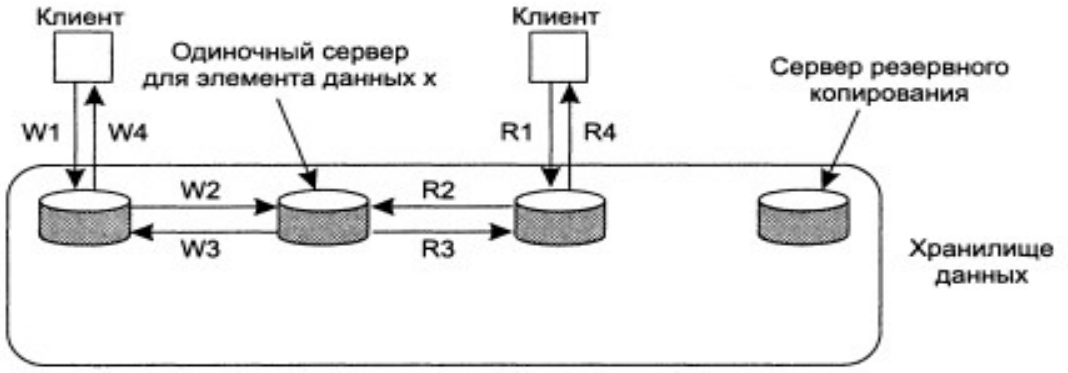
Лекция 08
РИС, III курс, ИСиТ

Протоколы непротиворечивости

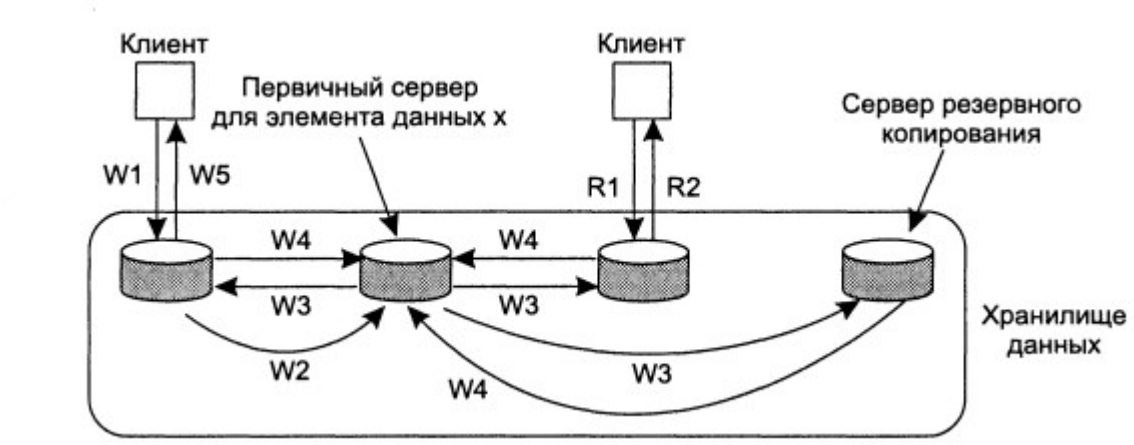
1. **Напоминание:** модели непротиворечивости.
2. **Протокол непротиворечивости:** реализация одной из модели непротиворечивости.
3. **Классификация протоколов непротиворечивости:** 1) существует первичная копия данных: все операции записи должны выполняться в первичной копии; 2) нет первичной копии: операцию записи может инициировать любая реплика.
4. **Протоколы на базе первичной копии:** протокол удаленной записи.

- *W1* — запрос на запись;
- *W2* — пересылка запроса на сервер;
- *W3* — подтверждение выполнения записи;
- *W4* — подтверждение выполнения записи;
- *R1* — запрос на чтение;
- *R2* — пересылка запроса на сервер;
- *R3* — возвращение ответа;
- *R4* — возвращение ответа.



