



# BCC 362 – Sistemas Distribuídos

Joubert de Castro Lima – [joubertlima@gmail.com](mailto:joubertlima@gmail.com)  
Professor Adjunto – DECOM

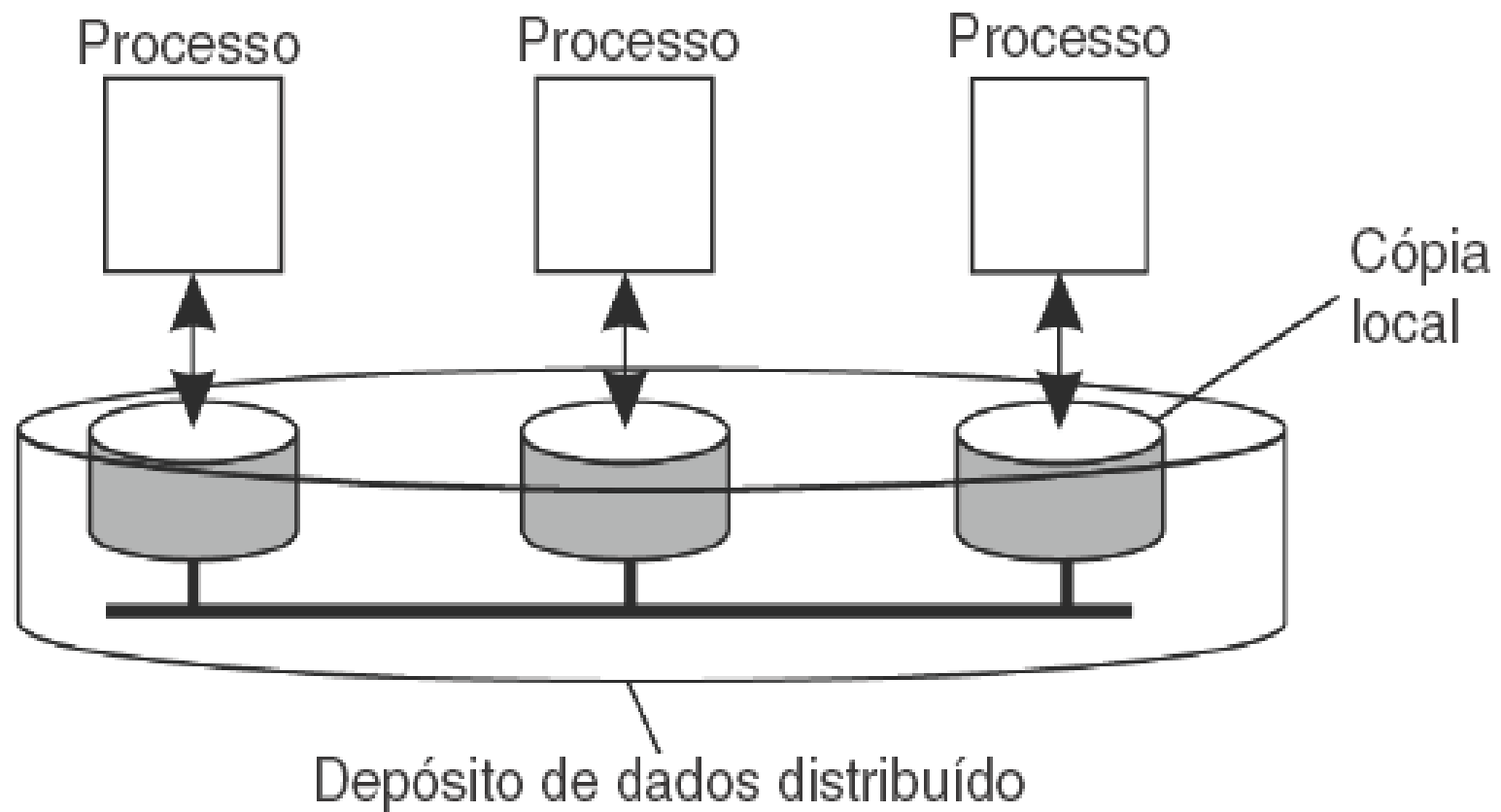
**UFOP**

Figuras e textos retirados do livro:  
Sistemas Distribuídos do Tanenbaum

# Reasons for replication

- (i) Reliability: If a file system has been replicated it may be possible to continue working after one replica crashes by simply switching to one the other replicas.
- (ii) Performance: It is important when the distributed system needs to scale in numbers and geographical area.

