito instantâneo em todos.
- Deve manter ordem temporal (relógio físico).
- Impossível de ser implementado.



(a)



(b)

Comportamento de dois processos operando sobre o mesmo dado:

(a) Um armazenamento com consistência estrita.

(b) Um armazenamento que viola a consistência estrita.

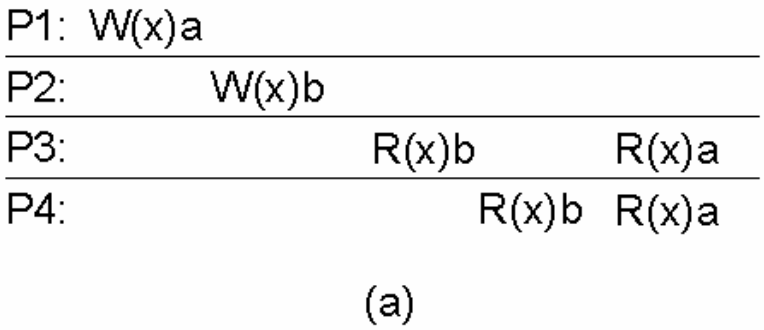
informática
Universidade Federal de Santa Maria

Consistência Seqüencial

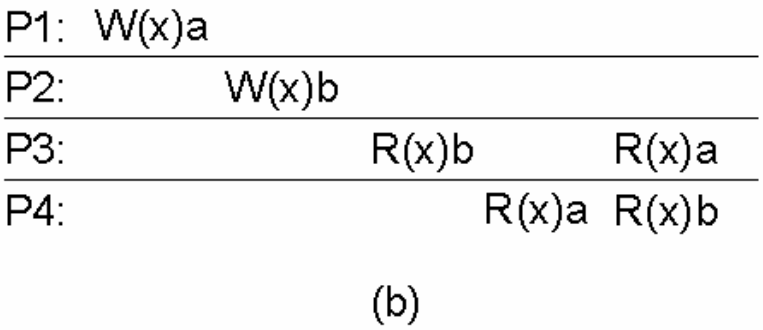
- Todos os processos observam as operações na mesma ordem (ordem total).
- Nada é dito sobre o instante temporal físico.
- Aplicado em memória compartilhada de sistemas multiprocessados.
- Equivalente a serializabilidade transacional (operação equivale a uma transação).
- A verificação de consistência se dá através da análise de históricos locais.
- Problema: $t_{read} + t_{write} \geq t_{transferência}$

informática
Universidade Federal de Santa Maria

Consistência Seqüencial



(a)



(b)

(a) Um armazenamento com consistência seqüencial.
(b) Um armazenamento com violação de consistência seqüencial.

informática
Universidade Federal de Santa Maria

Ex: Consistência Seqüencial

- Três processos executando concorrentemente e 4 execuções possíveis.

Process P1	Process P2	Process P3
x = 1;	y = 1;	z = 1;
print (y, z);	print (x, z);	print (x, y);

x = 1;	x = 1;	y = 1;	y = 1;
print ((y, z);	y = 1;	z = 1;	x = 1;
y = 1;	print (x,z);	print (x, y);	z = 1;
print (x, z);	print(y, z);	print (x, z);	print (x, z);
z = 1;	z = 1;	x = 1;	print (y, z);
print (x, y);	print (x, y);	print (y, z);	print (x, y);
Prints: 001011	Prints: 101011	Prints: 010111	Prints: 111111
Signature: 001011	Signature: 101011	Signature: 110101	Signature: 111111

informática
Universidade Federal de Santa Maria

Linearizabilidade

- Mais fraco do que a consistência estrita mas mais forte do que a consistência seqüencial.
- Todos os processos observam as operações na mesma ordem (ordem total) + timestamp.
- Se t_s $op_1(x) < t_s$ $op_2(x)$ então $op_1(x) \rightarrow op_2(x)$
- Utilizada para assistir verificação formal de algoritmos concorrentes.

informática
Universidade Federal de Santa Maria

Consistência Causal

- Writes potencialmente relacionados devem ser vistos por todos na mesma ordem.
- Writes concorrentes podem ser vistos em ordens diferentes.
- Implementação via vetor de relógios lógicos.

informática
Universidade Federal de Santa Maria

Consistência FIFO

- Observa apenas a ordem das operações no processo da operação.
- Fácil de implementar, pois precisa apenas da inclusão de um número de seqüência.
- Em memória compartilhada distribuída é chamado de PRAM –Pipelined RAM.

informática
Universidade Federal de Santa Maria

Ex: Consistência FIFO

```
x = 1;
print (y, z);
y = 1;
print(x, z);
z = 1;
print (x, y);
```

Prints: 00

(a)

```
x = 1;
y = 1;
print(x, z);
print ( y, z);
z = 1;
print (x, y);
```

Prints: 10

(b)

```
y = 1;
print (x, z);
z = 1;
print (x, y);
x = 1;
print (y, z);
```

Prints: 01

(c)

informática
Universidade Federal de Santa Maria

Problema da Consistência FIFO

- Fraco a ponto de permitir a morte de ambos os processos concorrentes, o que não é permitido com consistência seqüencial.

Process P1	Process P2
x = 1; if (y == 0) kill(P2);	y = 1; if (x == 0) kill(P1);

informática
Universidade Federal de Santa Maria

Consistência Fraca

- Synchronization accesses are sequentially consistent. All synchronization accesses must be performed before a regular data access and vice versa (programmer-imposed consistency).

