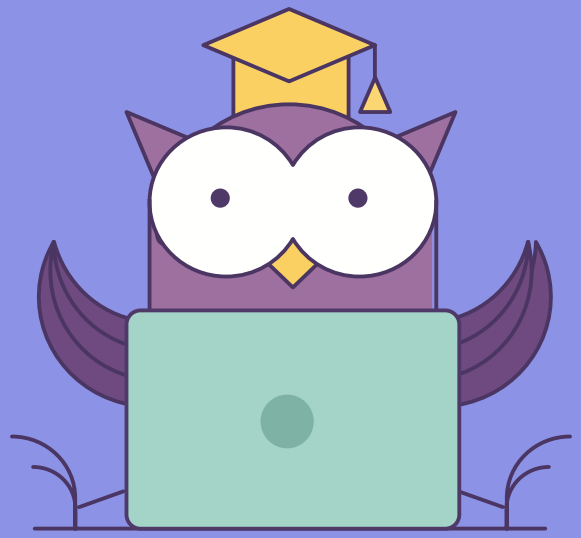
OTUS

PA системы. GOSSIP, CasPAXOS. CRDT структуры

Архитектор ПО

Ярослав Астафьев

Меня хорошо слышно
&& видно?



Напишите в чат, если есть проблемы!

Ставьте + если все хорошо

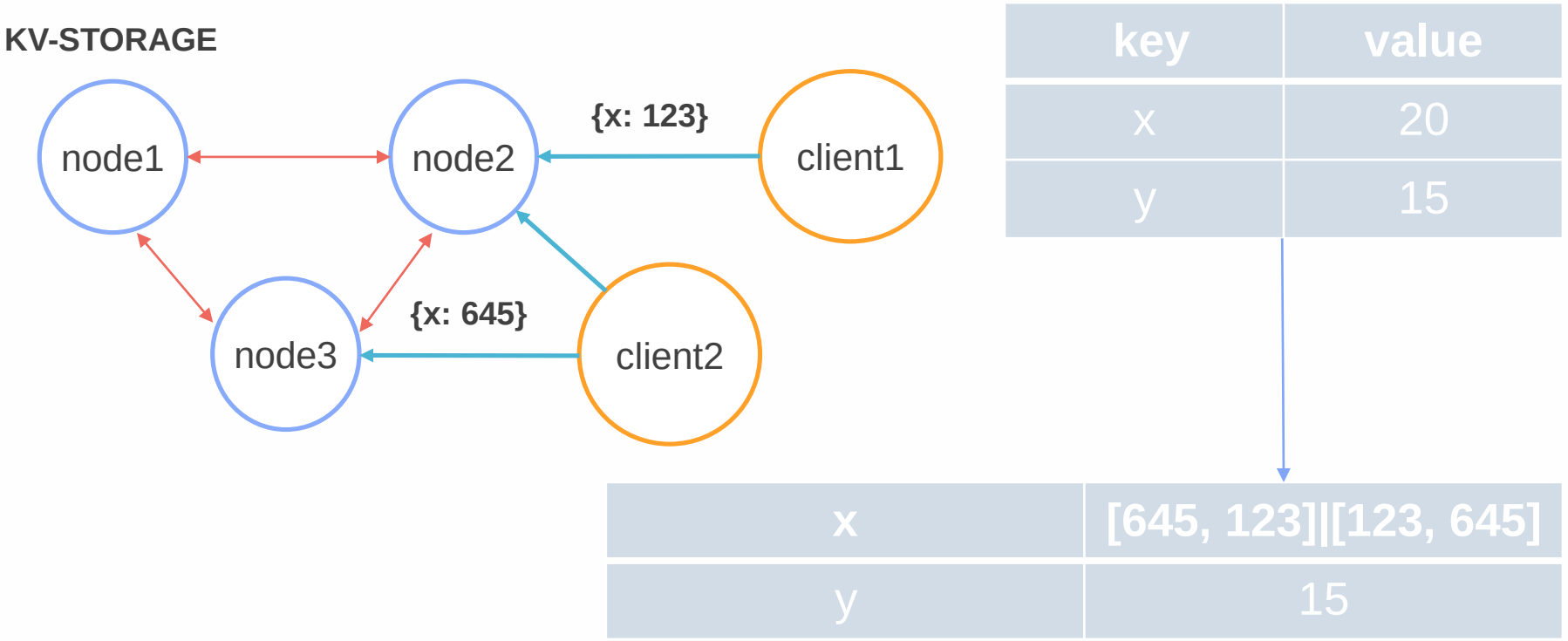
Содержание

- AP системы
- Проблемы master-master репликации
- Gossip: Scuttlebutt
- Репликация без master’а (dynamo-подобные БД)