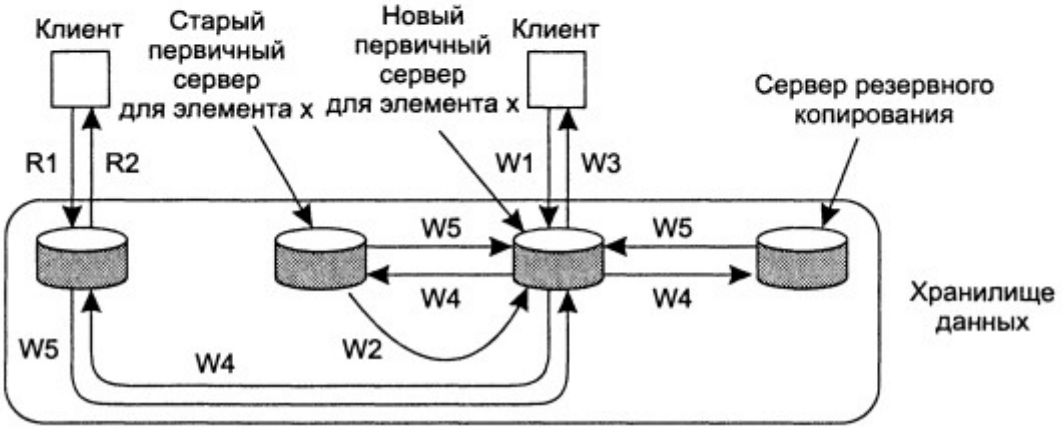
5. **Протоколы на базе первичной копии:** протокол первичного архивирования (primary backup protocol); реализует последовательную непротиворечивость; чтение с локального сервера; запись на центральный сервер; протоколы с блокирующим и неблокирующим режимом обновления (блокирующий – синхронный: пока не обновлены все реплики процесс не получает W4; неблокирующий – асинхронный: W4 получает сразу после обновления локальной копии); неблокирующий режим трудно защитить от сбоев.

- *W1* — запрос на запись;
- *W2* — пересылка запроса на первичный сервер;
- *W3* — сигнал на обновление резервных копий;
- *W4* — подтверждение обновления;
- *W5* — подтверждение выполнения записи;
- *R1* — запрос на чтение;
- *R2* — ответ для чтения.

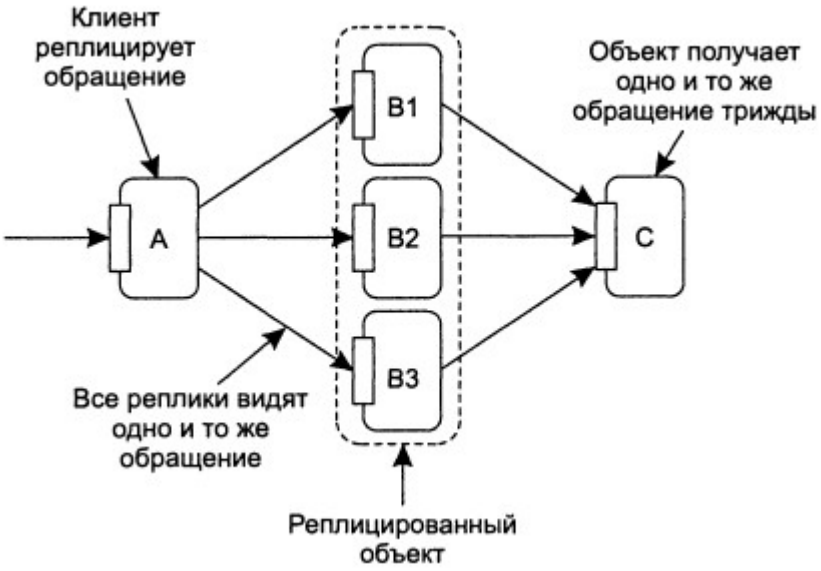


6. **Протоколы на базе первичной копии:** протоколы локальной записи; поддерживается единственная копия данных;
7. **Протоколы на базе первичной копии:** поддержка единственная копия данных; единственная копия данных перемещается между серверами; проблема; отследить месторасположение данных; обычно применяется широковещательный запрос для поиска данных. Пример: база данных GPS; мобильные компьютеры, не имеющих постоянного соединения с сетью.