**Problems:** Having multiple copies may lead to consistency problems. Whenever a copy is modified, that copy becomes different from the rest.

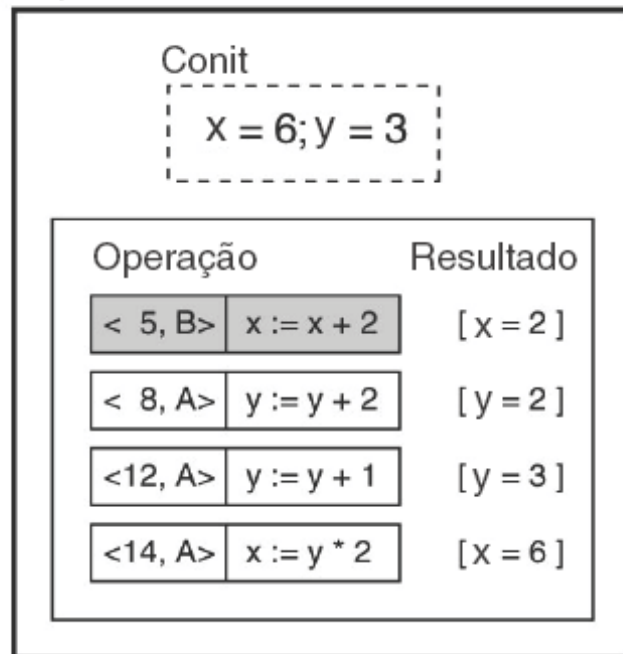


**Figura 7.1** Organização geral de um depósito de dados lógico, fisicamente distribuído e replicado por vários processos.

# Modelo de Consistência centrado nos Dados

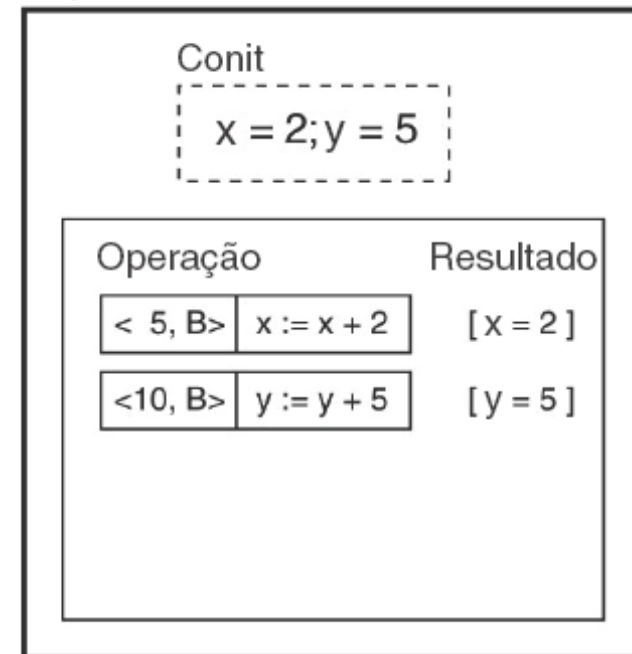
## Conit: Consistency Unit

Réplica A



Relógio vetorial A = (15,5)  
Desvio de ordenação = 3  
Desvio numérico = (1,5)

Réplica B



Relógio vetorial B = (0,11)  
Desvio de ordenação = 2  
Desvio numérico = (3,6)

**Figura 7.2** Exemplo de monitoração de desvios de consistência (adaptado de Yu e Vahdat, 2002).

# Consistência Sequencial

**Definição:** o resultado de qualquer execução é o mesmo que seria se as operações (leitura e escrita) realizadas por todos os processos no depósito de dados fossem executadas na mesma ordem sequencial e as operações de cada processo individual aparecessem nesta sequencia na ordem especificada por seu programa

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

(a)

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

(b)

Figura 7.5 (a) Depósito de dados seqüencialmente consistente. (b) Depósito de dados que não é seqüencialmente consistente.

