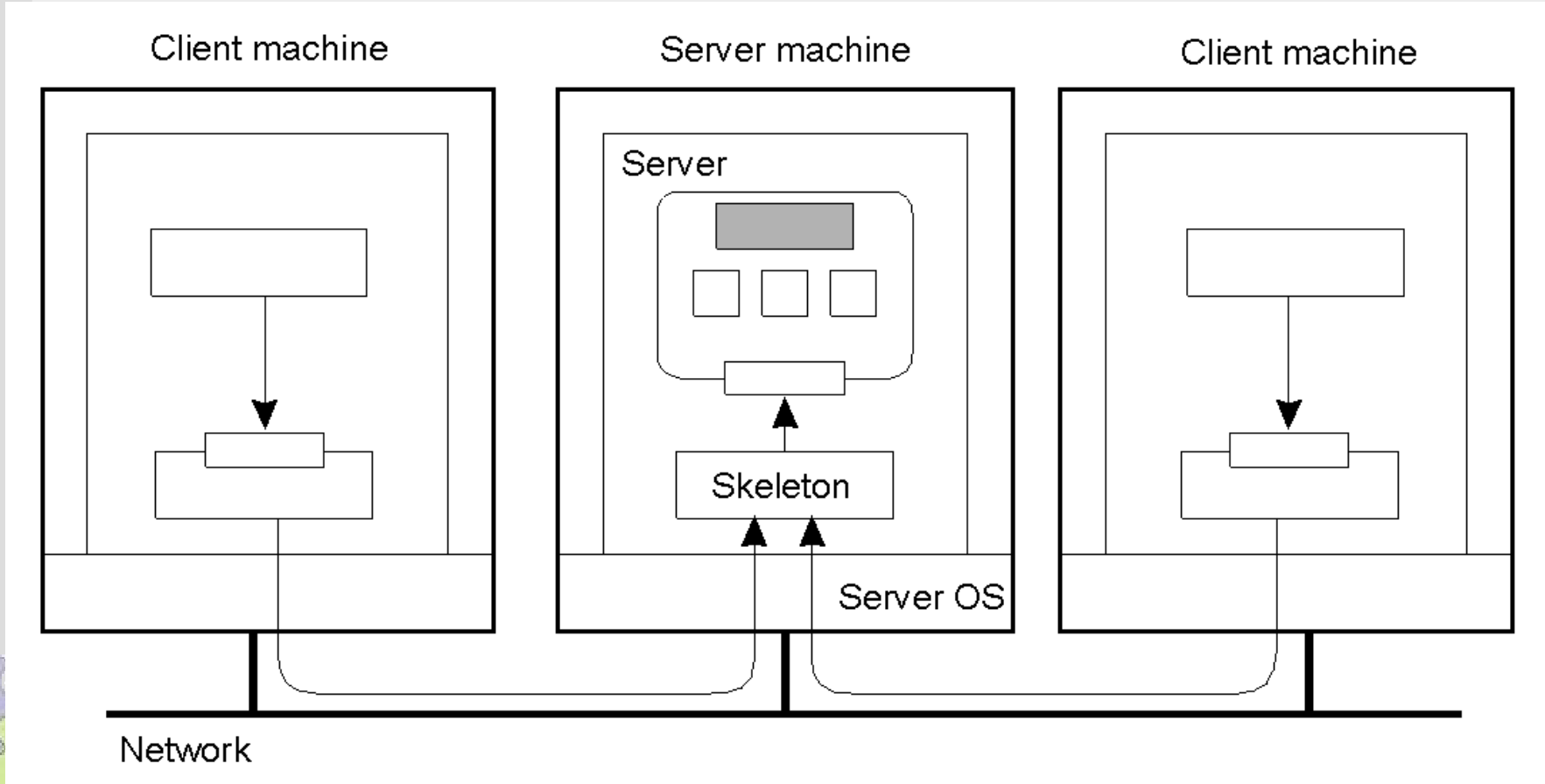
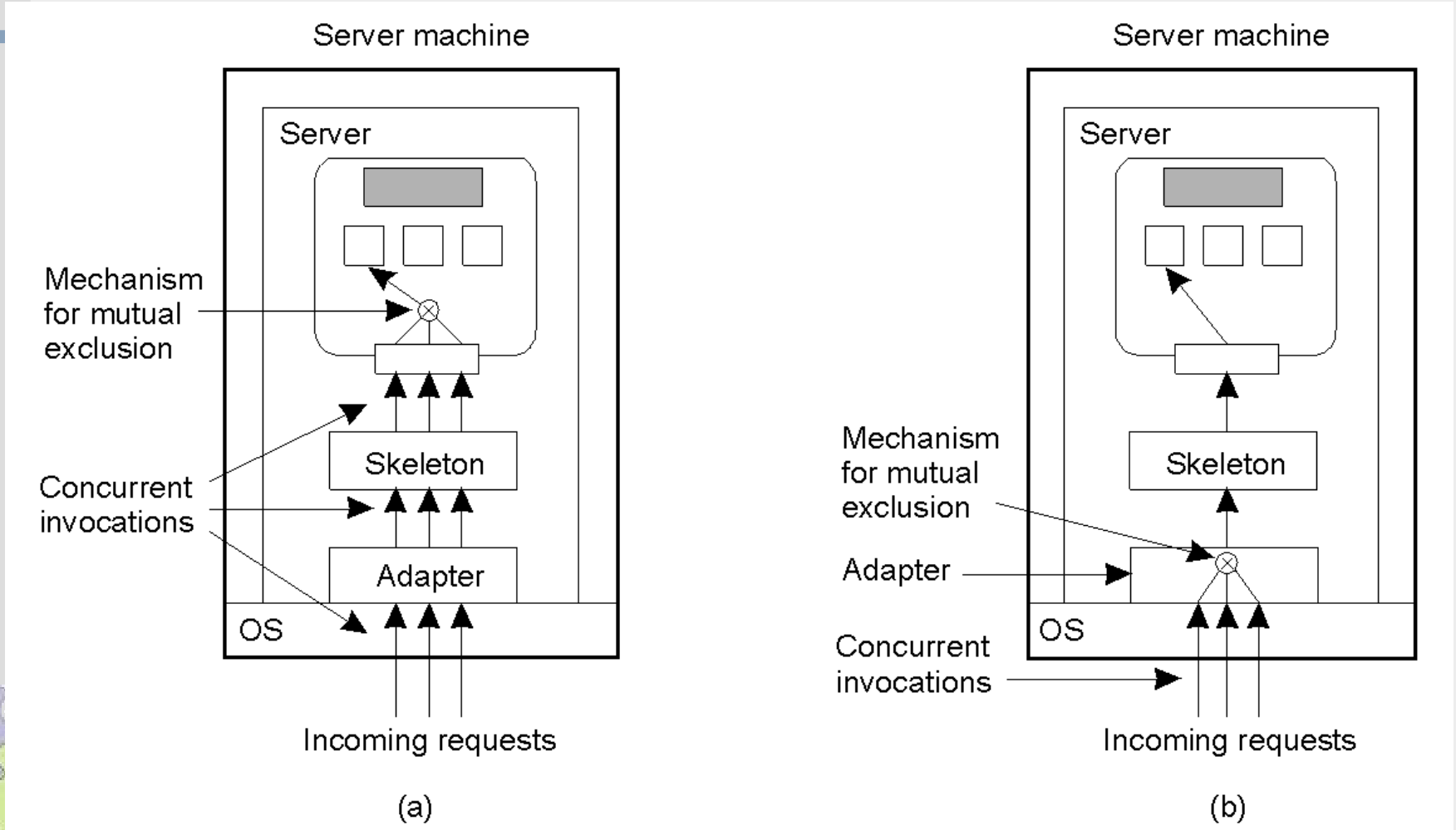
Распределенные системы