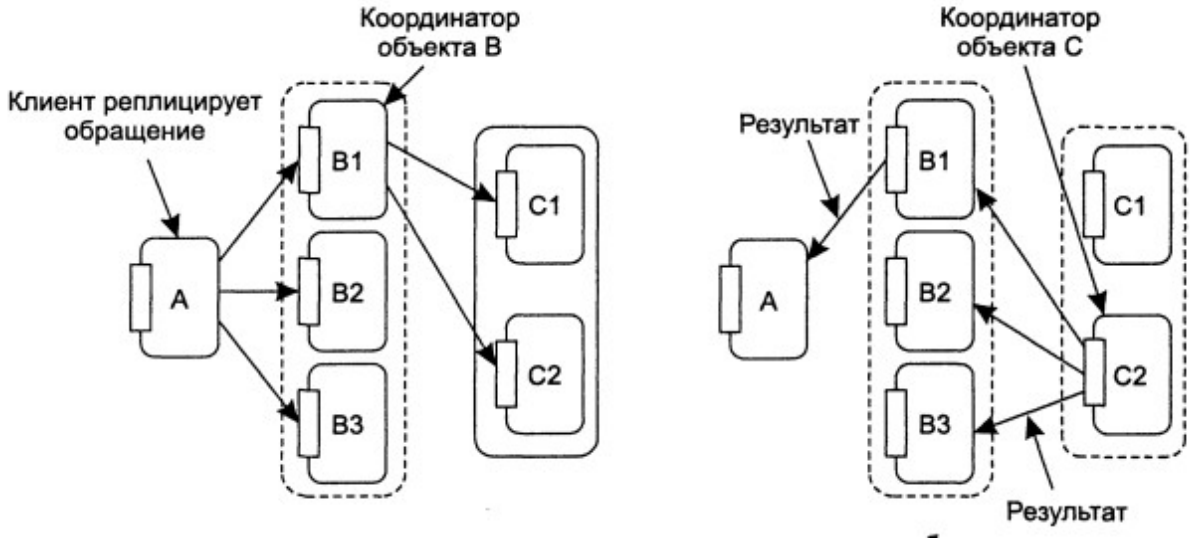
- *W1* — запрос на запись;
- *W2* — перемещение элемента данных *x* на новый первичный сервер;
- *W3* — подтверждение завершения записи;
- *W4* — сигнал на обновление резервных копий;
- *W5* — подтверждение обновления;
- *R1* — запрос на чтение;
- *R2* — ответ для чтения.



8. **Протоколы на базе реплицируемой записи:** 1) активная репликация; 2) протоколы кворума.
9. **Протоколы на базе реплицируемой записи:** активная репликация: с каждой репликой связан процесс-координатор, который выполняет реплицированную ему операцию обновления или записывает переданные ему реплицированные данные. Существует 2 проблемы: 1) все обновления должны осуществляться в одном порядке на каждой реплике; 2) как выполнять реплицированные обращения (транзитные репликации)
10. **Протоколы на базе реплицируемой записи:** активная репликация: для обеспечения правильной последовательности может быть применен централизованный сервер (sequencer), выполняющий хронологическую нумерацию, но в этом случае фактически осуществляется протокол на базе первичной копии;
11. **Протоколы на базе реплицируемой записи:** активная репликация: размножаются реплицированные сообщения.