# Consistência Causal

Definição: escritas que são potencialmente relacionadas por causalidade devem ser vistas por todos os processos na mesma ordem. Escritas concorrentes (sem dependência) podem ser vistas em ordem diferente em máquinas diferentes.

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

**Figura 7.8** Essa seqüência é permitida quando o depósito é consistente por causalidade, mas não quando o depósito é seqüencialmente consistente.

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

(a)

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

(b)

**Figura 7.9** (a) Violação de um depósito consistente por causalidade. (b) Seqüência correta de eventos em um depósito consistente por causalidade.



# Consistência FIFO

**Definição:** Gravações feitas por um único processo são vistas por todos os outros processos na ordem em que foram emitidos, mas escritas a partir de diferentes processos podem ser vistas em uma ordem diferente por diferentes processos

P1:  $\forall x (W(x) \rightarrow a)$

P2:            R(x)a      W(x)b      W(x)c

P3:  $R(x)b \quad R(x)a \quad R(x)c$

P4:  $R(x)a \quad R(x)b \quad R(x)c$

# Consistência de Entrada

**Consistency conditions:** *At an acquire, all remote changes to the guarded data must be made visible*

*Before updating a shared data item, a process must enter a critical region in exclusive mode to make sure that no other process is trying to update it at the same time.*

*If a process wants to enter a critical region in nonexclusive mode, it must first check with the owner of the synchronization variable guarding the critical region to fetch the most copies of the guarded shared data.*

P1:	Acq(Lx)	W(x) <sub>a</sub>	Acq(Ly)	W(y) <sub>b</sub>	Rel(Lx)	Rel(Ly)
P2:			Acq(Lx)	R(x) <sub>a</sub>		R(y) NIL
P3:				Acq(Ly)	R(y) <sub>b</sub>	

Figura 7.10 Sequência válida de eventos para consistência de entrada.

# Summary of Consistency Models

Consistency	Description
Strict	Absolute time ordering of all shared accesses matters.
Linearizability	All processes must see all shared accesses in the same order. Accesses are furthermore ordered according to a (nonunique) global timestamp
Sequential	All processes see all shared accesses in the same order. Accesses are not ordered in time
Causal	All processes see causally-related shared accesses in the same order.
FIFO	All processes see writes from each other in the order they were used. Writes from different processes may not always be seen in that order

(a)

Consistency	Description
Weak	Shared data can be counted on to be consistent only after a synchronization is done
Release	Shared data are made consistent when a critical region is exited
Entry	Shared data pertaining to a critical region are made consistent when a critical region is entered.

(b)