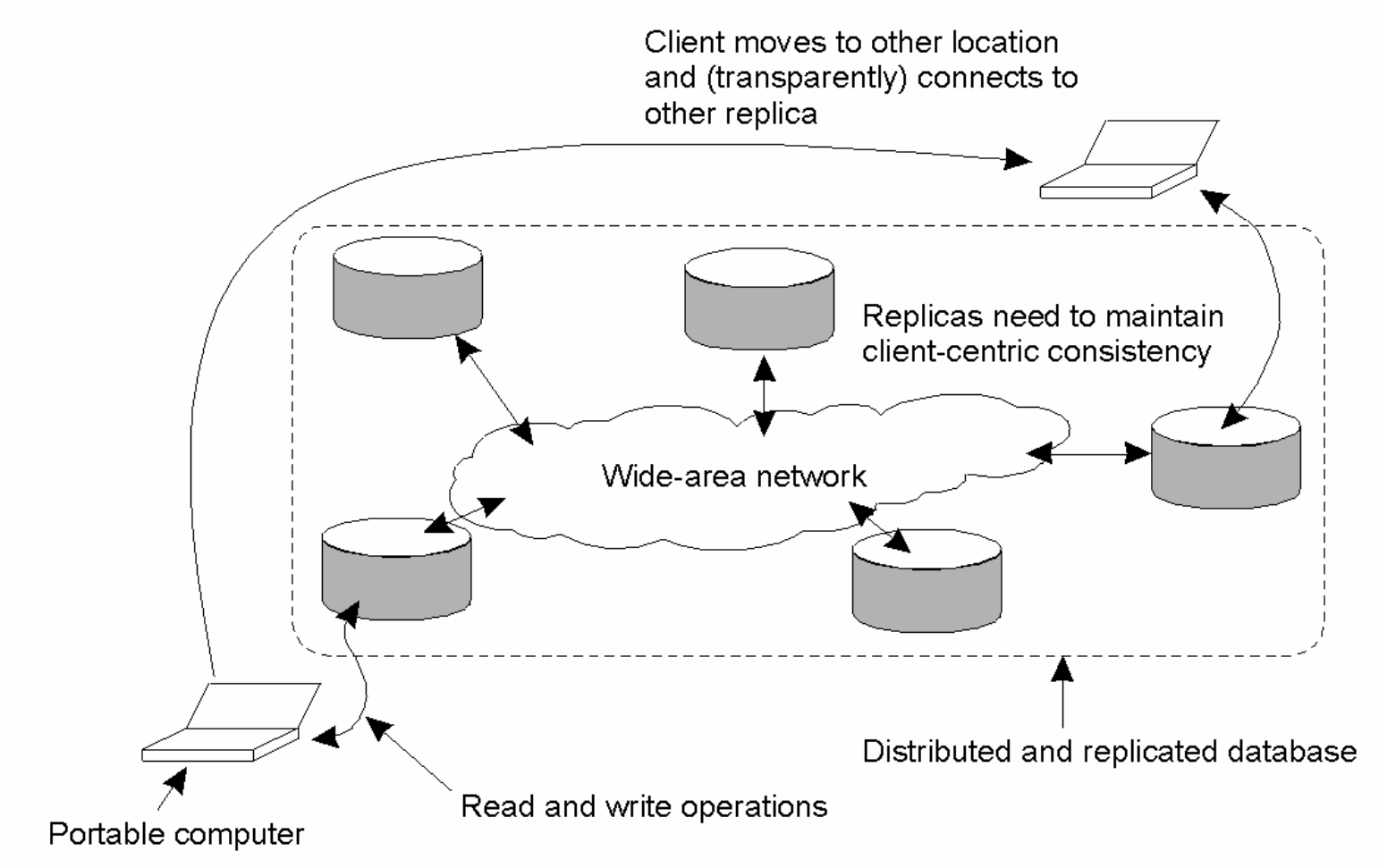
informática
Universidade Federal de Santa Maria

Consistência Fraca na Saída

- Same as weak consistency except that synchronization accesses must only be processor consistent with respect to each other.

informática
Universidade Federal de Santa Maria

Eventual Consistency

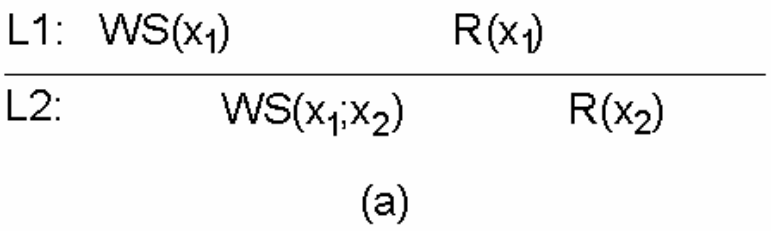


informática
Universidade Federal de Santa Maria

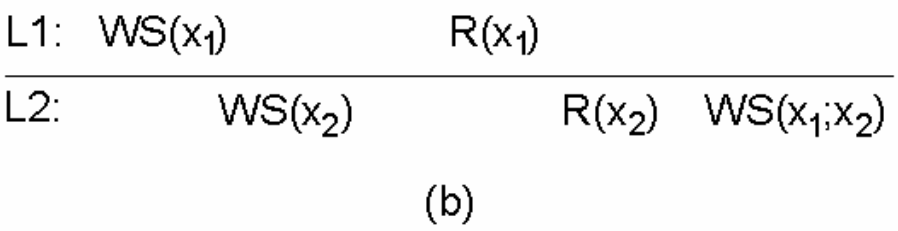
Monotonic Reads

- Se o processo lê o valor de um item de dado x, qualquer operação de leitura sucessiva de x pelo processo irá retornar sempre o mesmo valor ou um valor mais recente.

- The read operations performed by a single process *P* at two different local copies of the same data store.



(a)



(b)

a) A monotonic-read consistent data store

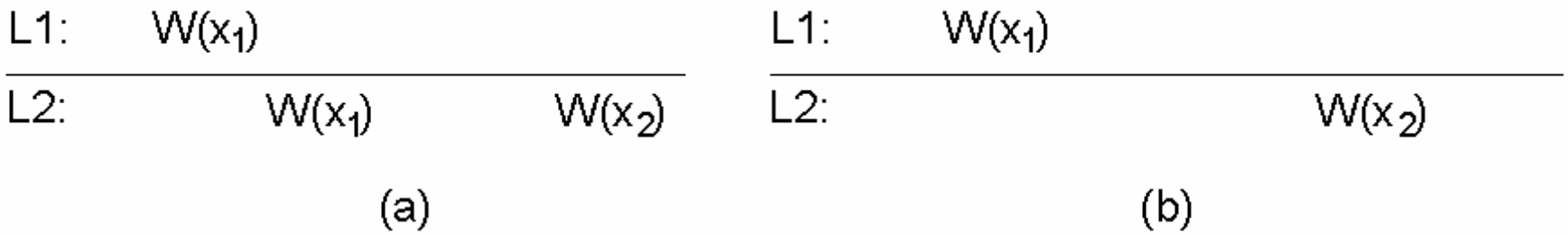
b) A data store that does not provide monotonic reads.

informática
Universidade Federal de Santa Maria

Monotonic Writes

- Uma operação de escrita por um processo sobre um item de dado x é completada antes de qualquer operação de escrita sucessiva do mesmo processo.