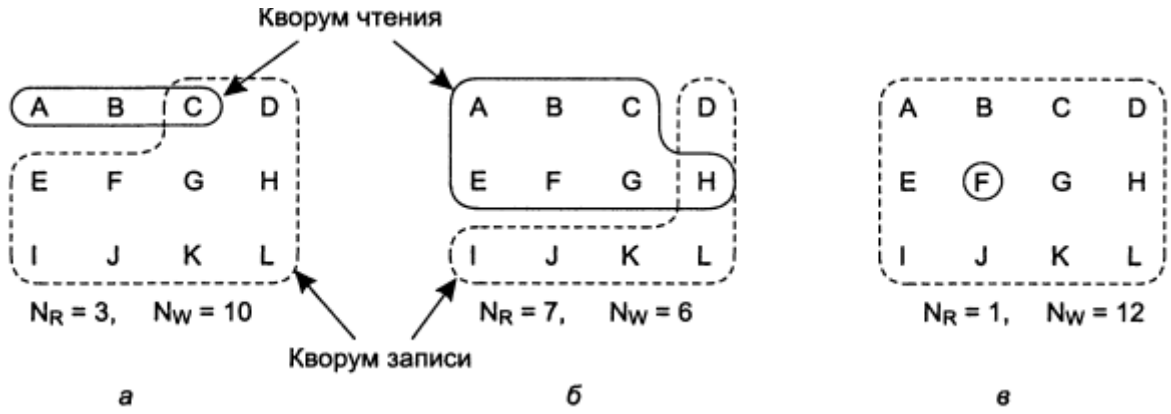


12. **Протоколы на базе реплицируемой записи:** активная репликация: размножению сообщений препятствует общий координатор всех реплик объекта .



13. **Протоколы на базе реплицируемой записи:** протоколы кворума. Данные имеют версию, которая увеличивается при обновлении. Пусть N серверов. При записи уведомляется более чем Nw > N/2 +1 серверов о новой версии данных (кворум записи). При чтении вычисляется новая

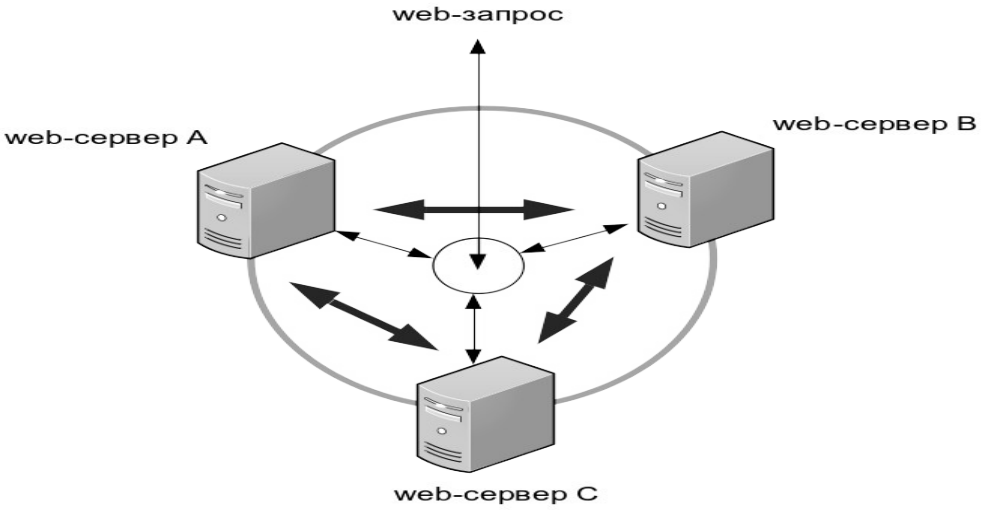
(максимальная) версия путем опроса более Nr > N - Nw серверов. При чтении обновляется локальная версия данных. Рисунок: а – правильно, б – может привести к повторной записи; в)крайний случай (ROWA)