Непротиворечивость и репликация

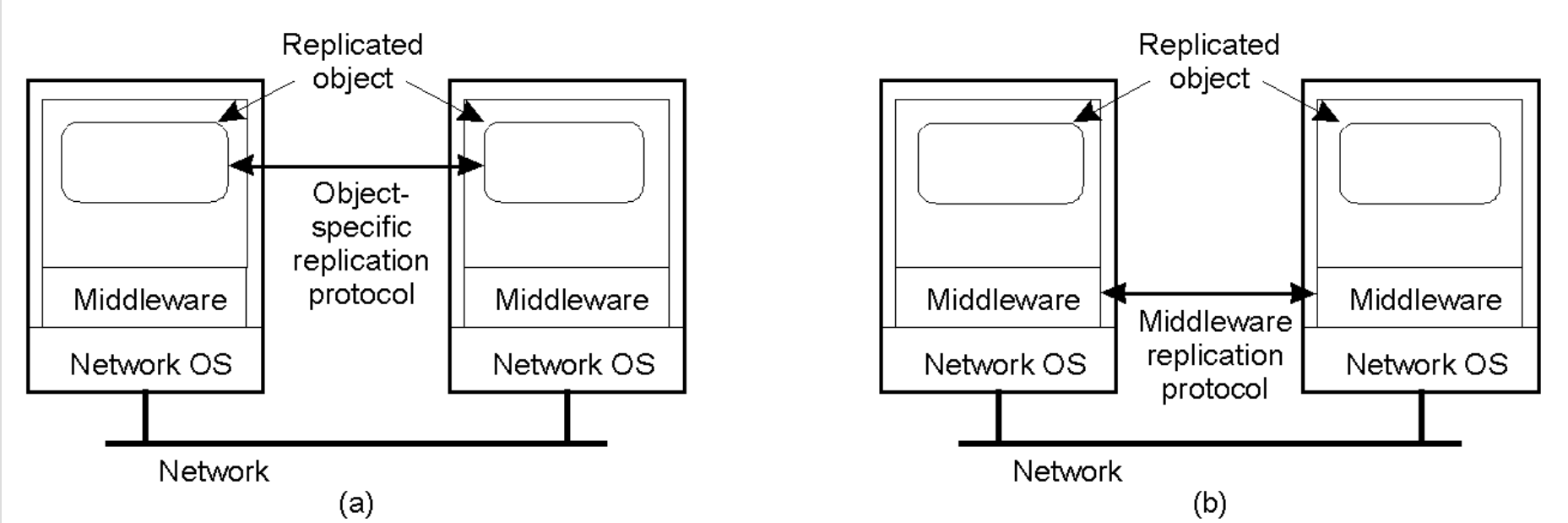
Репликация объектов



Репликация объектов



Репликация объектов



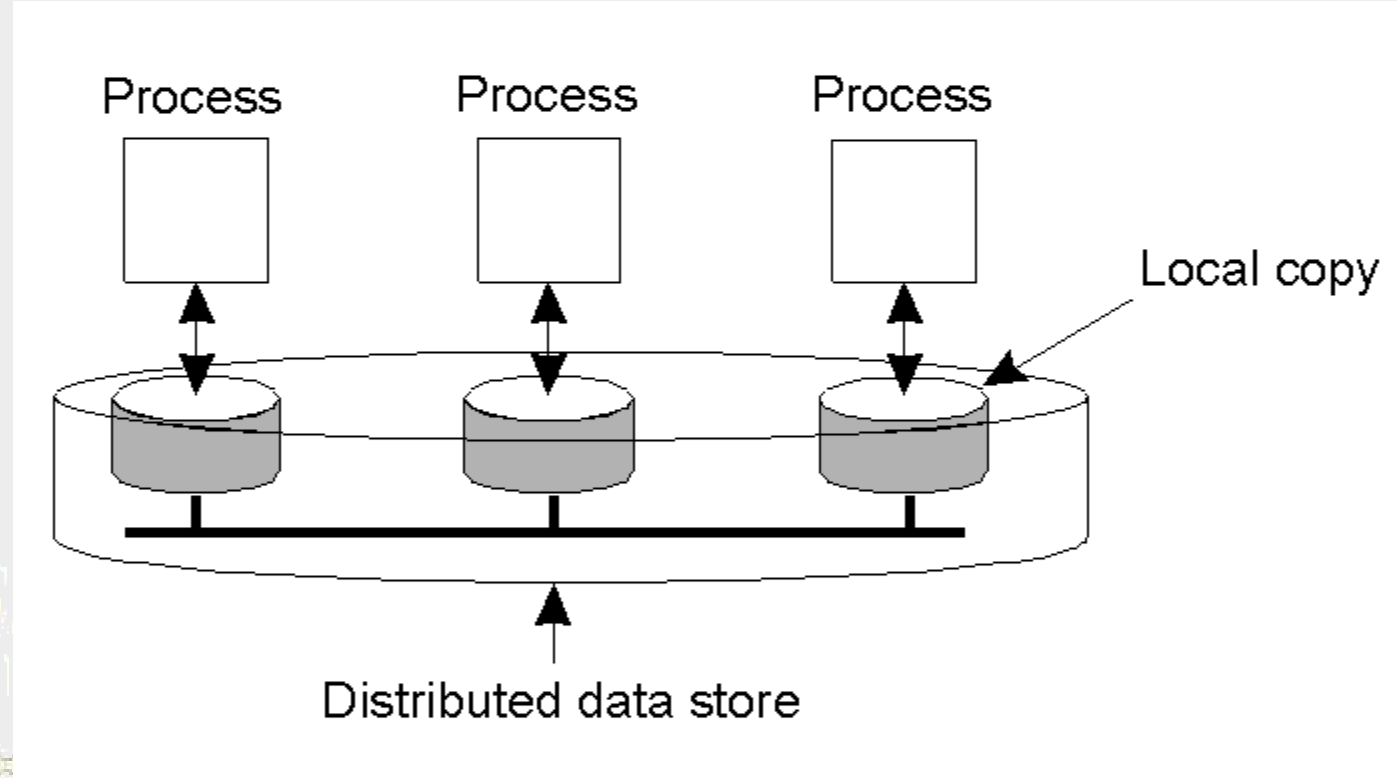
a) Специфический для объектов протокол репликации
b) Протокол репликации промежуточного уровня

Модели непротиворечивости

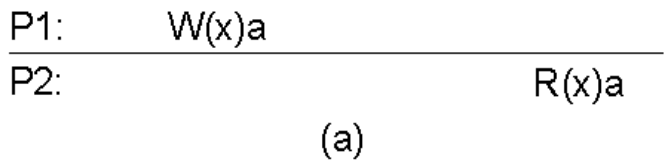
- Хранилище данных
- Операции чтения и записи
- Модель непротиворечивости – контракт между хранилищем данных и множеством процессов
- Обычно процесс ожидает прочитать данные в состоянии, соответствующем последней операции записи – СТРОГАЯ НЕПРОТИВОРЕЧИВОСТЬ

Распределенное хранилище данных

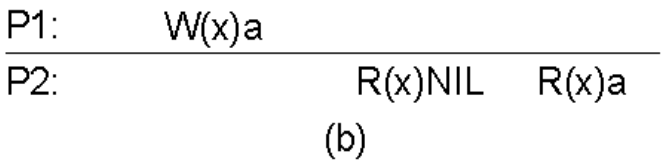
- Общая организация логического хранилища данных, физически распределенного и реплицируемого между несколькими процессами



Строгая непротиворечивость



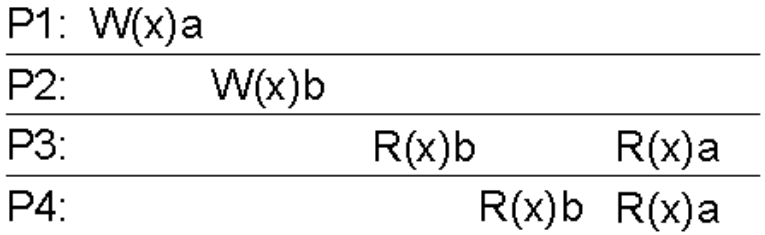
(a)



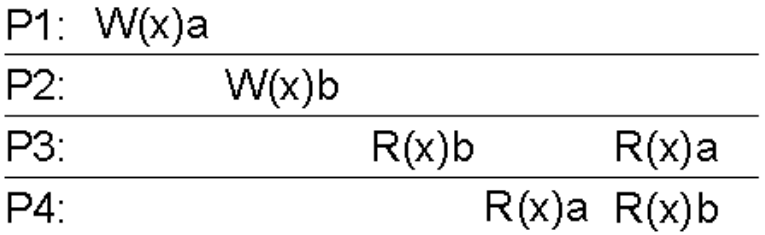
(b)

- Хранилище со строгой непротиворечивостью
- Хранилище без строгой непротиворечивостью

Линеаризуемость и последовательная непротиворечивость



(a)



(b)

a) Хранилище с последовательной непротиворечивостью
b) Хранилище без последовательной непротиворечивости

Все операции упорядочены с точки зрения хранилища, все процессы видят операции в одном порядке – условие последовательной непротиворечивости

Линеаризуемость и последовательная непротиворечивость

- Линеаризуемое хранилище – операции получают глобальную метку времени, с использованием часов, имеющих конечную точность
- Более строгая модель, чем последовательная
- Если $ts_{op1}(x) < ts_{op2}(x)$, то операци OP1 предшествует операции OP2

Линеаризуемость и последовательная непротиворечивость

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

- Три параллельных процесса
- Начальные значения переменных – 0
- Все инструкции атомарны

Линеаризуемость и последовательная непротиворечивость

- 4 варианта корректного выполнения

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

Prints: 001011

Signature: 001011

(a)

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

Prints: 101011

Signature: 101011

(b)

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

Prints: 010111

Signature: 110101

(c)

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

Prints: 111111

Signature: 111111

(d)

Причинная непротиворечивость

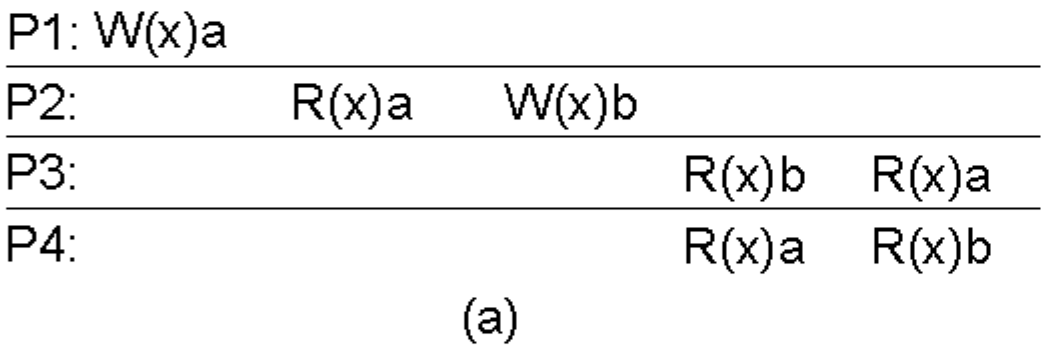
- Необходимое условие:
  - Операции записи, которые потенциально зависимы, должны быть видимы всеми процессами в одном порядке.
  - Параллельные операции могут выполняться в разном порядке на разных машинах