- The write operations performed by a single process *P* at two different local copies of the same data store



a) A monotonic-write consistent data store.
b) A data store that does not provide monotonic-write consistency.

informática
Universidade Federal de Santa Maria

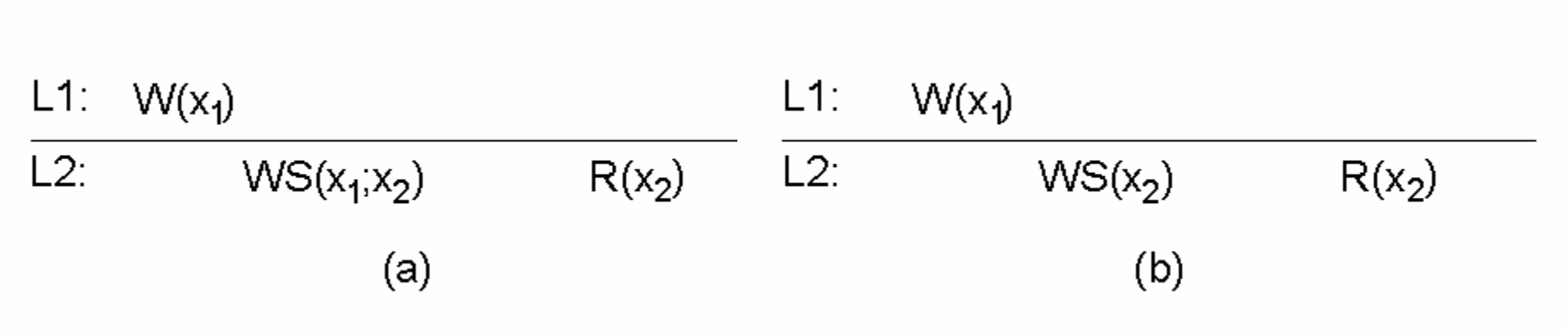
Monotonic Writes

- Relação com o modelo FIFO:
  - Em ambos a ordem das operações nos processos é mantida em qualquer local.
  - No modelo FIFO são considerados vários processos concorrentes.
  - No modelo Monotonic Write é considerado apenas um único processo (cliente).

informática
Universidade Federal de Santa Maria

Read Your Writes

- O efeito de uma operação de escrita por um processo sobre um item de dado x sempre será visto pela operação de leitura de x pelo mesmo processo.



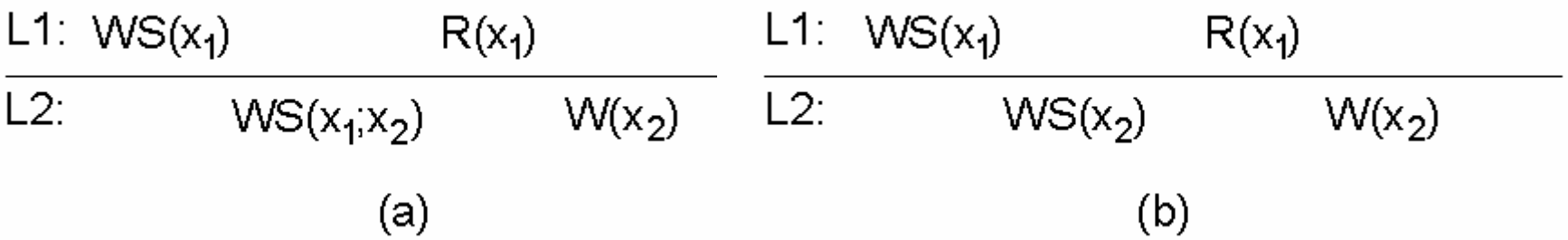
a) A data store that provides read-your-writes consistency.

b) A data store that does not.

informática
Universidade Federal de Santa Maria

Writes Follow Reads

- Uma operação de escrita por um processo sobre um item de dado x que sucede uma operação de leitura de x pelo mesmo processo, será resolvida sobre uma cópia de x que contém o valor lido ou um mais recente.



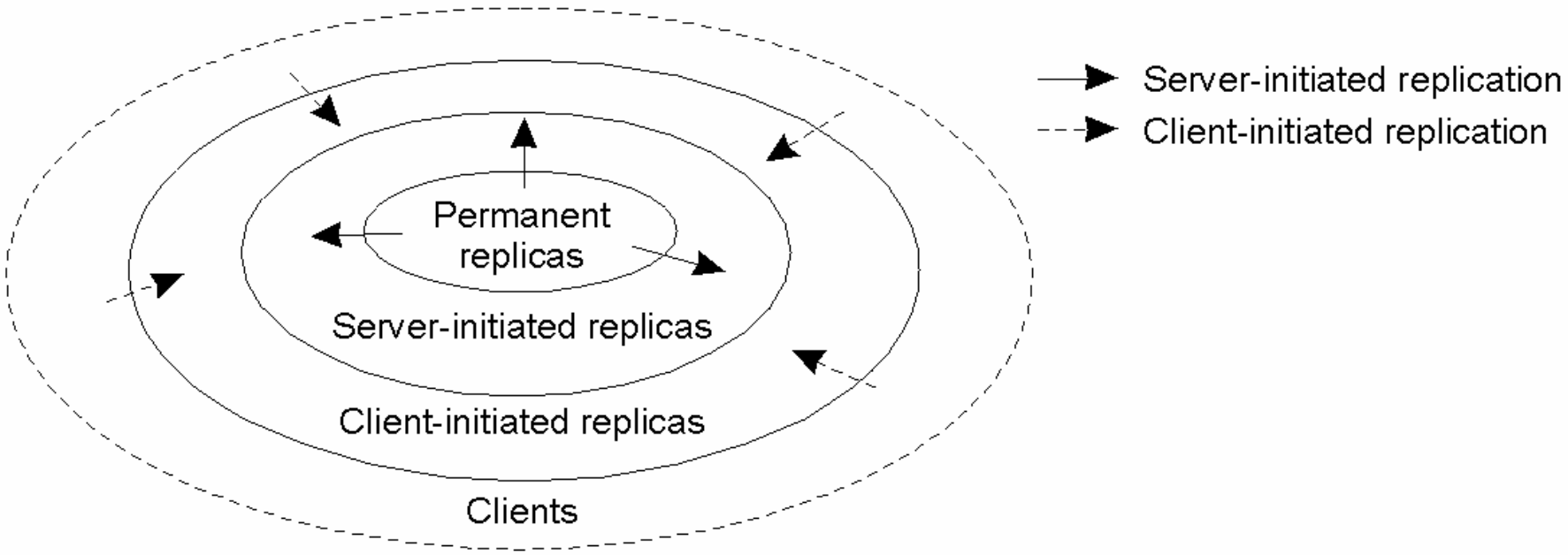
a) A writes-follow-reads consistent data store