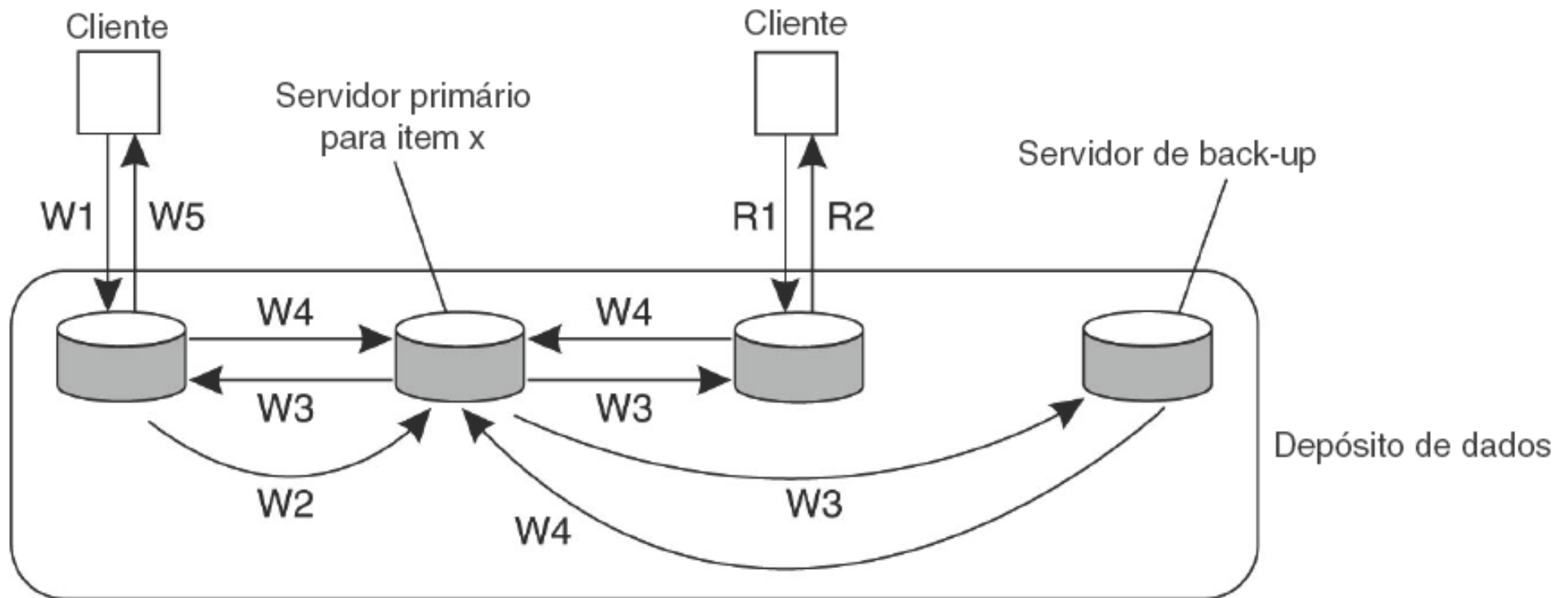
(a) Sem sincronização

(b) Com sincronização

# **CONSISTENCY PROTOCOLS**

## **PARTE 1**

# PROTOCOLO DE ESCRITA REMOTA

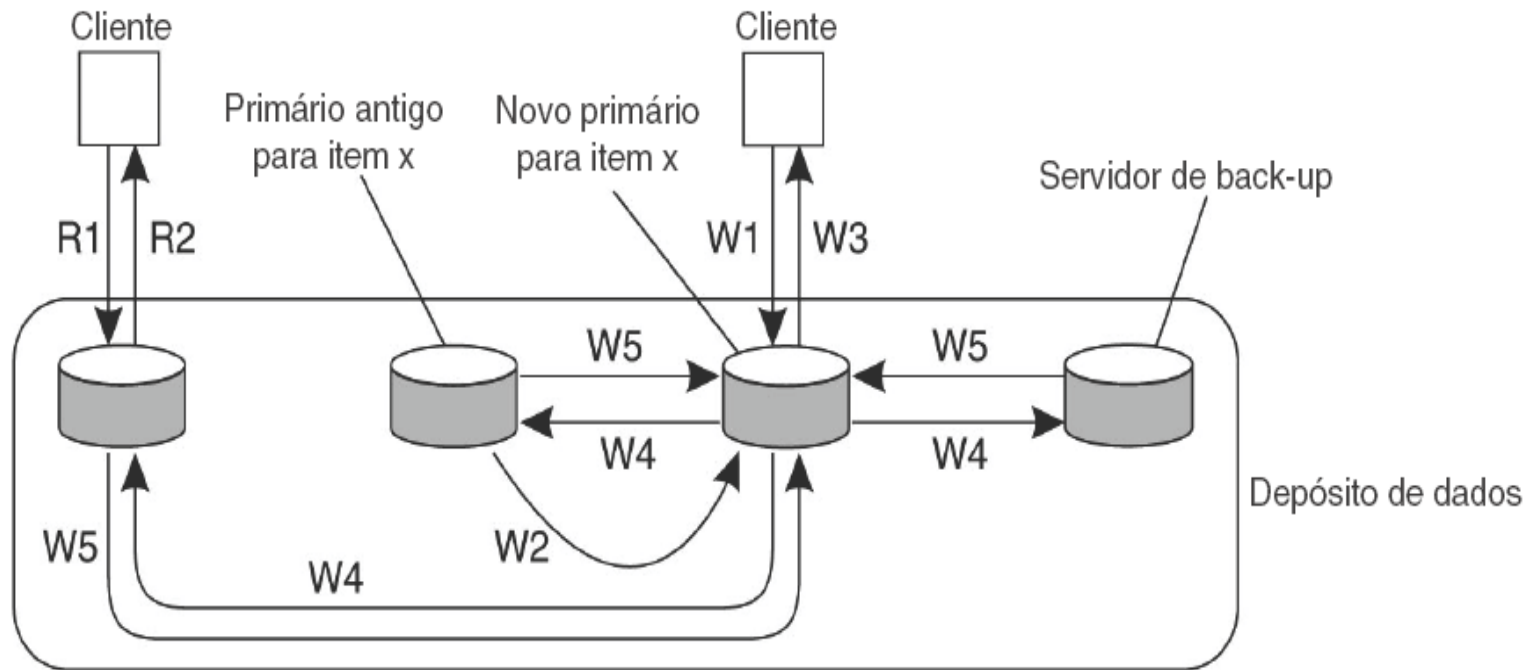


W1. Requisição de escrita  
W2. Repassa requisição ao primário  
W3. Diz aos back-ups para atualizar  
W4. Reconhece atualização  
W5. Reconhece escrita concluída

R1. Requisição de leitura  
R2. Resposta à leitura

Figura 7.19 Princípio de um protocolo de primário e backup.