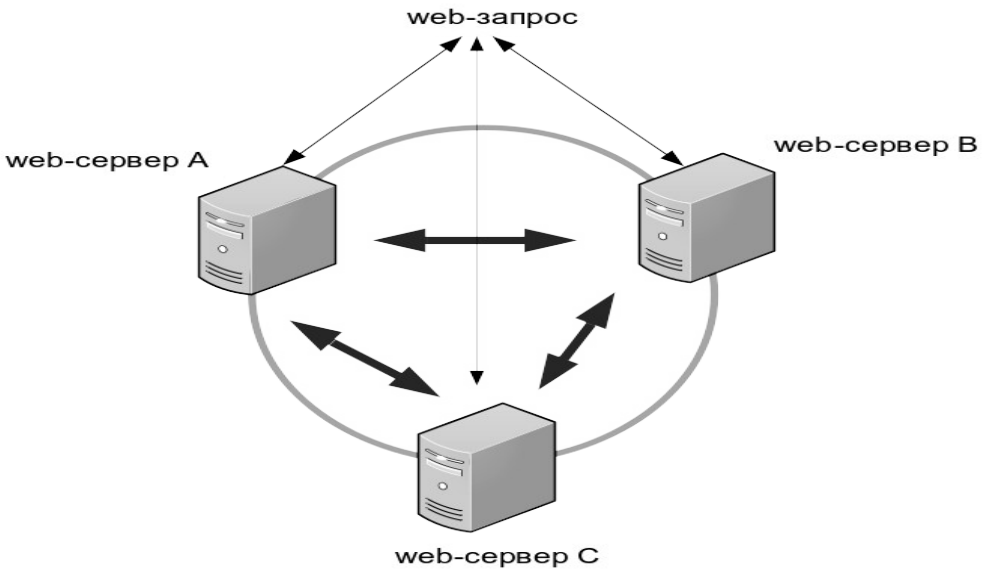


14. **Протоколы согласования кэшей:**

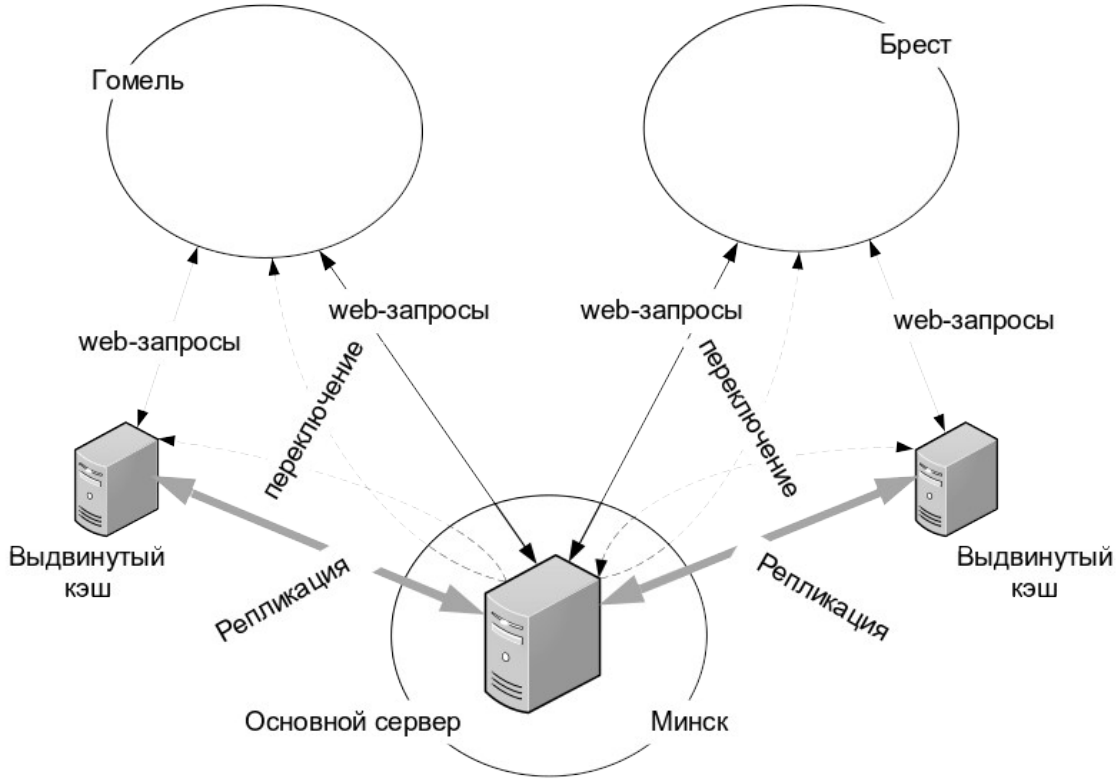
15. **Лабораторная работа 3.** Распределенная транзакция в БД Oracle 12c; механизмы автономных транзакций в Oracle.
16.
17.
18.
19.
20.
21.
22.
23.
24.
25.
26. **Размещение реплик:** постоянные реплики; реплики, инициируемые сервером; реплики, инициируемые клиентом.