b) A data store that does not provide writes-follow-reads consistency

informática
Universidade Federal de Santa Maria

Posicionamento das Réplicas

- Onde, quando e por quem os dados estão distribuídos?



A organização lógica dos diferentes conjuntos de cópias de um dado armazenado dentro de três anéis concentricos.

informática
Universidade Federal de Santa Maria

Réplicas Permanentes

- Normalmente em pequeno número.
- Modelo de replicação estática
- Exemplos:
  - Web server: replicação do servidor numa lan; ao chegar uma requisição, ela é direcionada a uma das réplicas.
  - Site web distribuído (Mirror): site é distribuído geograficamente; clientes escolhem um site de uma lista.
  - Banco de dados federados: BD distribuído e replicado.

informática
Universidade Federal de Santa Maria

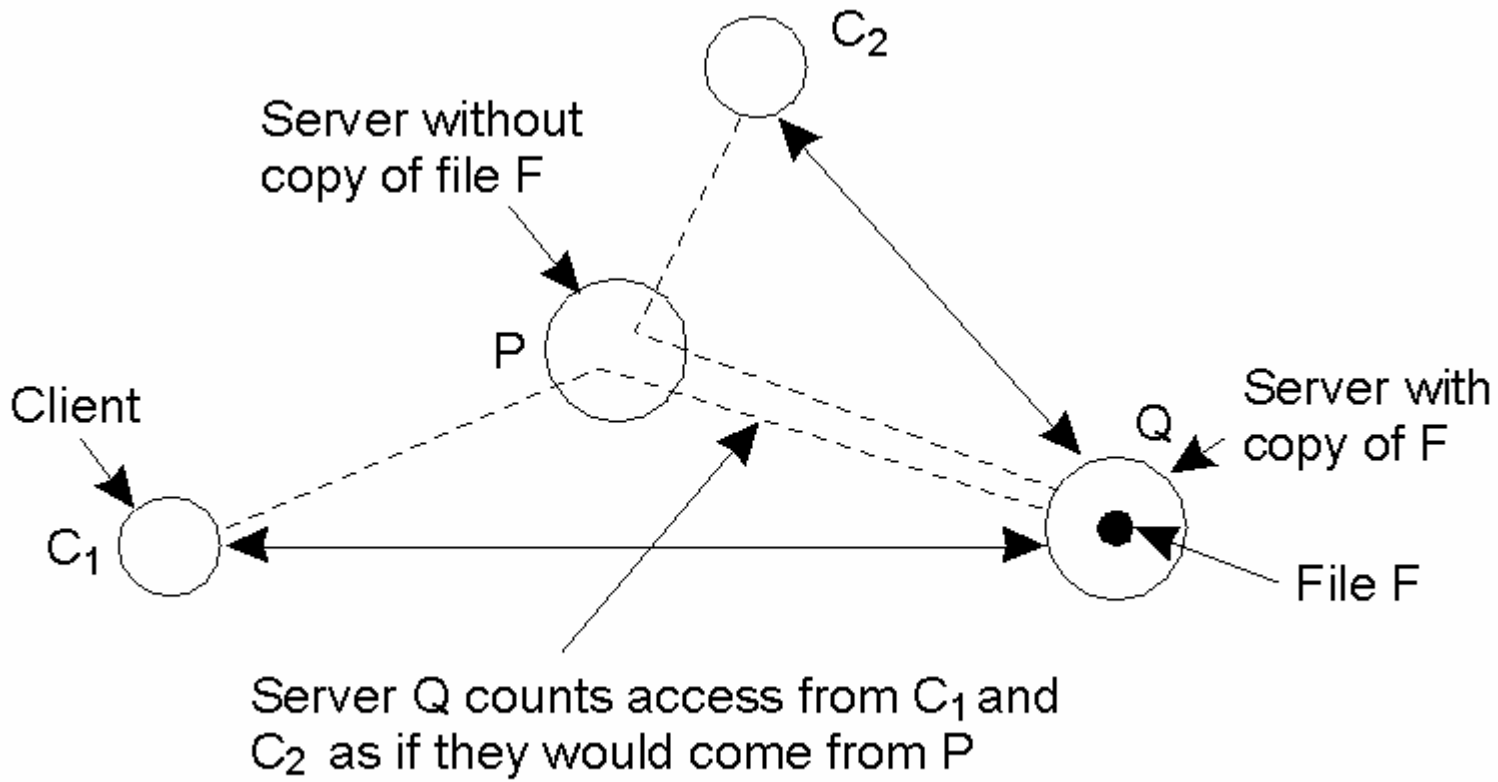
Réplicas Iniciadas pelo Servidor

- Modelo de replicação dinâmica.
- Réplicas são criadas de acordo com a necessidade. Ex: para melhorar desempenho
- No caso de servidor web, são também conhecidas por “push caches”.
- Problema: decidir onde e quando réplicas deveriam ser criadas ou deletadas.

informática
Universidade Federal de Santa Maria

Réplicas Iniciadas pelo Servidor

- Contando acesso de diferentes clientes.
- Decisão baseada em threshold `del(S,F)| rep(S,F)`



informática
Universidade Federal de Santa Maria

Réplicas Iniciadas pelo Cliente

- A réplica é uma cache.
- A consistência da réplica é por conta do usr.
- Usado somente para melhorar desempenho.
- Na web, uma réplica pode ser compartilhada por mais do que um cliente.