01

AP системы

AP (equal nodes)



- **Согласование изменений.** Разрешается локально (в большинстве случаев)
- **Изменение** может быть предложено **любым** участником
- **Конфликт** решается, обычно, **слиянием** (merge) или **перезаписью** (last-win)

- Скорость может быть выше (не всегда)
- Нет единой точки принятия изменений
- не может быть гарантирована 100% консистентность

02

Проблемы **master-master** репликации

Применение в 1 ЦОДе – сомнительная идея.

Варианты применения:

- Географическая распределенность
- Hot-standby реплика
- Offline клиенты. Реализовать сложно. CouchDB была сделана специально для этого случая.

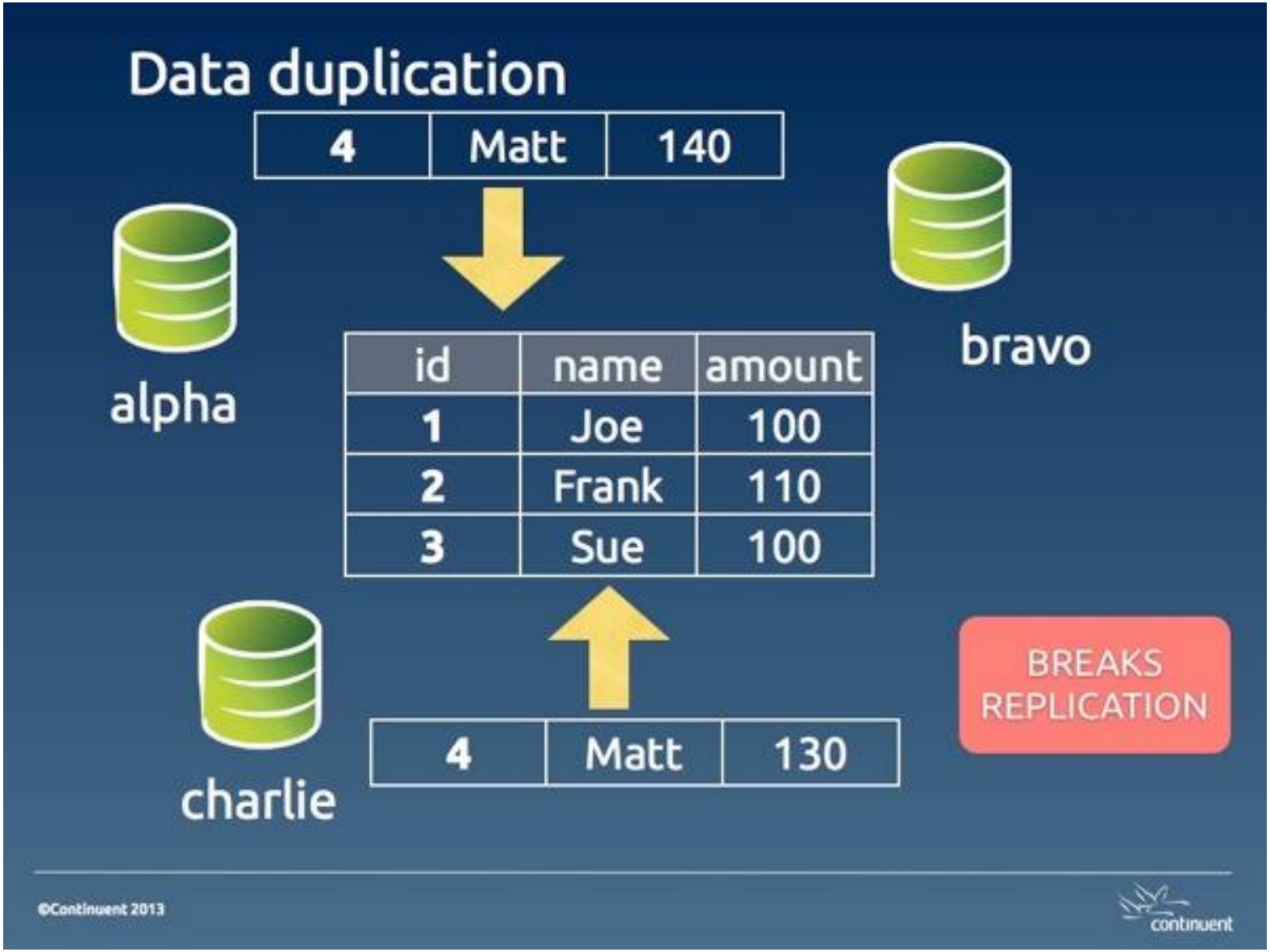
Цена master-master:

- Усложнение логики
- Конфликты

Для географически распределенных ЦОД будут следующие преимущества:

- Производительность
- Устойчивость к уходу ЦОДа
- Устойчивость к проблемам сети

Конфликт в **master-master** репликации



Решение конфликтов

- Избегание конфликтов
- Last wins
- Ранг реплик. Выигрывает запись от старшей реплики.
- Слияние
- Решение конфликтов на клиенте
- Conflict-free replicated data types (CRDT)
- Mergeable persistent data structures

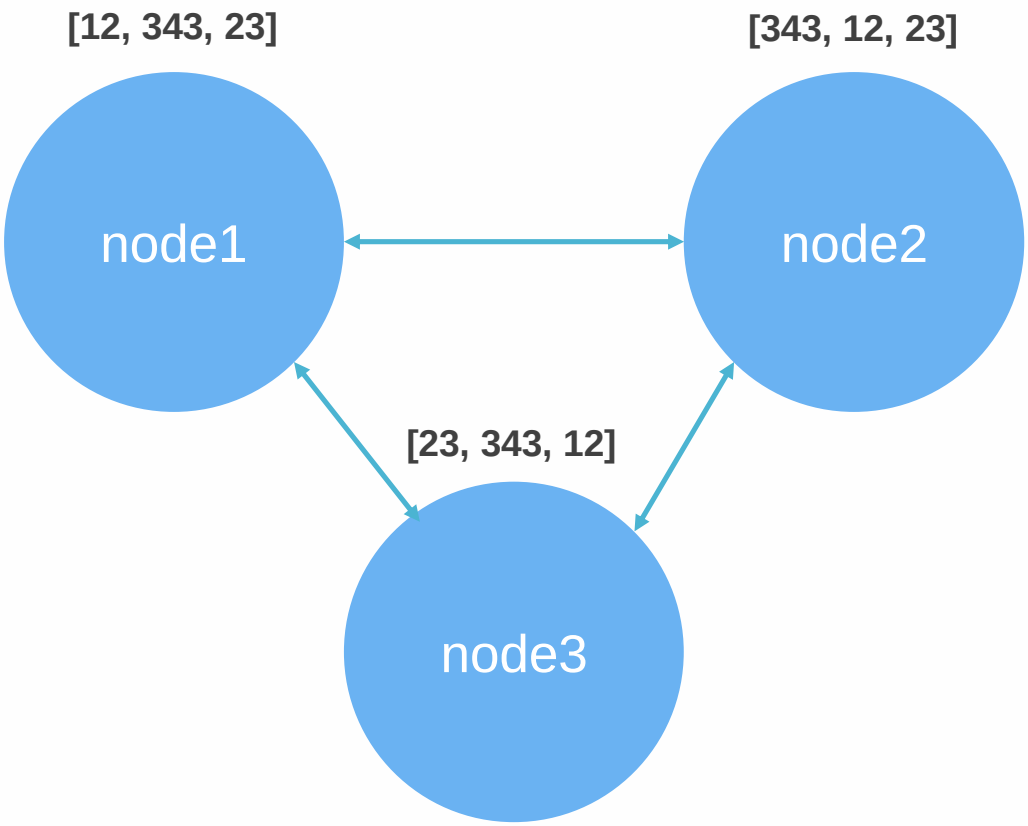
03

Gossip: Scuttlebutt

Свойства

- AP алгоритм
- Синхронная система: использует таймеры
- Скорость работы выше чем у CP алгоритмов в ряде случаев (например, при принятии изменений)

Принцип работы



- Каждая нода может коммитить
- Изменения сливаются локально
- Последовательность может быть разная