Причинная непротиворечивость

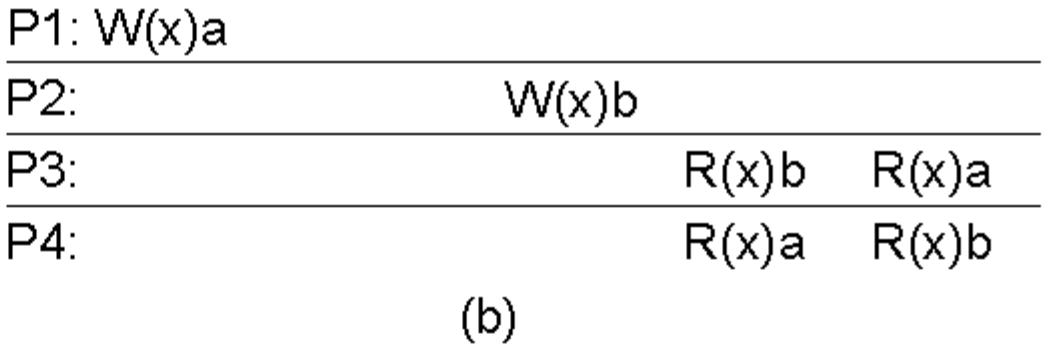
P1:	W(x)a			W(x)c	
P2:		R(x)a	W(x)b		
P3:		R(x)a		R(x)c	R(x)b
P4:		R(x)a		R(x)b	R(x)c

- Данная последовательность удовлетворяет условиям причинной непротиворечивости, но не последовательной или строгой непротиворечивости
- Операции записи, связанные причинно-следственной связью должны наблюдаться всеми процессами в едином порядке, а параллельные операции записи могут наблюдаться в разном порядке в разных процессах(W2(x)b, W1(x)c)

Причинная непротиворечивость



(a)



(b)

a) Нарушение причинной непротиворечивости

b) Корректная последовательность событий при причинной непротиворечивости

Непротиворечивость FIFO

- Записи, выполненные одним процессом, видны всем остальным процессам в том порядке, в котором они были выполнены, но операции записи, выполненные разными процессами могут быть видны в разном порядке

Непротиворечивость FIFO

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

- Корректная последовательность выполнения команд

Непротиворечивость 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)

- Очередность выполнения операций с точки зрения процессов P1(a), P2(b), P3(c)

Непротиворечивость FIFO

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

- Два параллельных процесса могут быть прерваны оба, с точки зрения FIFO

Слабая непротиворечивость

- НЕ ВСЕ ПРИЛОЖЕНИЯ нуждаются в том, чтобы наблюдать все операции записи!
- Процесс находится в критической секции и не нуждается в репликации промежуточных результатов
- Операция - Synchronize(S), S – синхронизирующая переменная
- Свойства:
  - Доступ к синхронизирующим переменных обладает последовательной непротиворечивостью
  - С переменной не может быть произведена ни одна операции записи до полного завершения предыдущих операций
  - С элементами данных не может быть произведена ни одна операций до полного завершения всех операций с синхронизирующими переменными
- Непротиворечивое выполнение группы операций, конвейер выполнения операций

Слабая непротиворечивость

```
int a, b, c, d, e, x, y;          /* variables */
int *p, *q;                       /* pointers */
int f( int *p, int *q);           /* function prototype */

a = x * x;                        /* a stored in register */
b = y * y;                        /* b as well */
c = a*a*a + b*b + a * b;          /* used later */
d = a * a * c;                    /* used later */
p = &a;                           /* p gets address of a */
q = &b                            /* q gets address of b */
e = f(p, q)                       /* function call */
```

- Значения переменных могут быть сохранены в регистрах до определенного момента

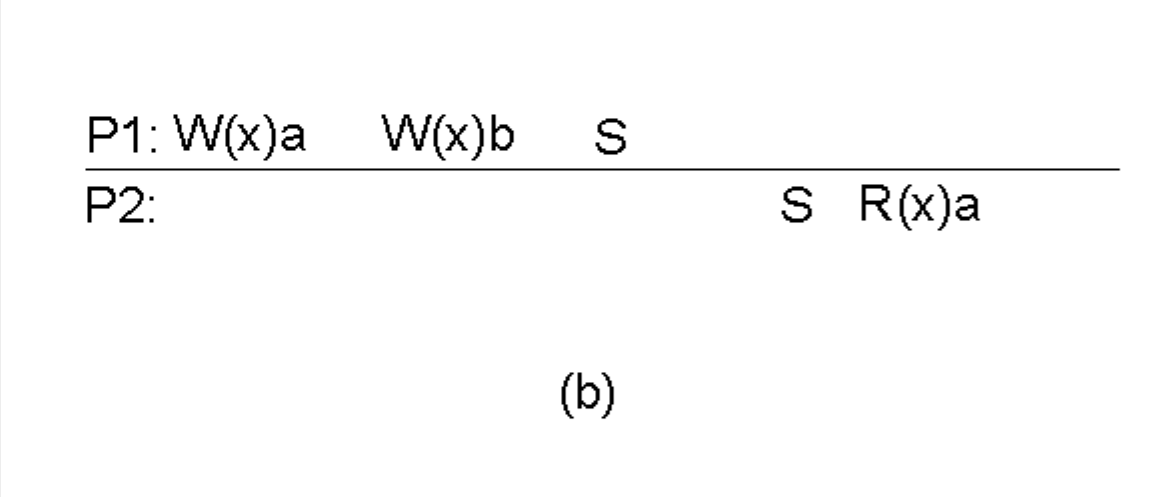
Слабая непротиворечивость

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

(a)

a) Корректная последовательность событий для слабой непротиворечивости

Слабая непротиворечивость



b) Недопустимая последовательность для слабой непротиворечивости

Свободная непротиворечивость

P1:	Acq(L)	W(x)a	W(x)b	Rel(L)						
P2:					Acq(L)	R(x)b	Rel(L)			
P3:										R(x)a

- Корректная последовательность для свободной непротиворечивости

Свободная непротиворечивость

- Операции захвата (Acquire) и освобождения (Release)

Правила:

- Перед выполнением операций чтения и записи, все предыдущие захваты этого процесса должны быть завершены
- Перед выполнением освобождения все предыдущие операции записи и чтения должны быть завершены
- Доступ к синхронизирующим переменным должен обладать непротиворечивостью FIFO

- Варианты реализации: ленивая(lazy) свободная непротиворечивость

Поэлементная непротиворечивость

- Каждый совместно используемый элемент данных ассоциируется со своей переменной синхронизации
- После захвата переменной синхронизации непротиворечивыми становятся только данные ассоциированные с ней

Правила:

- Захват доступа к синхронизирующей переменной не разрешается до тех пор, пока не осуществлены все обновления отслеживаемых совместных используемых данных
- Пока один из процессов имеет эксклюзивный доступ к переменной синхронизации, никакой другой процесс не может захватить эту переменную.
- После эксклюзивного доступа к переменной, любой другой процесс может получить доступ к переменной только с разрешения процесса-владельца

Реализация прозрачного получения доступа к переменной с точки зрения клиента

Поэлементная непротиворечивость

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

- Корректная последовательность для поэлементной непротиворечивости

Модели непротиворечивости

Непротиворечивость	**Описание (без использования средств синхронизации)**
Строгая	Абсолютная упорядоченность по времени выполнения
Линеаризуемая	Все процессы видят обращения других совместных процессов в одинаковом порядке. Обращения упорядочены в соответствии с глобальным, но не уникальным временем
Последовательная	Все процессы видят обращения других совместных процессов в одинаковом порядке. Обращения не упорядочены
Причинная	Все процессы видят причинно-следственный порядок выполнения в одном и том же порядке.
FIFO	Все процессы видят записи в порядке их использования по отношения к другому процессу. Операции записи разных процессов не всегда видятся в таком порядке

Модели непротиворечивости

Непротиворечивость	**Описание (с использованием средств синхронизации)**
Слабая	Разделяемые данные считаются непротиворечивыми после выполнения синхронизации
Свободная	Разделяемые данные становятся непротиворечивыми после выхода из критической секции
Поэлементная	Разделяемые данные, ожидающие вхождения в критическую секцию, становятся непротиворечивыми в момент входа в секцию