informática
Universidade Federal de Santa Maria

Propagação de Atualizações

- Abordagens para decidir o que propagar:
  - Propagar somente uma notificação de um update. Muito update pouco read.
    - Ex: protocolos de invalidação
  - Transferir os dados de uma cópia para outra. Interessante quando há muitos updates e reads.
  - Propagar a operação de atualização para outras réplicas.
    - Replicação ativa

informática
Universidade Federal de Santa Maria

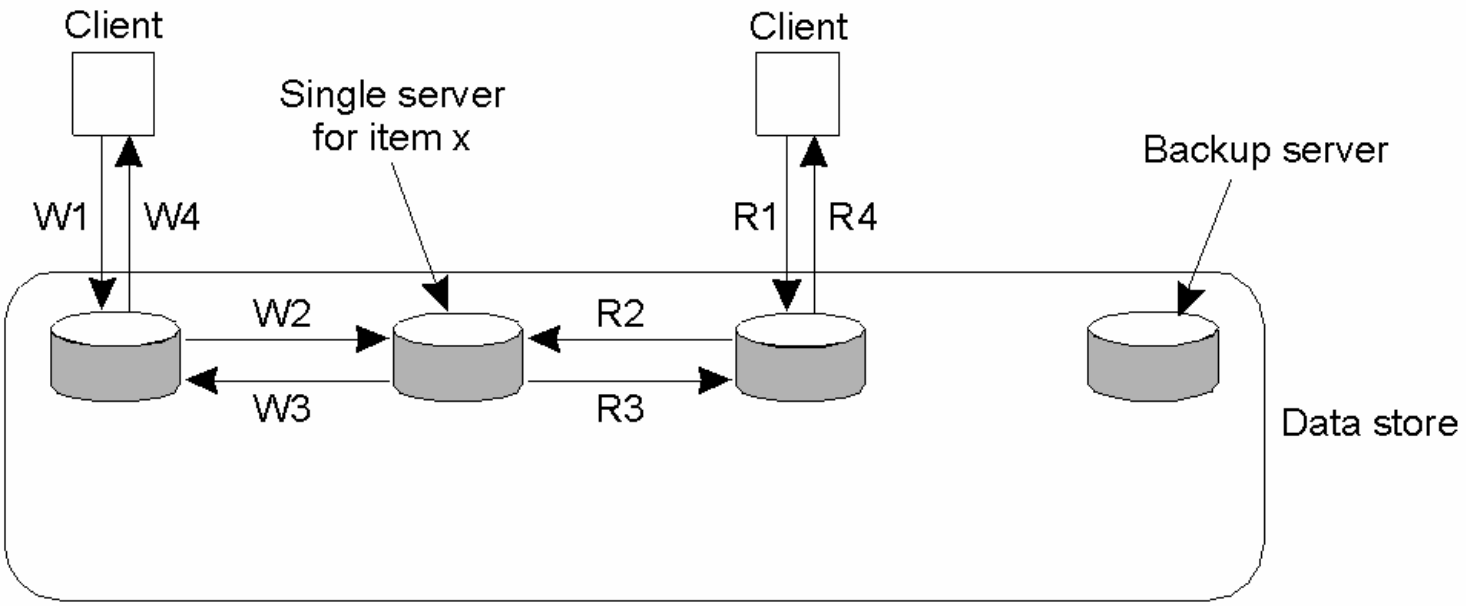
Pull versus Push Protocols

Issue	Push-based	Pull-based
State of server	List of client replicas and caches	None
Messages sent	Update (and possibly fetch update later)	Poll and update
Response time at client	Immediate (or fetch-update time)	Fetch-update time

- A comparison between push-based and pull-based protocols in the case of multiple client, single server systems.

informática
Universidade Federal de Santa Maria

Remote-Write Protocols (1)



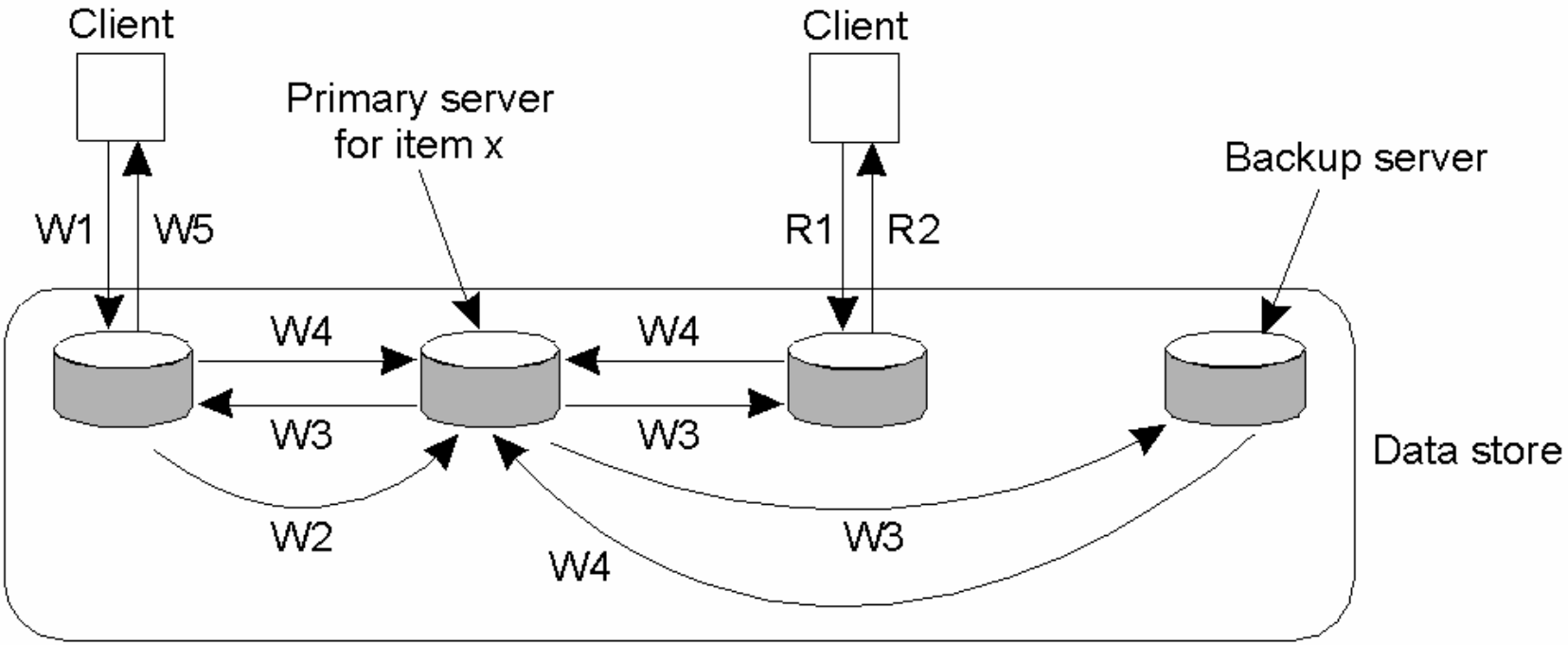
W1. Write request
W2. Forward request to server for x
W3. Acknowledge write completed
W4. Acknowledge write completed

