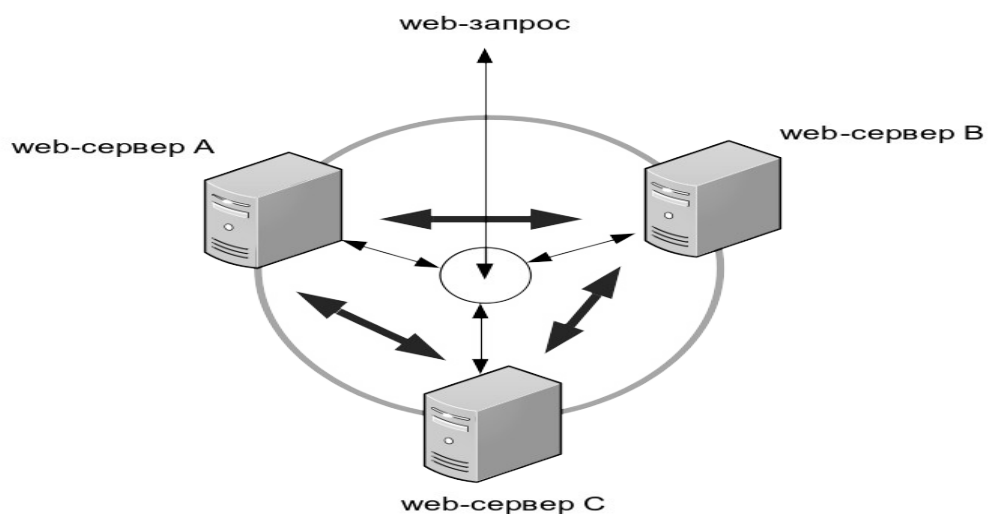


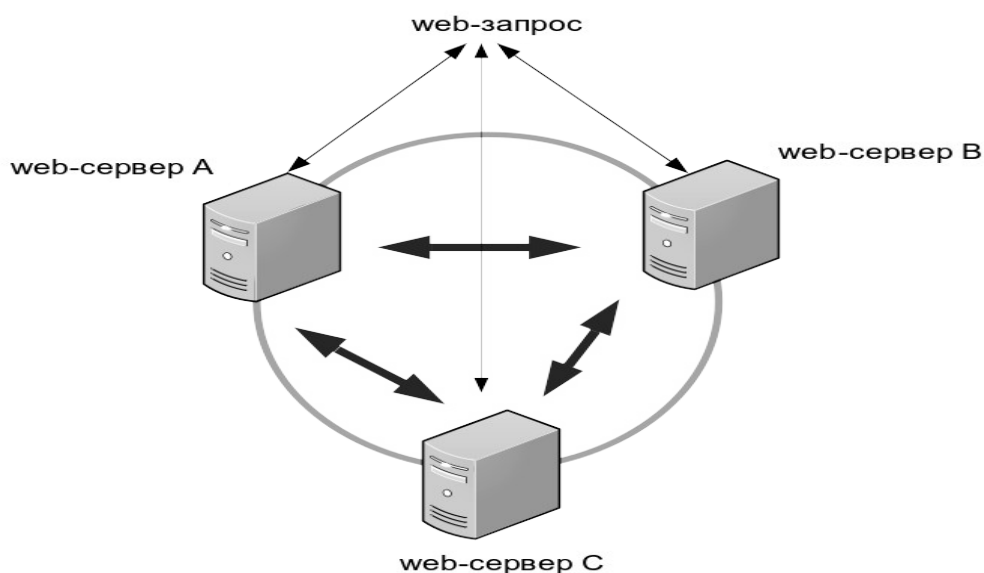
Протоколы распределения обновлений репликации

1. **Размещение реплик:** постоянные реплики; реплики, инициируемые сервером; реплики, инициируемые клиентом.