# PROTOCOLO DE ESCRITA LOCAL



W1. Requisição de escrita

W2. Move item x para novo primário

W3. Reconhece escrita concluída

W4. Diz aos back-ups para atualizar

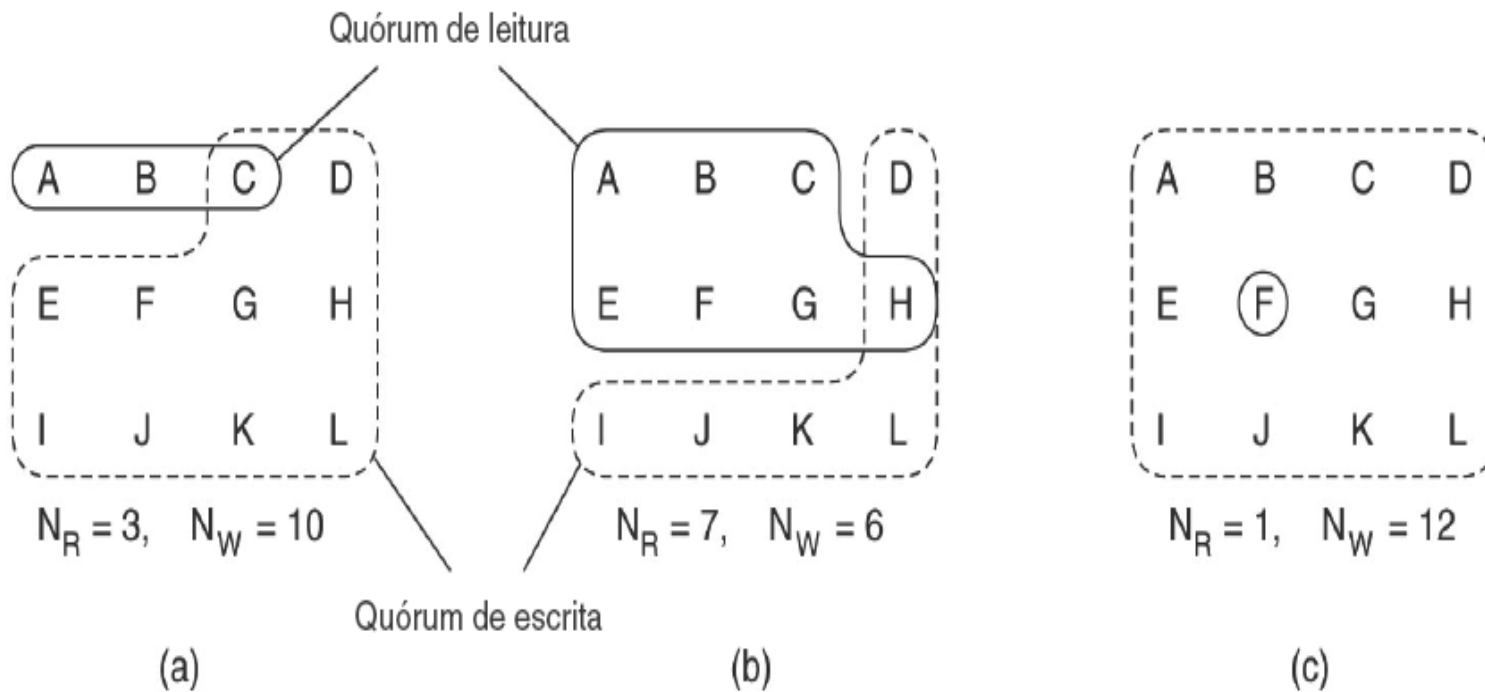
W5. Reconhece atualização

R1. Requisição de leitura

R2. Resposta à leitura

**Figura 7.20** Protocolo de primário e backup no qual a cópia primária migra para o processo que quer realizar uma atualização.