R1. Read request
R2. Forward request to server for x
R3. Return response
R4. Return response

- Primary-based remote-write protocol with a fixed server to which all read and write operations are forwarded.

informática
Universidade Federal de Santa Maria

Remote-Write Protocols (2)



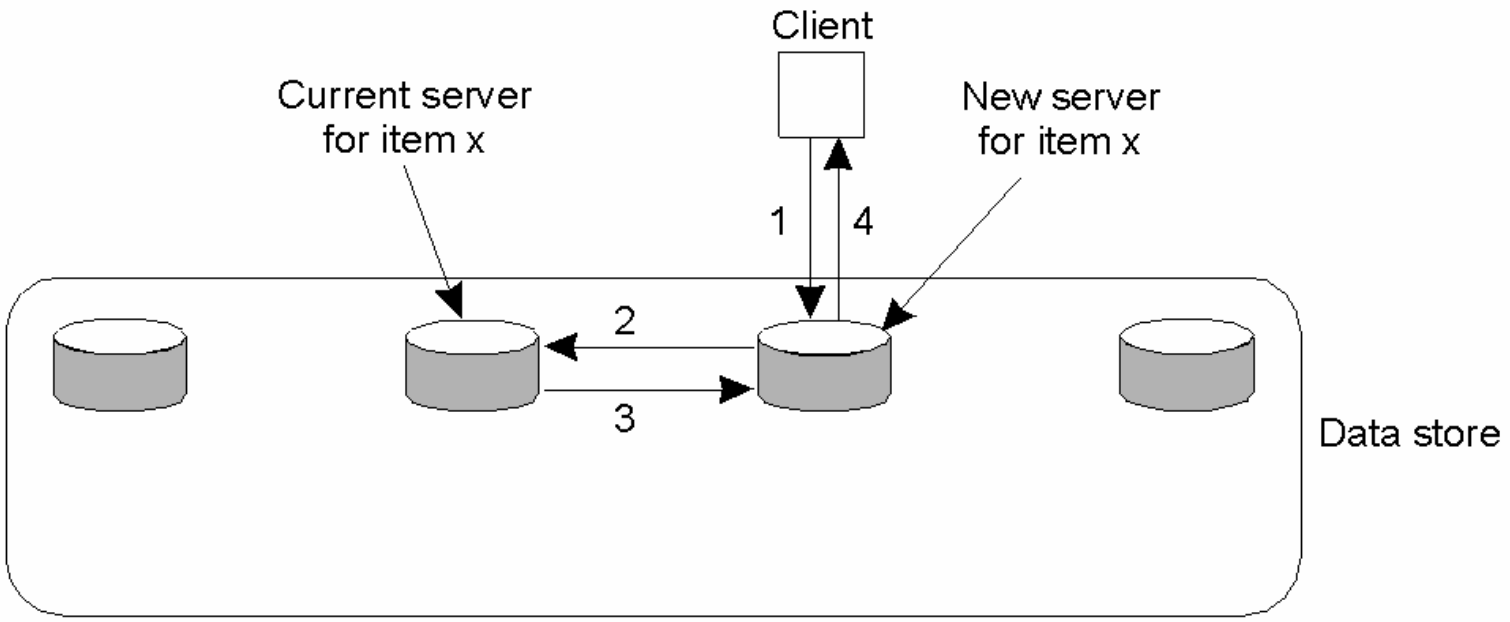
W1. Write request
W2. Forward request to primary
W3. Tell backups to update
W4. Acknowledge update
W5. Acknowledge write completed

R1. Read request
R2. Response to read

O princípio do protocolo primário-backup.

informática
Universidade Federal de Santa Maria

Local-Write Protocols (1)