27. **Постоянные реплики:** реплики, действующие на постоянной основе в независимости от времени. Пример 1: web-серверы в кластер, кольцевое распределение запроса между серверами в кластере. Пример 2: зеркало – клиент сам выбирает сервер (например, для скачивания).

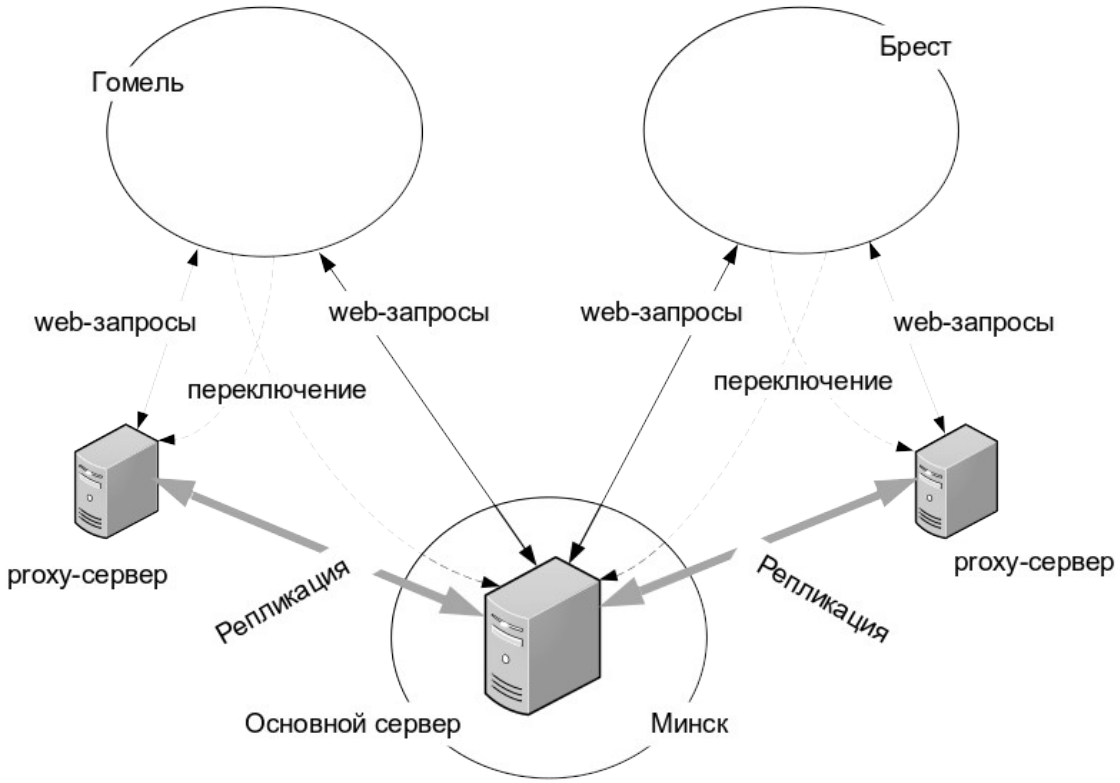




28. **Реплики, инициируемые сервером:** обычно для повышения производительности.