# PROTOCOLO DE ESCRITA REPLICADA

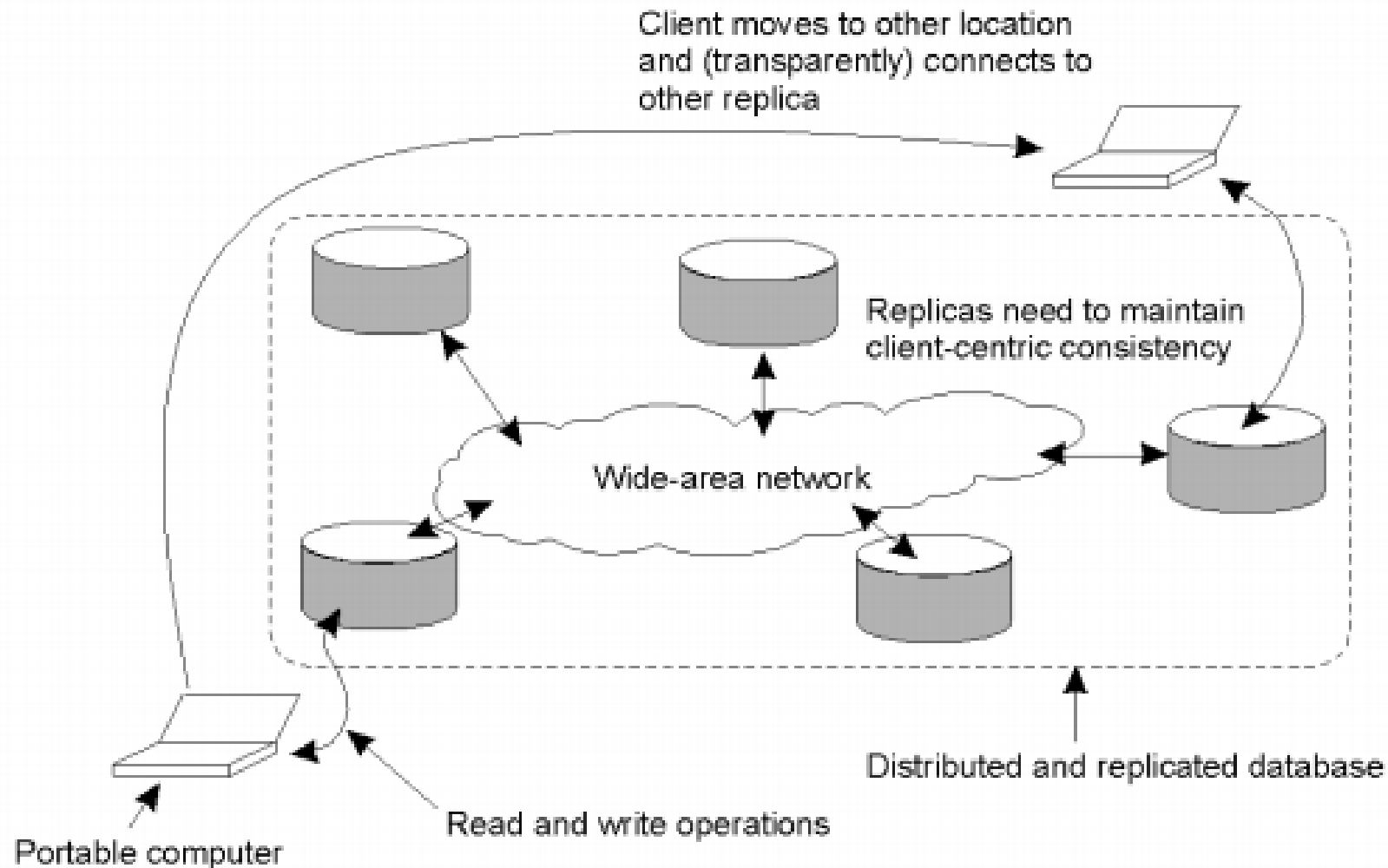


**Figura 7.21** Três exemplos do algoritmo de votação. (a) Escolha correta de conjunto de leitura e de escrita. (b) Escolha que pode levar a conflitos escrita-escrita. (c) Escolha correta, conhecida como ROWA (lê uma, escreve todas).

# Modelo de Consistência centrado no Cliente



# Eventual Consistency



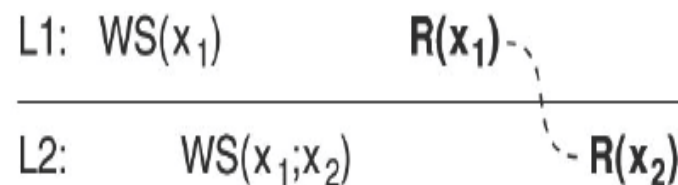
The principle of a mobile user accessing different replicas of a distributed database.

This problem motivates the client-centric consistency

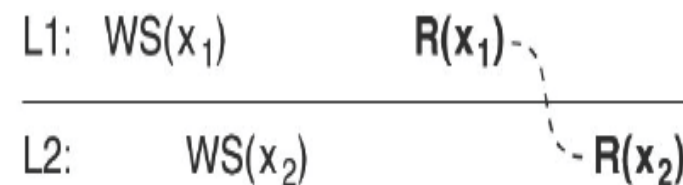
# Monotonic Reads

**Definition:** *If a process reads the values of a data item  $x$ , any successive read operation on  $x$  by that process will always return that same value or a more recent value.*

**Notation:**  $X_i$  denotes the version of data item  $x$  a local copy  $L_i$ ;  $WS(x_i)$  a serie of write operations at  $L_i$ ;  $WS(x_i;x_j)$  a serie of write operations performed at local copy  $L_j$



(a)

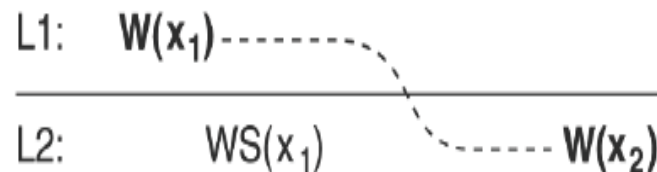


(b)