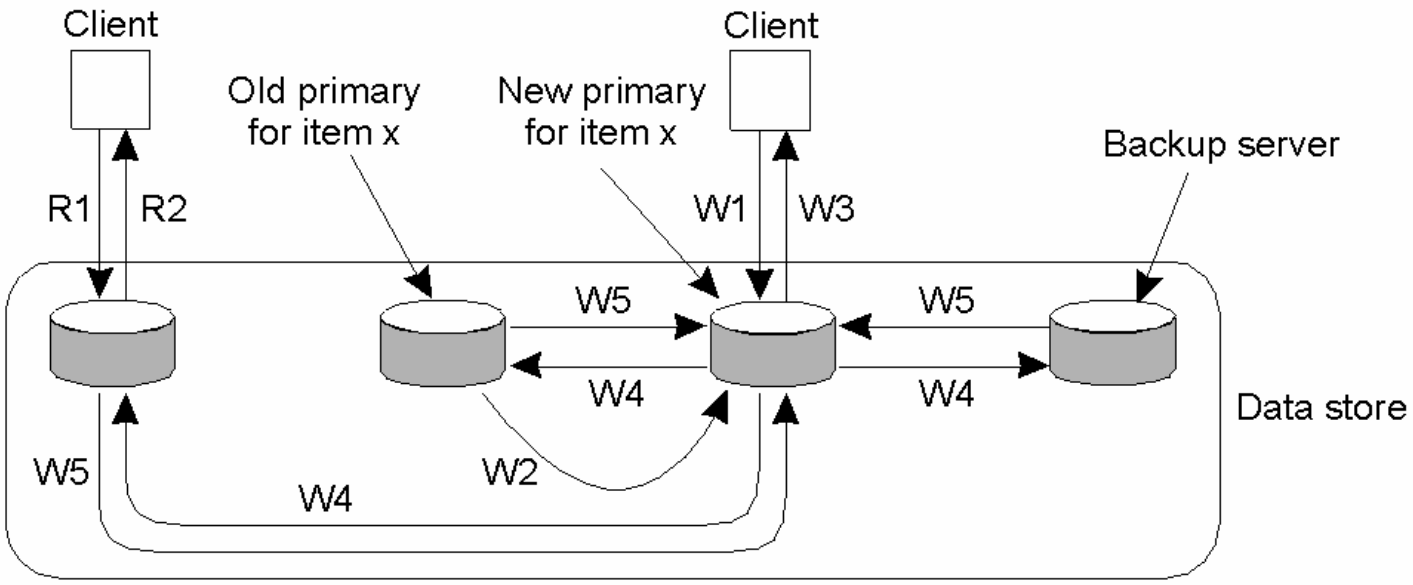


1. Read or write request
2. Forward request to current server for x
3. Move item x to client's server
4. Return result of operation on client's server

Primary-based local-write protocol in which a single copy is migrated between processes.

informática
Universidade Federal de Santa Maria

Local-Write Protocols (2)



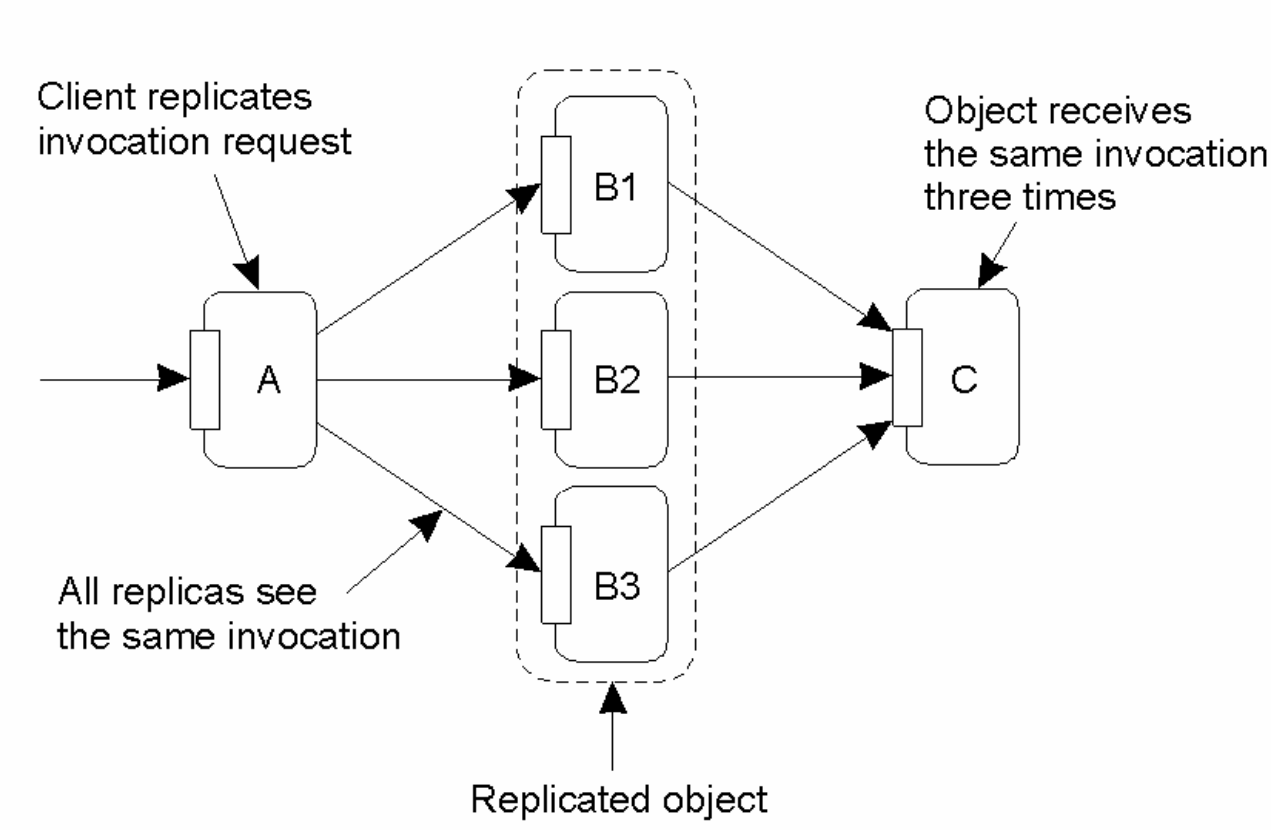
W1. Write request
W2. Move item x to new primary
W3. Acknowledge write completed
W4. Tell backups to update
W5. Acknowledge update

R1. Read request
R2. Response to read

- Primary-backup protocol in which the primary migrates to the process wanting to perform an update.