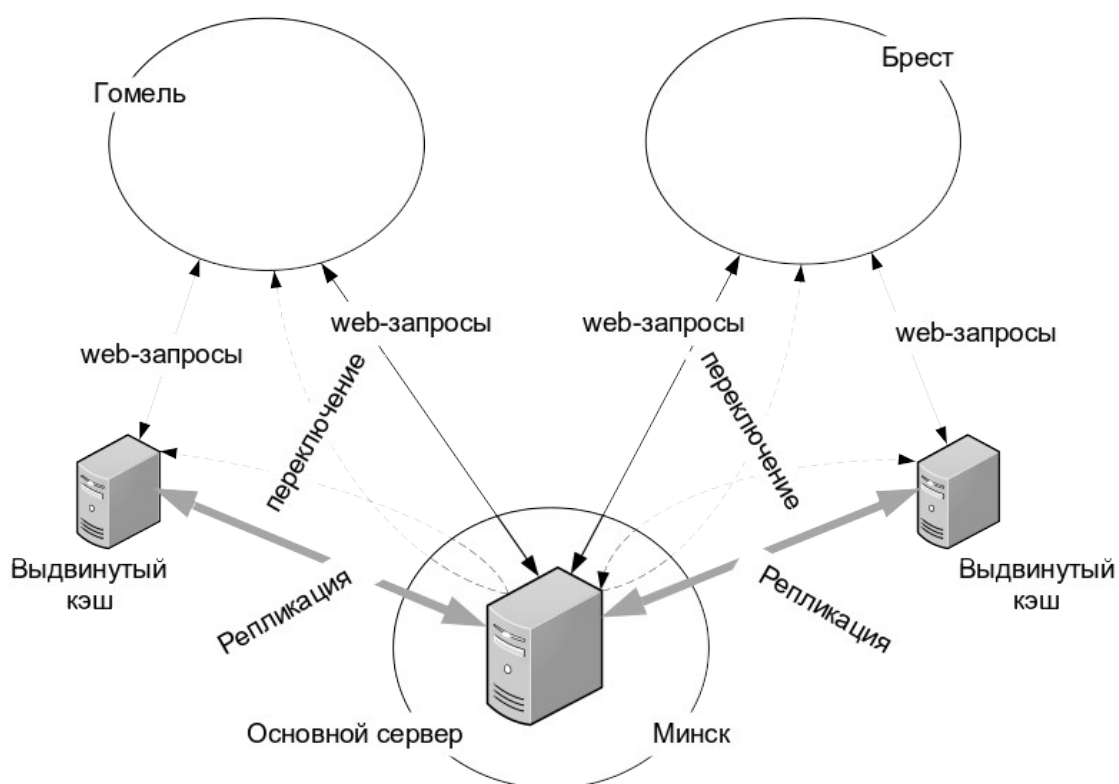


2. **Постоянные реплики:** реплики, действующие на постоянной основе в независимости от времени. Пример 1: web-серверы в кластер, кольцевое распределение запроса между серверами в кластере. Пример 2: зеркало – клиент сам выбирает сервер (например, для скачивания).

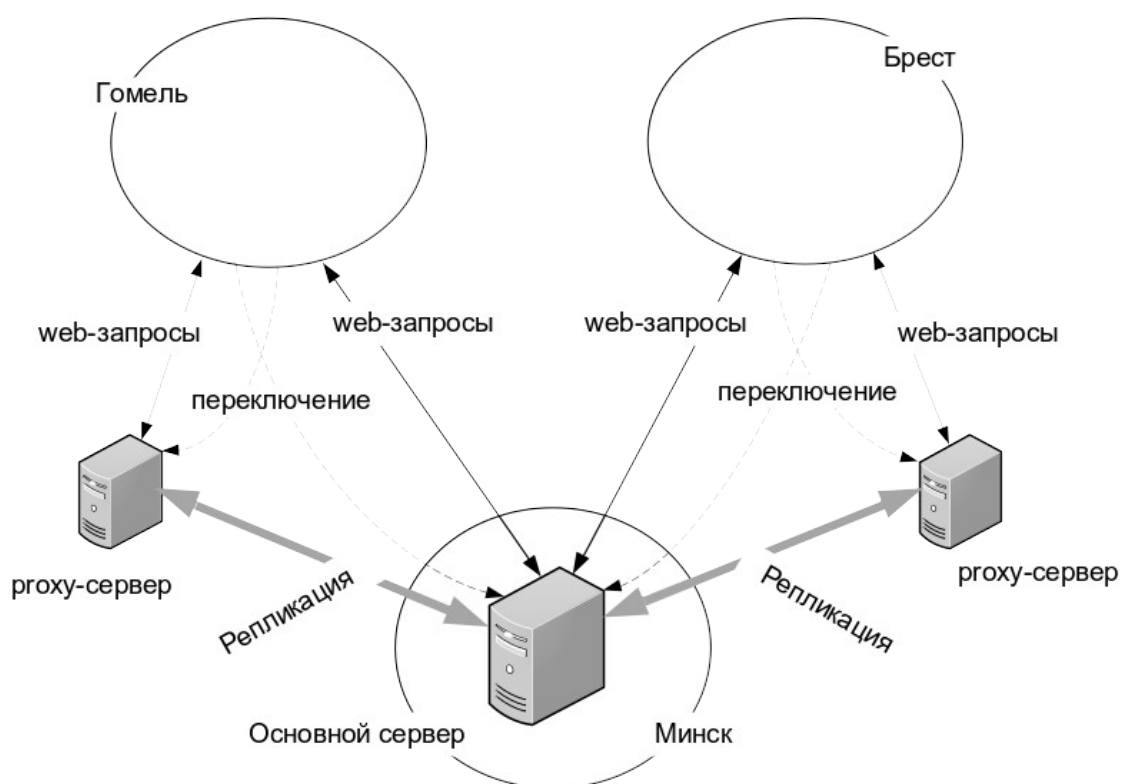




3. **Реплики, инициируемые сервером:** обычно для повышения производительности.