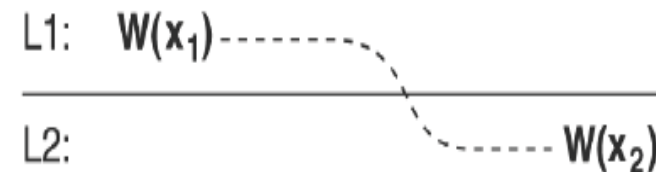
**Figura 7.12** Operações de leitura executadas por um único processo  $P$  em duas cópias locais diferentes do mesmo depósito de dados. (a) Depósito de dados que oferece consistência de leitura monotônica. (b) Depósito de dados que não oferece consistência de leitura monotônica.

# Monotonic Writes

**Definition:** *A write operation by a process on a data item  $x$  is completed before any successive write operation on  $x$  by the same process*



(a)



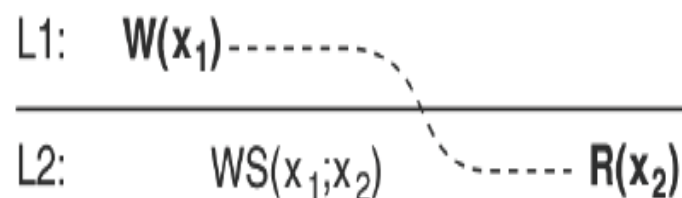
(b)

**Figura 7.13** Operações de escrita executadas por um único processo  $P$  em duas cópias locais diferentes do mesmo depósito de dados. (a) Depósito de dados consistente por escrita monotônica. (b) Depósito de dados que não oferece consistência por escrita monotônica.

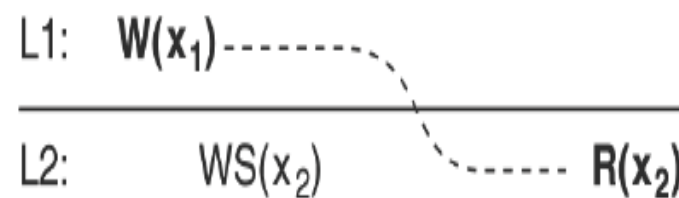
# Read Your Writes

**Definition:** *The effect of a write operation by a process on data item  $x$  will always be seen by a successive read operation on  $x$  by the same process.*

This model has a similar effect of monotonic reads. The only one difference is that the consistency is determined by the last write operation by process P, instead of its last read operation.



(a)

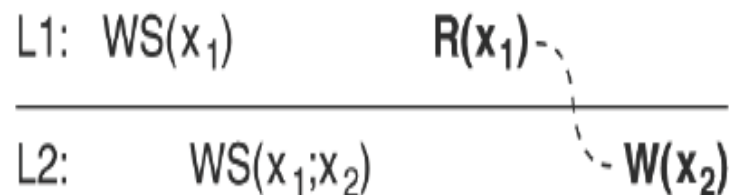


(b)

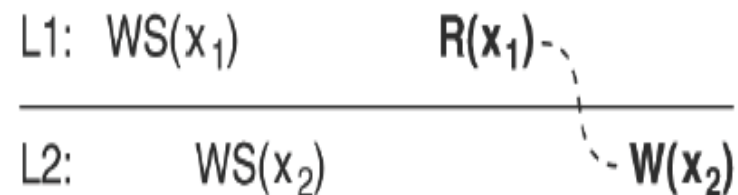
Figura 7.14 (a) Depósito de dados que oferece consistência leia-suas-escritas. (b) Depósito de dados que não fornece tal consistência.

# Writes Follow Reads

**Definition:** *A write operation by a process on a data item  $x$  following a previous read operation on  $x$  by the same process, is guaranteed to take place on the same or a more recent value of  $x$  that was read.*



(a)



(b)

**Figura 7.15** (a) Depósito de dados consistente por escritas-seguem-leituras. (b) Depósito de dados que não provê consistência escritas-seguem-leituras.