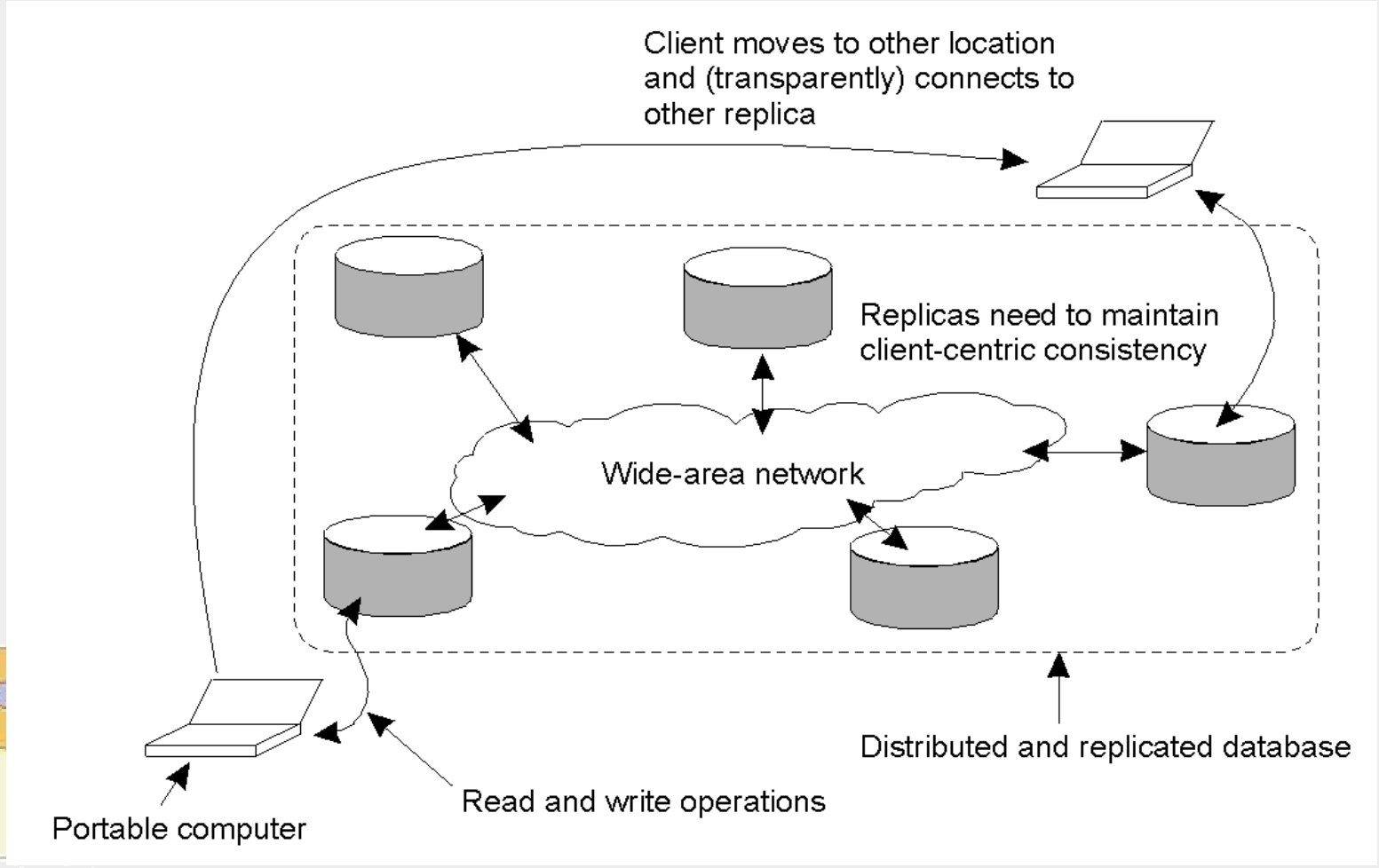
Модели непротиворечивости, ориентированные на клиента

- Специальный класс хранилищ данных
- Отсутствие одновременных изменений или легкость разрешения
- Большинство операций – операции чтения

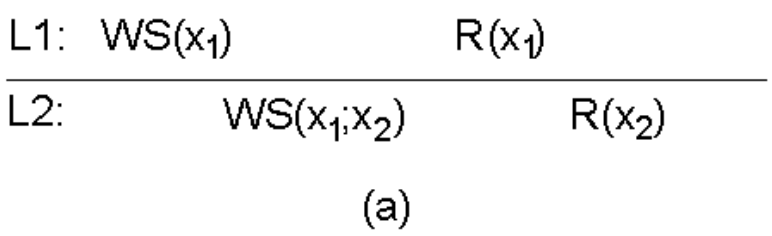
Примеры использования: DNS, WWW

- Потенциальная непротиворечивость – гарантия одному клиенту на непротиворечивый доступ к хранилищу данных

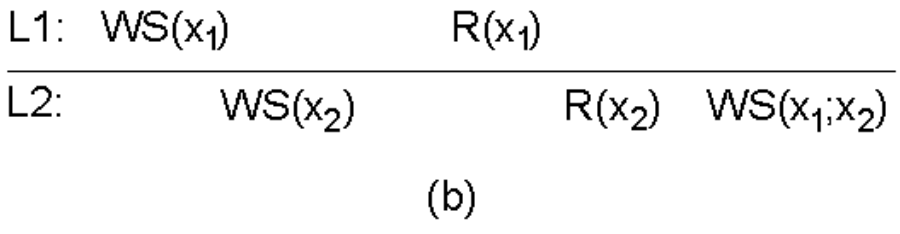
Потенциальная непротиворечивость



Монотонное чтение



(a)



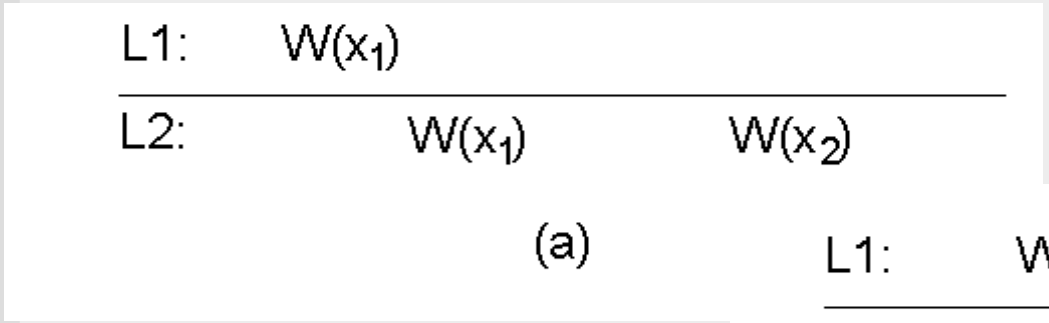
(b)

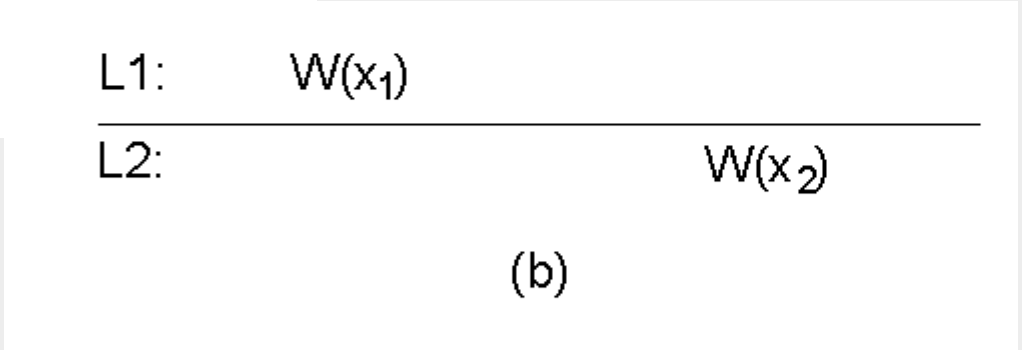
- Операции чтения, выполняемые процессом P из двух различных локальных копий одного и того же хранилища данных

a) Непротиворечивость монотонного чтения данных

b) Хранилище данных с нарушением монотонного чтения данных

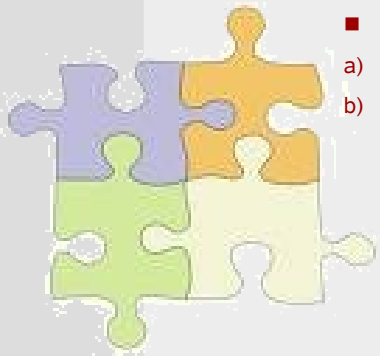
Монотонная запись



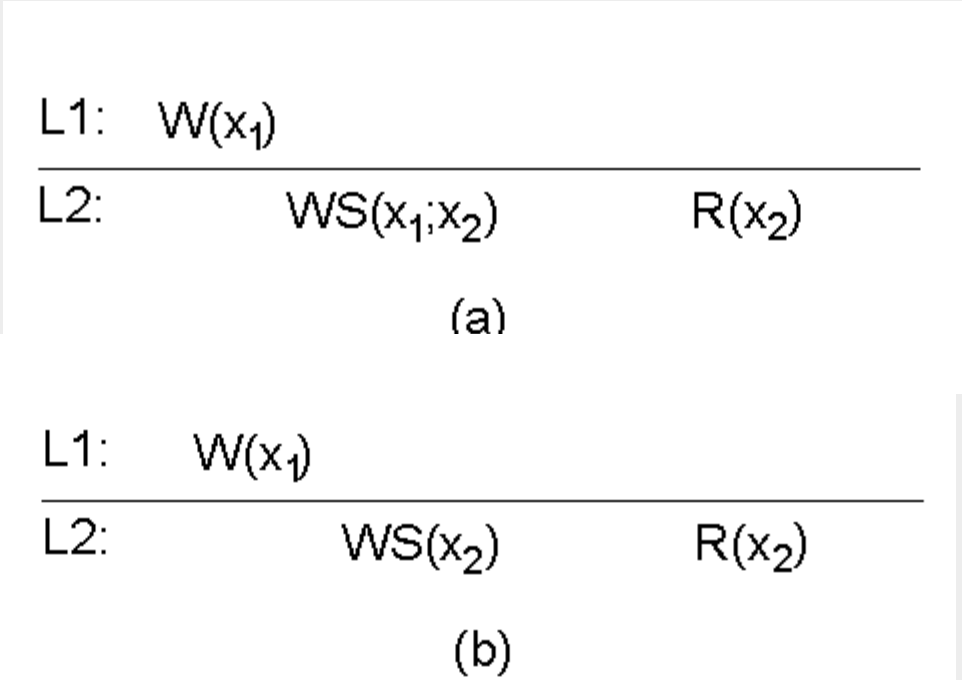


- Операция записи процесса в элемент x завершается раньше любой другой записи этого процесса. Процесс всегда «видит» результат своих предыдущих изменений
- Операция записи процесса P для двух различных локальных копий хранилища

a) Хранилище с непротиворечивостью монотонной записи
b) Хранилище данных без непротиворечивости монотонной записи