Demo

https://awinterman.github.io/simple-scuttle/

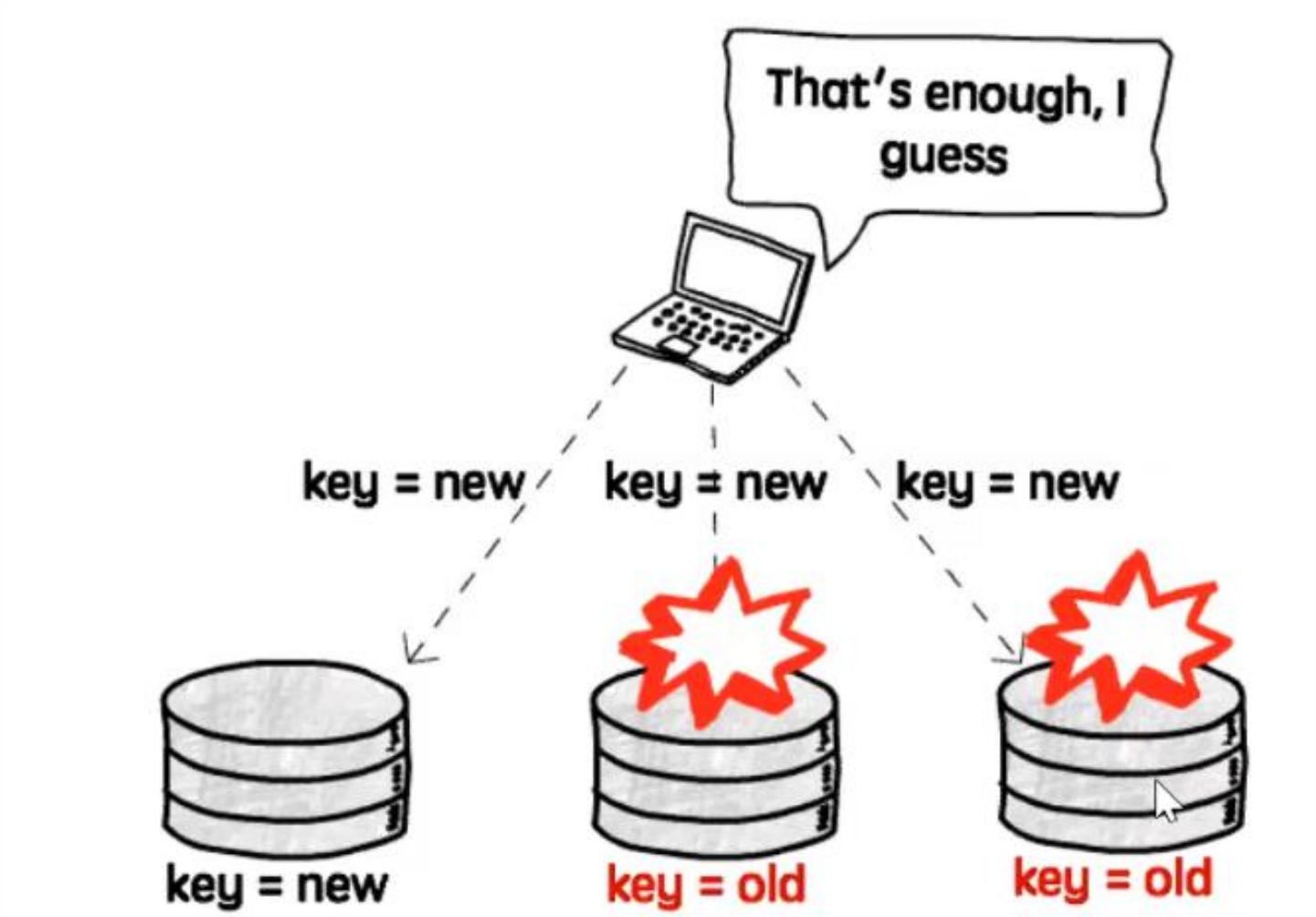
Кто использует?