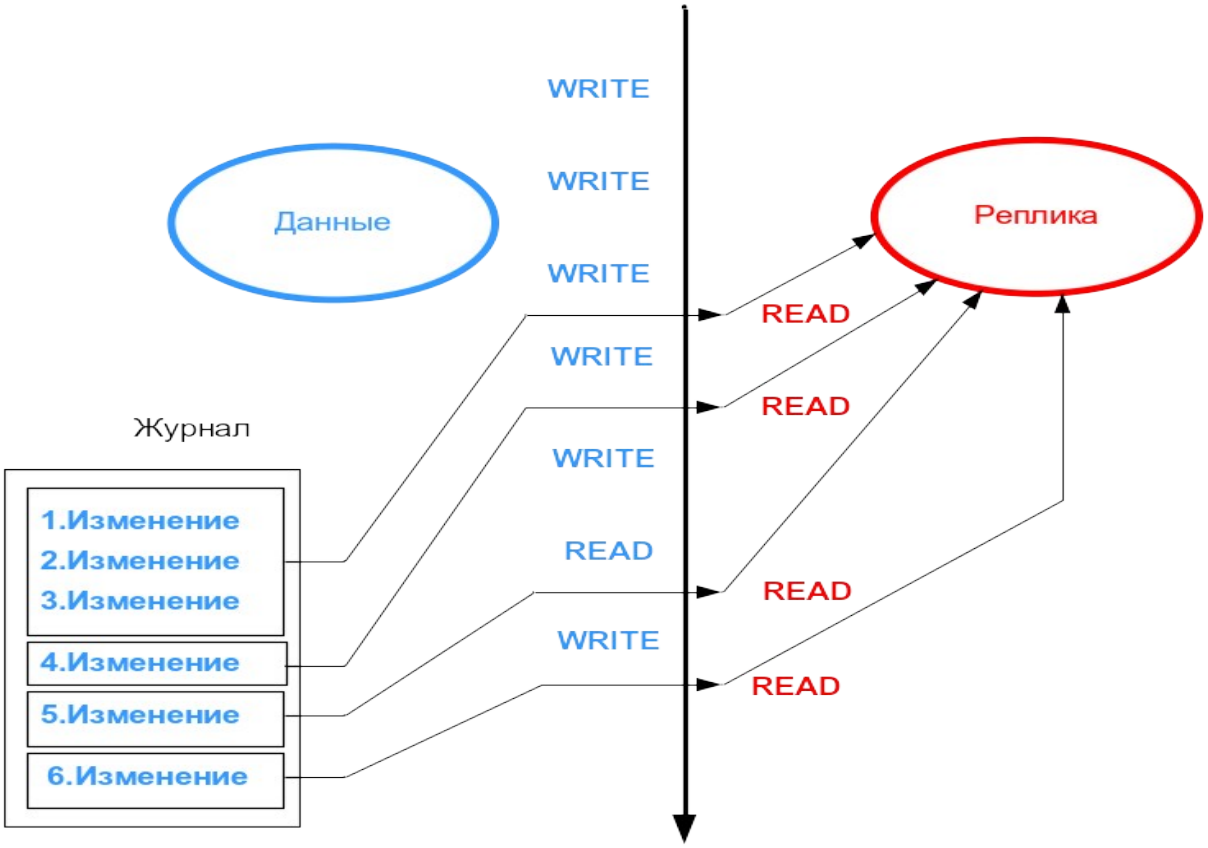


29. **Реплики, инициируемые клиентом:** создание клиентского кэша. Пример 1: кэш создаваемый браузером (обычно get-запросы). Пример 2: proxy-серверы.