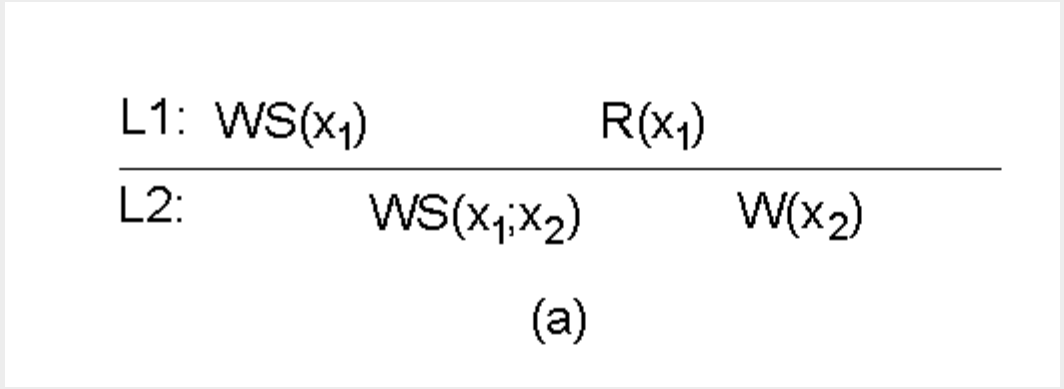


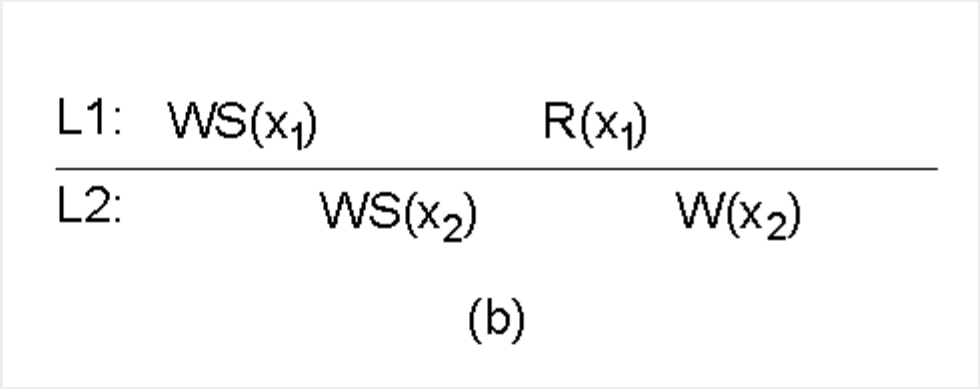
Чтение собственных записей



- Примеры: изменение web-страницы, изменение пароля

Запись за чтением

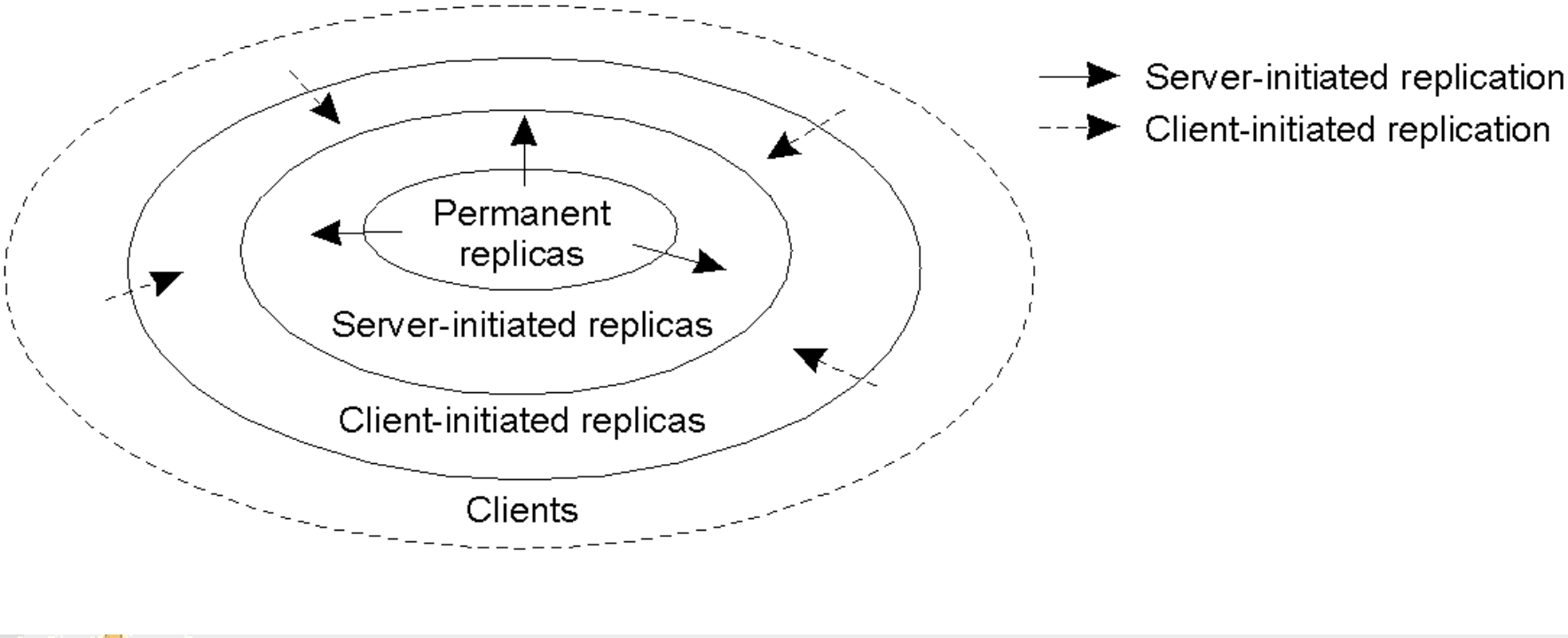




- Ответ на письмо в группе новостей

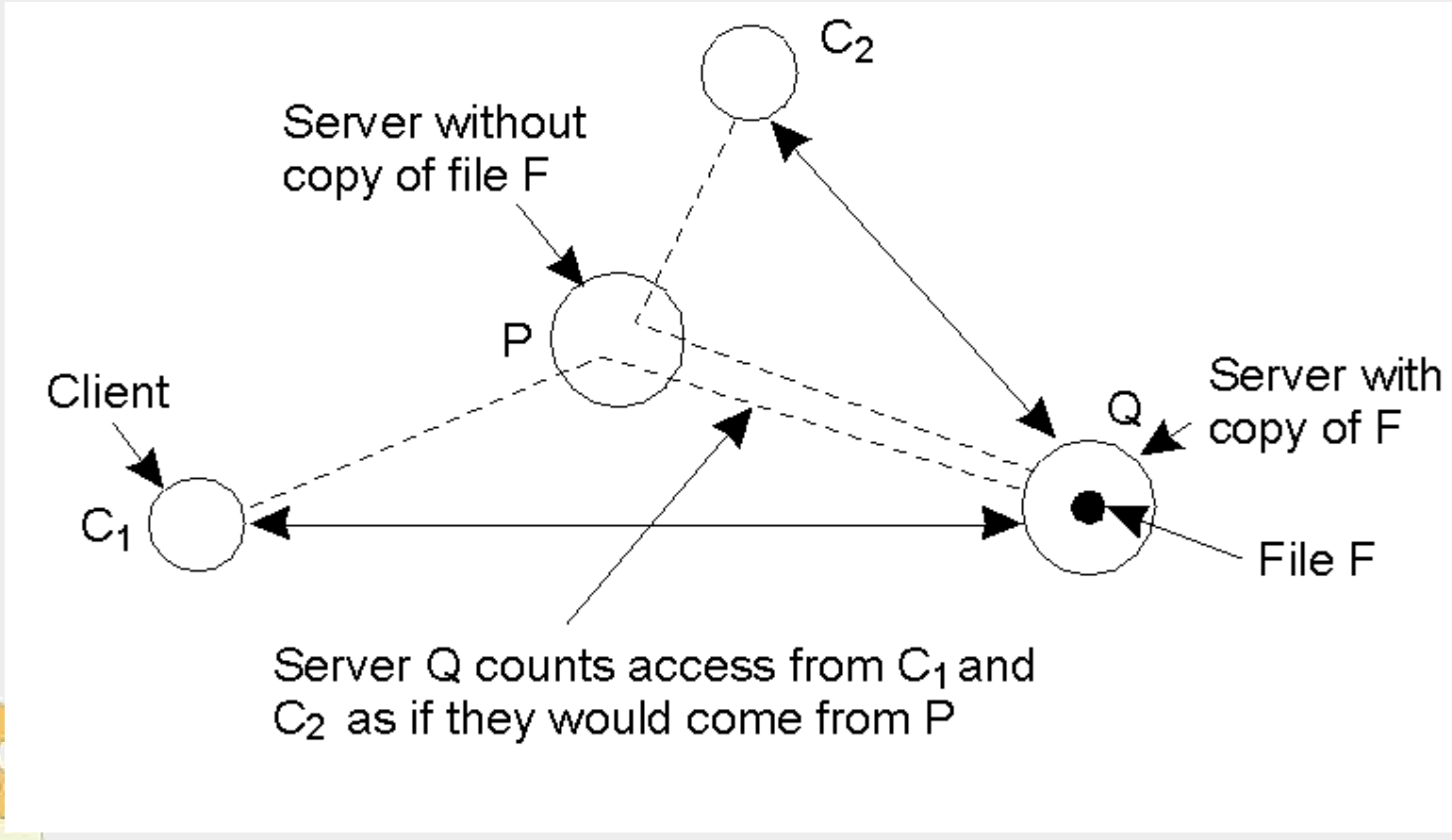
Протоколы распределения

Размещение реплик



- Логическая организация различных типов копий хранилищ данных
- Пример постоянных реплик: WEB – зеркала, кластеры рабочих станций