informática
Universidade Federal de Santa Maria

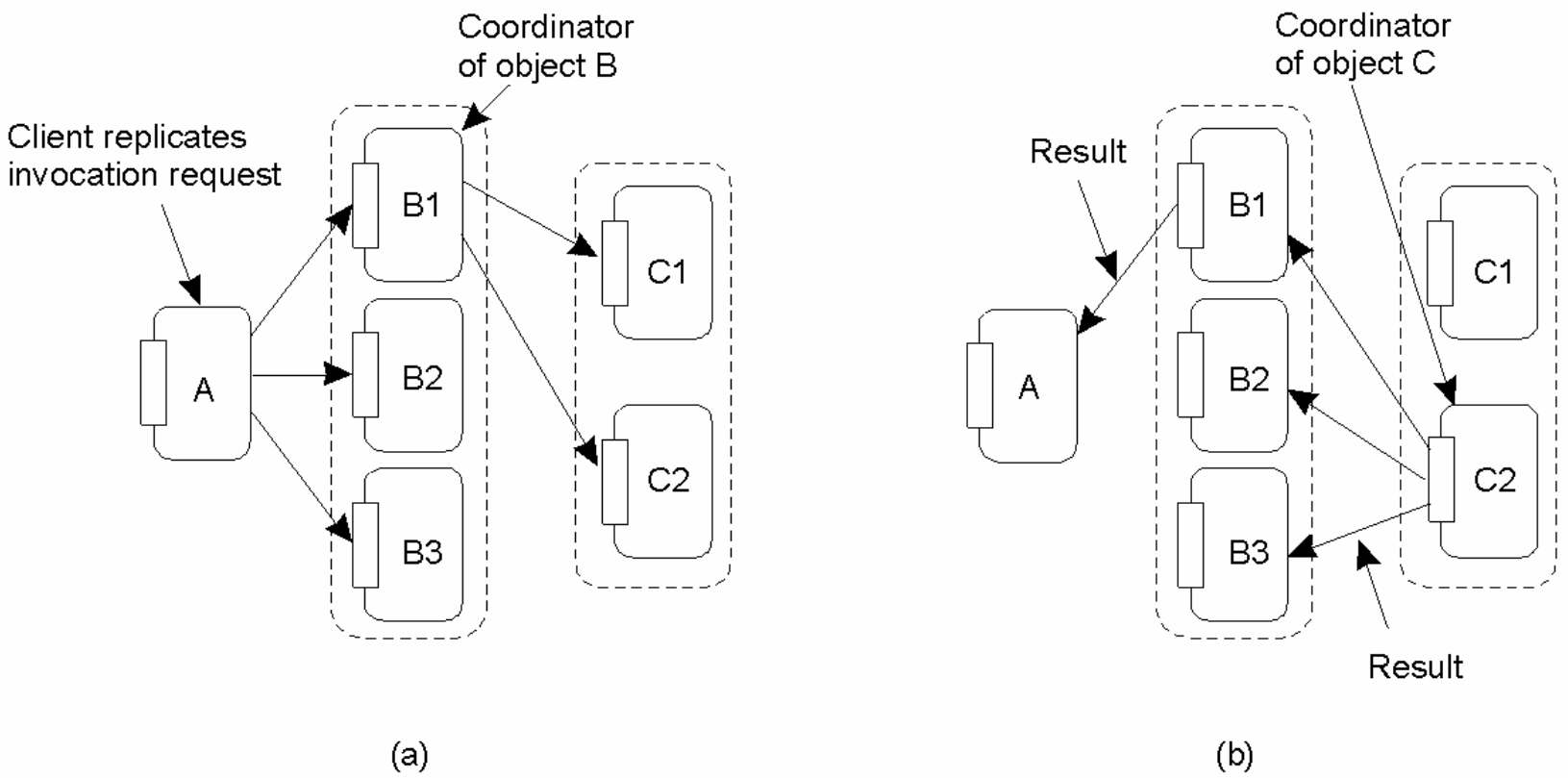
Active Replication (1)



- The problem of replicated invocations.

informática
Universidade Federal de Santa Maria

Active Replication (2)

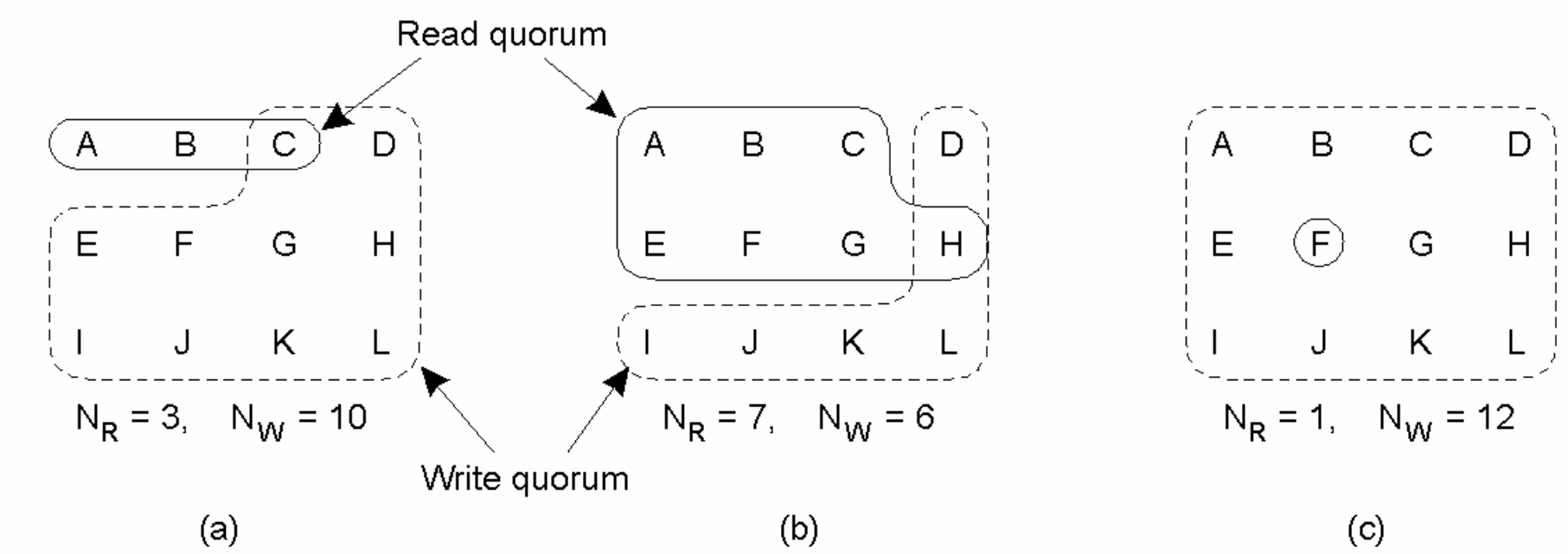


a) Forwarding an invocation request from a replicated object.

b) Returning a reply to a replicated object.

informática
Universidade Federal de Santa Maria

Quorum-Based Protocols



- Three examples of the voting algorithm:

a) A correct choice of read and write set

b) A choice that may lead to write-write conflicts

c) A correct choice, known as ROWA (read one, write all)