Распространение обновлений

30. **Что распространять:** 1) извещения об обновлении; 2) данные от одной копии к другой; 3) операции ко всем копиям.
31. **Извещения об обновлении:** протоколы о несостоятельности; сокращение трафика; применяются, если операции записи преобладают над операциями чтения (следующая запись тех же данных, без промежуточного чтения, делает бесполезным предыдущее изменение данных). Когда и как проводить физические изменения данных зависит от типа поддерживаемой непротиворечивости.
32. **Данные от одной копии к другой:** применяется в том случае, если операции чтения преобладают над операциями записи – вероятность эффективности обновлений становится выше. Вместо прямой пересылки данных обычно ведутся журналы изменений, в которых при конкретной пересылке могут быть объединены несколько операций обновлений.



33. **Операции ко всем копиям (активная репликация):** распространяется сама операция обновления данных; требуется наличие исполнительного механизма (способного повторить операцию обновления данных) и процессорной мощности. Аналог – журнал повтора Oracle: данные – буферный кэш, реплика - БД на диске, обновление – накат журнала.

Продвижение и извлечение

34. **Продвижение (push):** изменение данных влечет автоматическое изменение реплики; используются для поддержки высокого уровня непротиворечивости; чаще всего для постоянных репликацией и репликаций инициируемых сервером; используется для хранилищ с высокой интенсивностью операций чтения; требуется иметь список всех реплик и их состояний; могут продвигаться только уведомления, а потом по запросу клиента (фактически pull) будет обновлена реплика.
35. **Извлекать (pull):** чтение данных влечет запрос на изменение и изменение реплики перед чтением;

используется репликациями, инициируемые клиентом; применяется для хранилищ с высокой интенсивностью операций записи; cash-miss (кэш-промах) – операция чтения застает устаревший кэш; если клиент получает уведомления (при push уведомлений), то это позволяет избежать лишних запросов на обновление реплики.

36. **Аренда (lease):**контракт между сервером (владелец данных) и клиентом (реплика), по которому сервер продвигает (push) изменения в кэш (реплику) клиента; обычно аренда имеет ограниченный строк действия; после его окончания, клиент самостоятельно отсылает запросы (pull) или продлевает аренду.
37. **Гибкая аренда:** 1) на основе частоты изменения данных: чем реже изменяются, тем больше срок аренды; 2) на основе частоты запросов: чем чаще обращения на обновление кэша, тем больше срок аренды; 3) объем пространства памяти на сервере, необходимый для хранения данных о репликациях: чем меньше объем, тем более длительная аренда.

Целевая и групповая рассылка

38. **Целевая рассылка (unicasting):** сервер знает обо всех N конкретных репликациях и каждой репликации высылает соответствующее сообщение (уведомление изменение и пр.); применяется при небольшом ограниченном количестве реплик.
39. **Групповая рассылка (multicasting):** сервер не знает о конкретных репликациях, доставку сообщений берет на себя сеть (например, запустив широковещательное – broadcasting-сообщение); в основном применяется при push-методе изменения многих реплик.

Эпидемические протоколы (epidemic protocols)

40. **Эпидемические протоколы:** основное назначение – минимизировать количество сообщений при репликации данных. Модель: распределенное хранилище – много серверов с локальными данными (репликами); изменения только на одном сервере, от этого сервера распространяются сообщения о репликации.
41. **Терминология:** инфицированный (infective) - сервер, получивший изменения и готовый отсылать сообщения дальше; восприимчивый (susceptible) – не получил изменения, но готовый их получать; очищенный (removed) – получил изменения, но не способен отсылать сообщения.
42. **Антиэнтропия (antientropy):** сервер случайным образом поочередно выбирает другой сервер; три способа обмена сообщениями: сервер продвигает свои обновления; сервер извлекает обновления; два сервера обмениваются обновлениями; если сервер пытается продвинуть свои изменения на другой сервер и обнаружил, что изменение уже проведены раньше, то он с вероятностью 1/k – становится очищенным; алгоритм останавливается когда все сервера очищены.