Реплики, инициируемые сервером



Реплики, инициируемые клиентом

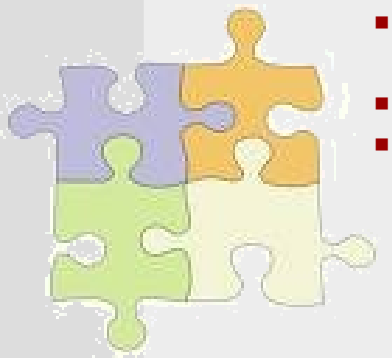
- Кэш WEB-страниц
- Кэш совместно используемых файлов
- Ограничение времени хранения
- Использование промежуточных уровней

Распространение обновлений

- Распространение извещений об обновлении (invalidation protocol)
- Передача данных из одной копии в другую
- Распространение операций обновления

Продвижение(Push) против извлечения(Pull)

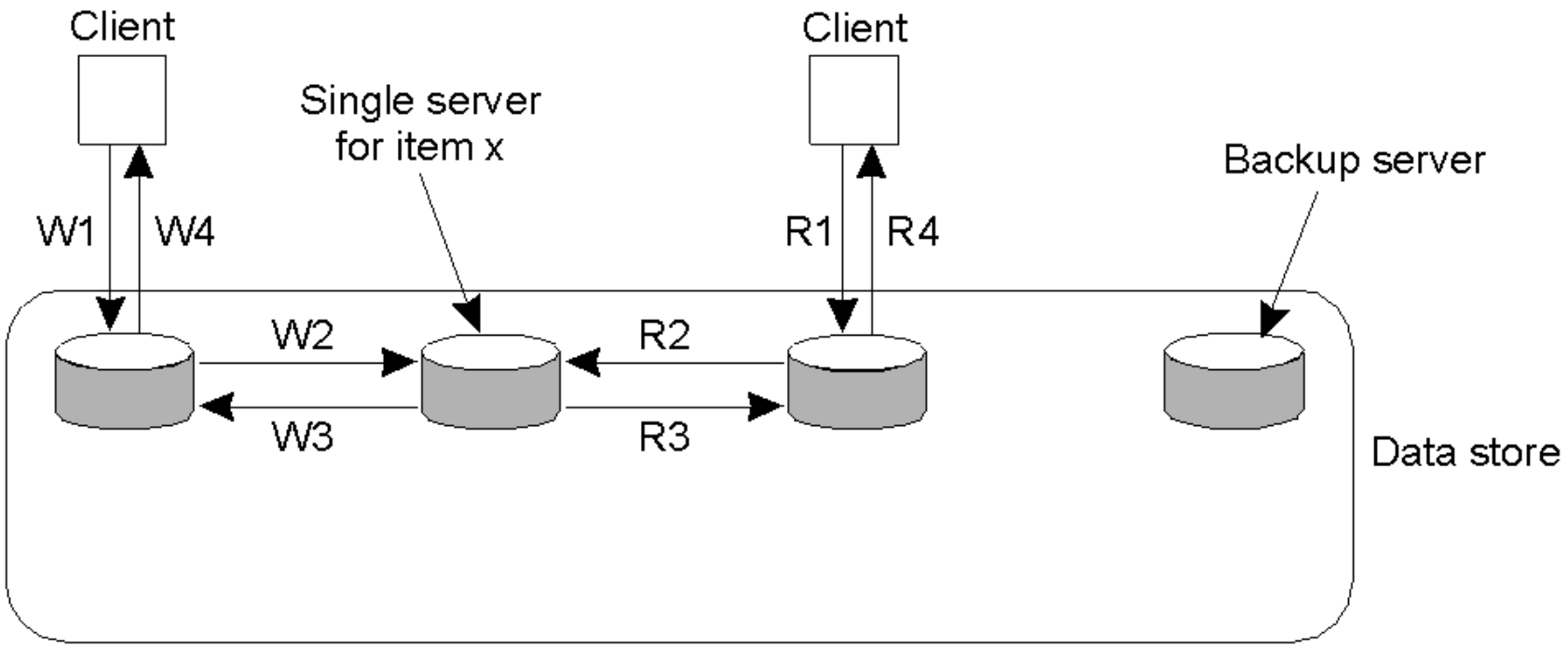
Аспект	Push	Pull
Состояние сервера	Список клиентских реплик и кэшей	Ничего
Отправка сообщений	Обновление (и возможно извлечение обновления позже)	Запрос и обновление
Время отклика для клиента	Немедленно (или время извлечения обновления)	Fetch-update time



- 1 сервера, несколько клиентов
- Смешанная форма распространения обновлений – аренда
- Эпидемические протоколы