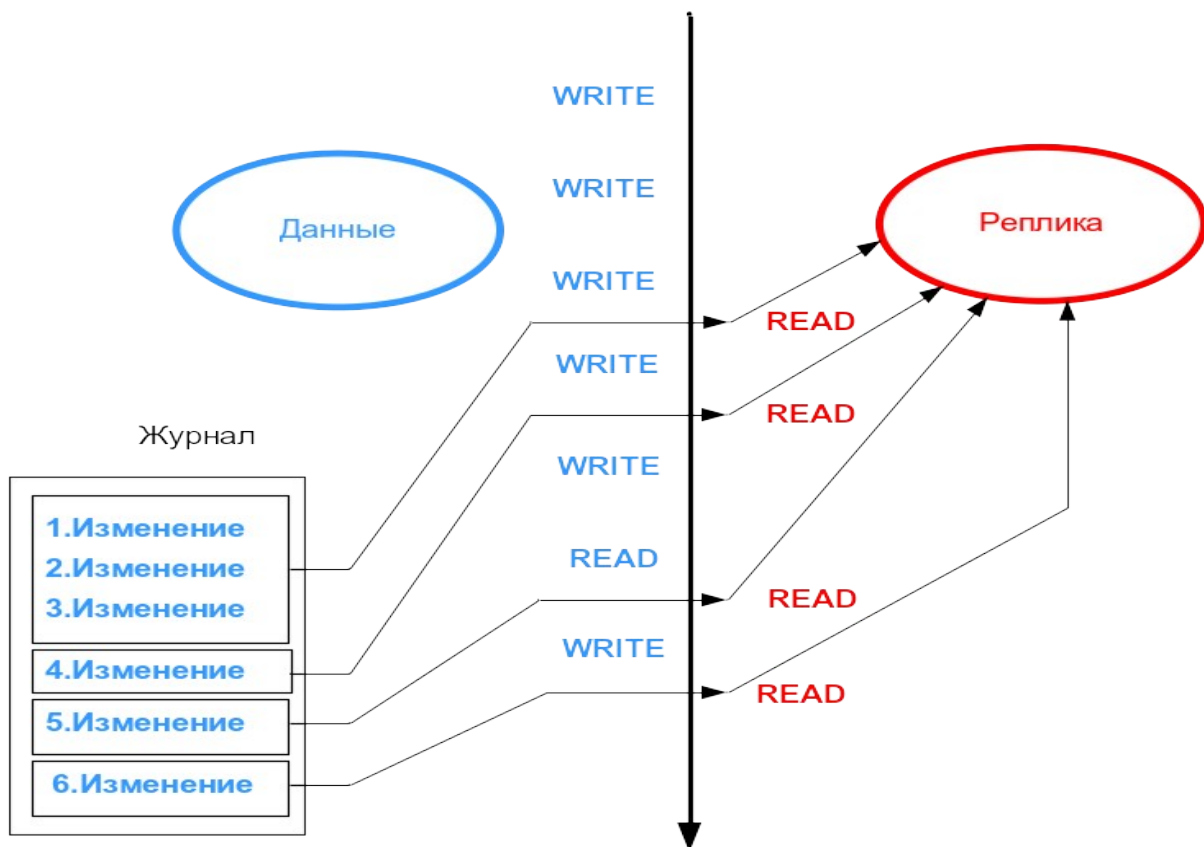


4. **Реплики, инициируемые клиентом:** создание клиентского кэша. Пример 1: кэш создаваемый браузером (обычно get-запросы). Пример 2: проху-серверы.



Распространение обновлений

5. **Что распространять:** 1) извещения об обновлении; 2) данные от одной копии к другой; 3) операции ко всем копиям.
6. **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.
7. **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.



8. **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

9. **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.
10. **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 застаёт устаревший кэш; если клиент получает уведомления (при push уведомлений), то это позволяет избежать лишних запросов на обновление реплики.

11. **Аренда (lease)**: контракт между сервером (владелец данных) и клиентом (реплика), по которому сервер продвигает (push) изменения в кэш (реплику) клиента; обычно аренда имеет ограниченный срок действия; после его окончания, клиент самостоятельно отсылает запросы (pull) или продлевает аренду.
12. **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

13. **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.
14. **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)

15. **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.
16. **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.
17. **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.