Протоколы непротиворечивости

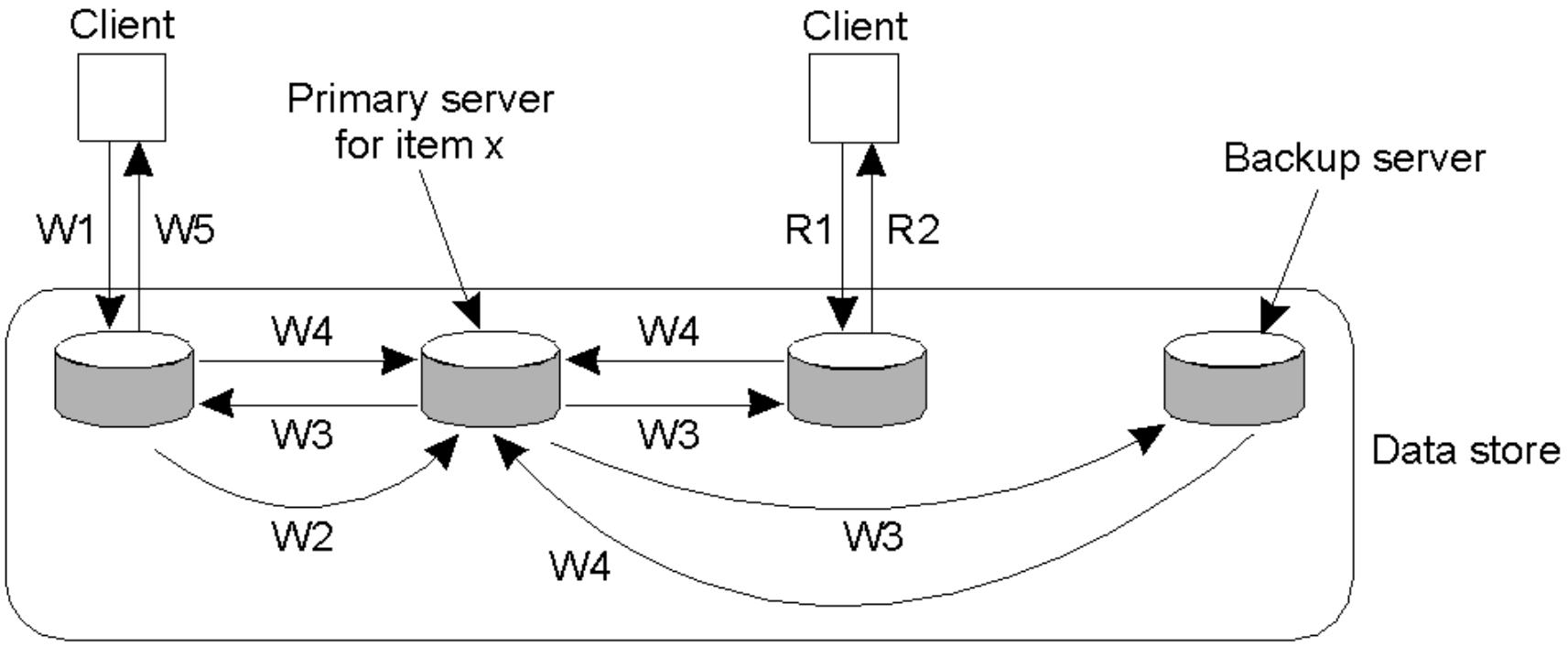
Протокол удаленной записи



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

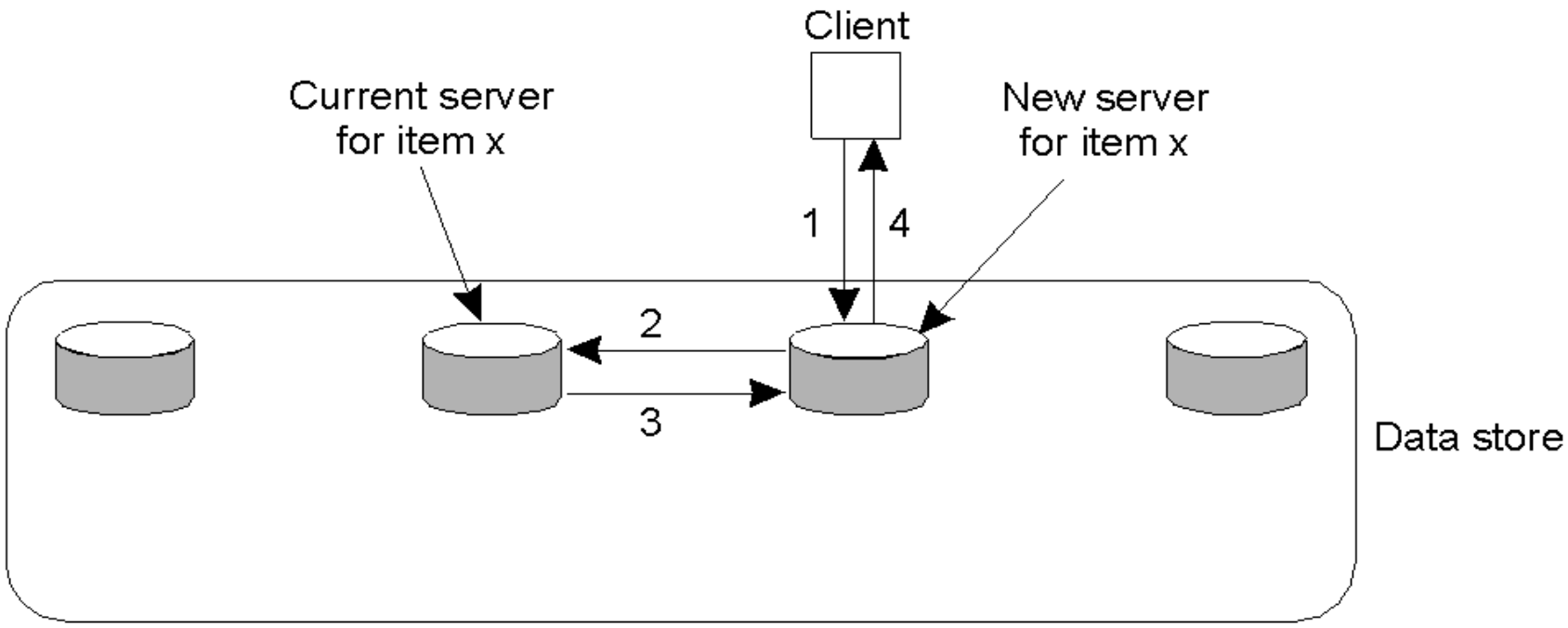
Протокол удаленной записи



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

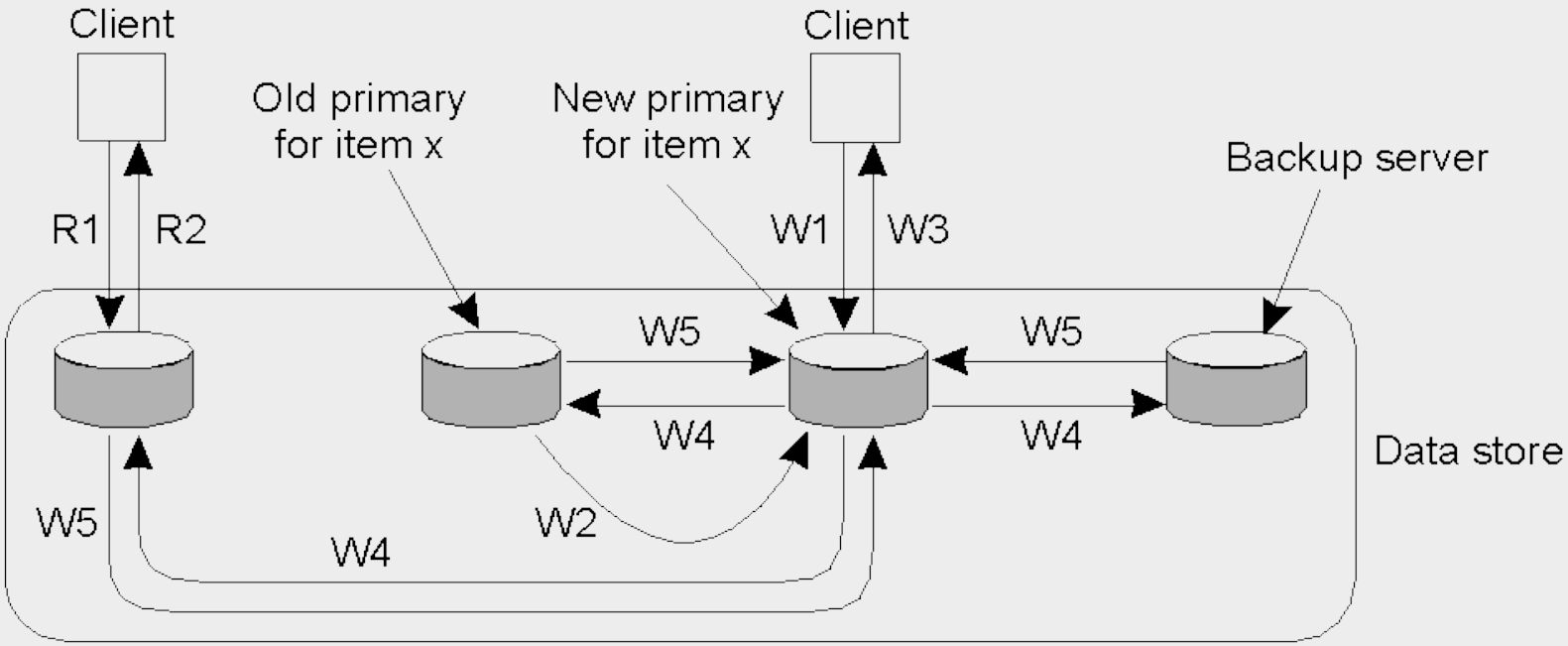
R1. Read request
R2. Response to read

Протокол локальной записи



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

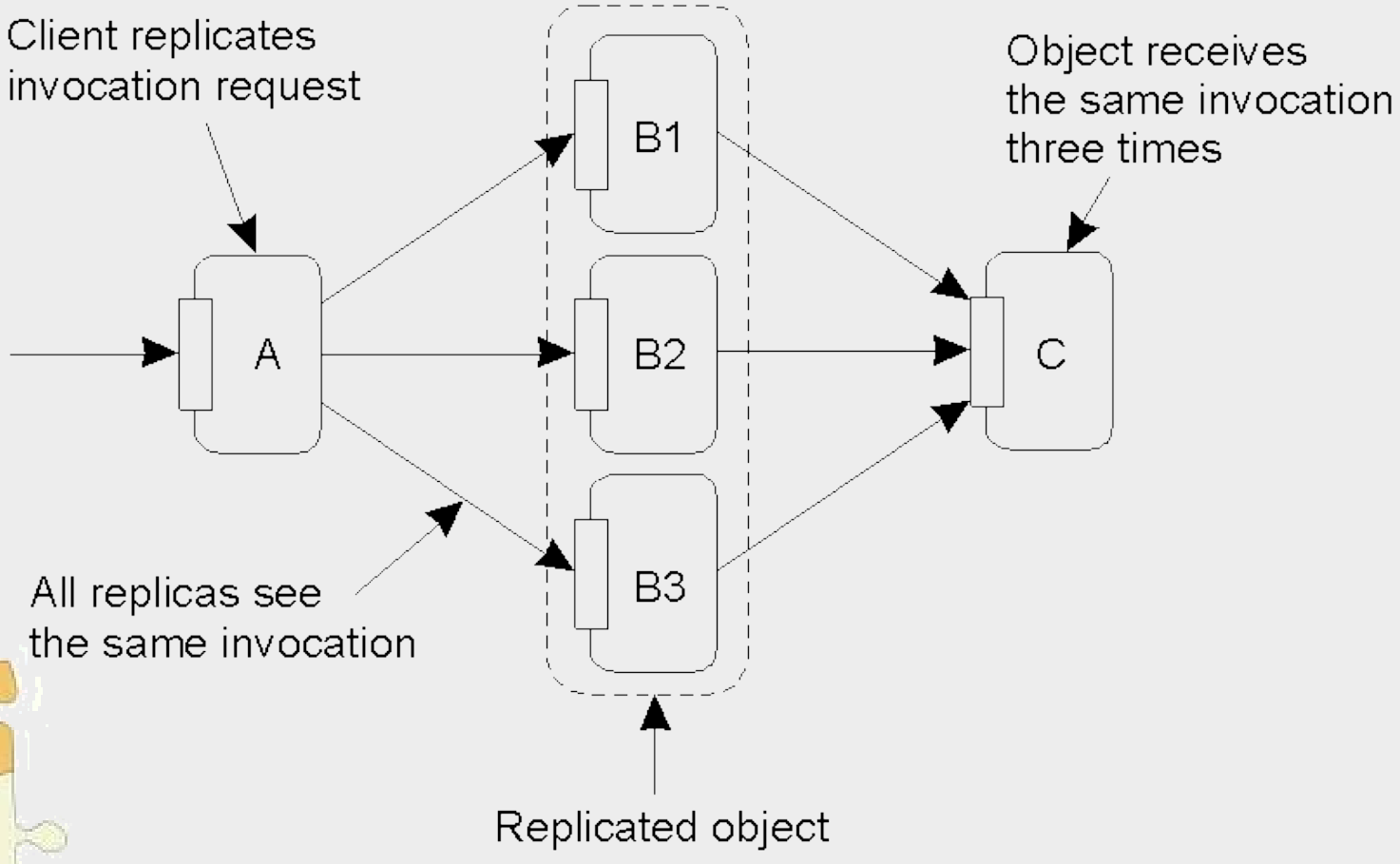
Протокол локальной записи



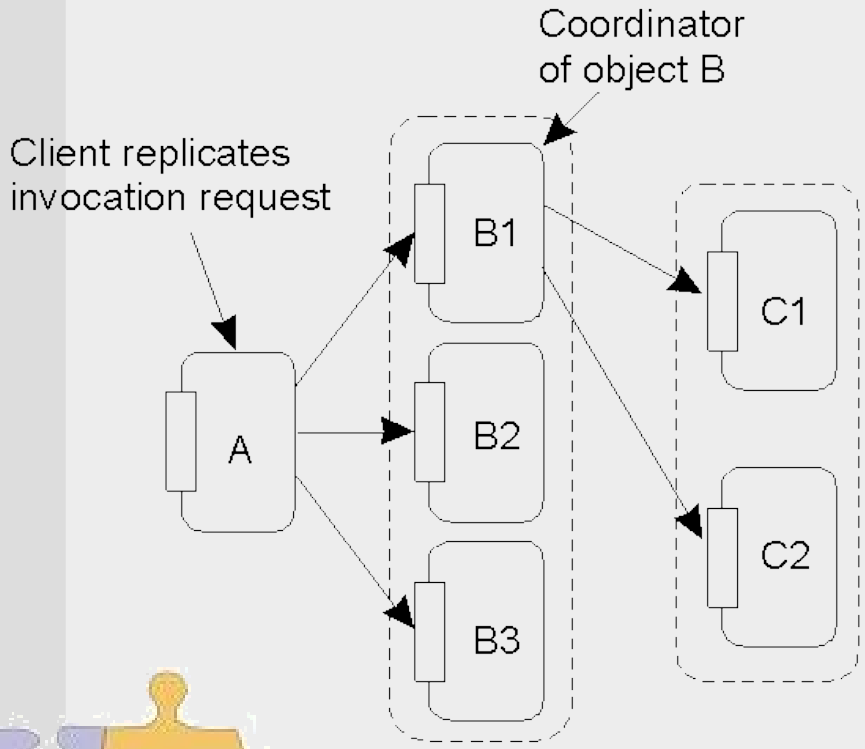
W1. Write request
W2. Move item x to new primary
W3. Acknowledge write completed
W4. Tell backups to update
W5. Acknowledge update

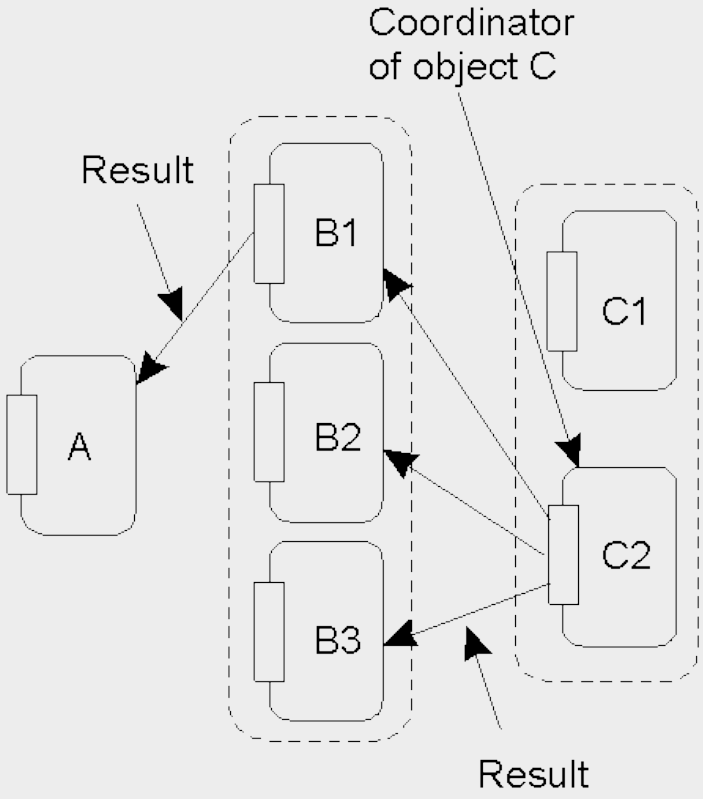
R1. Read request
R2. Response to read

Активная репликация



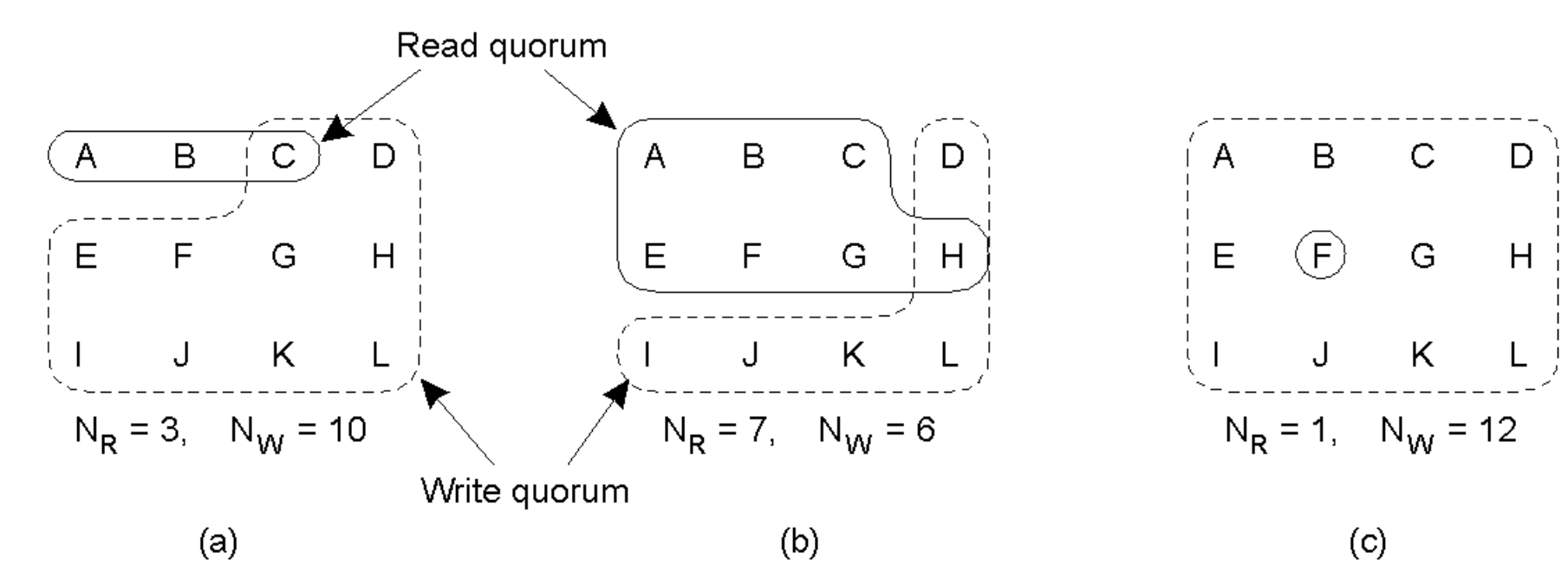
Активная репликация





- Проблема многократного выполнения
- Координаторы объектов

Протоколы кворума



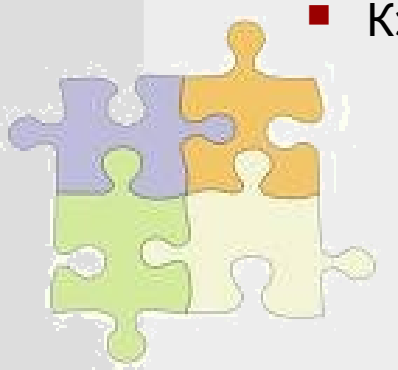
Примеры работы алгоритма:

a) Корректный выбор множества серверов для чтения и записи
b) Выбор, приводящий к ошибке WRITE-WRITE
c) Правильный выбор ROWA (read one, write all)

$N_R + N_W > N, N_W > N/2$

Протоколы согласования кэшей

- Стратегия обнаружения согласованности
  - Обязательное подтверждение непротиворечивости
  - Оптимистичное использование значения
- Стратегия установления согласованности
  - Запрет кэширование
  - Рассылка оповещений
  - Распространение обновлений
- Модификация кэшируемых данных
  - Кэш сквозной записи (write-through cache)
  - Кэш отложенной записи(write-back cache)



Вопросы?