- Consul
- AWS
- IBM cloud
- Azure
- Google cloud

04

Репликация без **master**’а (**dynamo**-подобные БД)

Репликация без **master**'а



Такая репликация есть в:

- DynamoDB
- Cassandra
- Scylla
- Riak
- Voldemort

Формула для расчета кворума: $w + r > n$

Поддержание консистентности

- Обновление при чтении
- Противодействие энтропии

Проблемы

- Нестрогий кворум. Возможно чтение старых данных при w + r < n
- Нет отката транзакций
- Конфликт записей и потерянные обновления
- Проблемы с линеаризуемостью

Вывод- гарантий нет

Важный вывод- как всегда в интернете бывает 2 исхода: успех и неизвестность

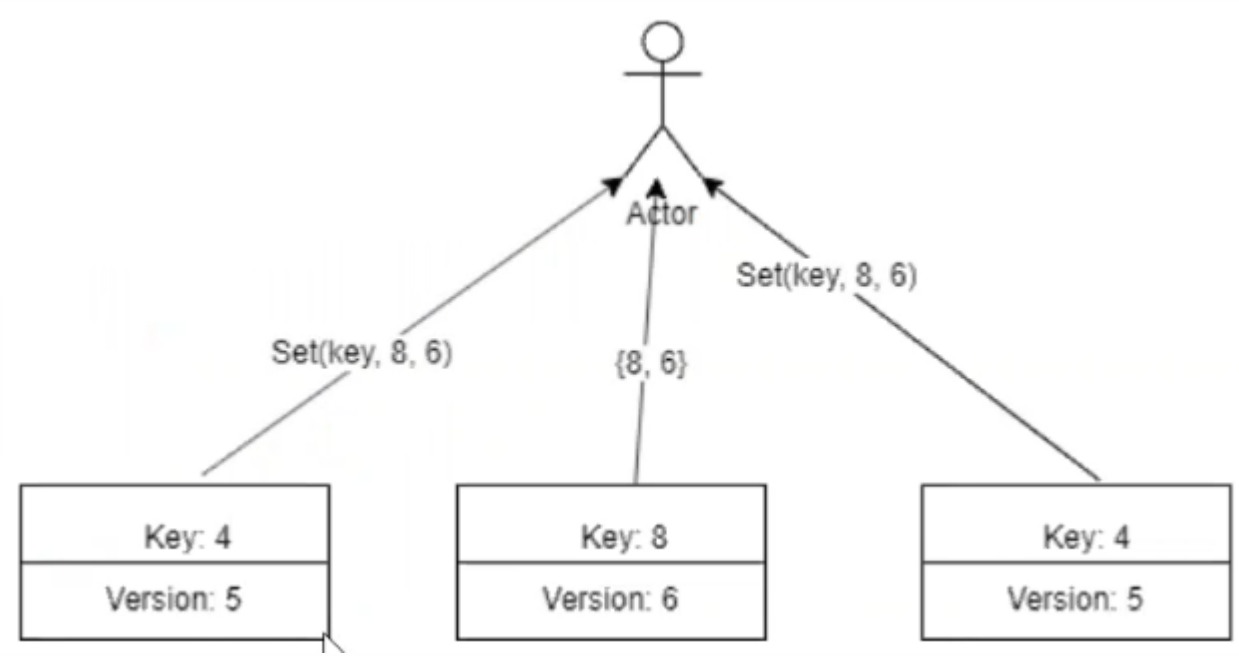
Нестрогий кворум

Что лучше для системы при отсутствии кворума?

- Вернуть ошибку?
- Применить запись без кворума?
- Вернуть устаревшие данные?

Можно писать в узлы, не входящие в n. Это называется нестрогий кворум.

Конфликты


Actor
Set(key, 8, 6)
Set(key, 8, 6)
{8, 6}
Key: 4
Version: 5
Key: 8
Version: 6
Key: 4
Version: 5

Last write wins решение

Выигрывает последняя запись.

Нормально не может работать из-за физической невозможности синхронизации часов.

Единственный нормальный способ- не обновлять ключи.

«Происходит до» решение

Две операции конкуренты тогда и только тогда, когда они независимы.

- Сервер хранит номера версий для всех ключей, увеличивая номер версии всякий раз при выполнении записи значения для этого ключа, и сохраняет новый номер версии вместе с записанным значением.
- При чтении ключа клиентом сервер возвращает все неперезаписанные значения, а также последний номер версии. Клиент должен прочитать ключ перед операцией записи.
- Клиент, записывая значение для ключа, должен включить номер версии из предыдущей операции чтения, а также объединить все полученные при предыдущей операции чтения значения. (Полученный в результате операции записи ответ может быть таким же, как и для чтения, с возвратом всех текущих значений, что позволяет соединять несколько операций записи последовательно, подобно примеру с корзиной заказов.)
- Сервер, получив информацию об операции записи с конкретным номером версии, может перезаписать все значения с этим или более низким номером версии (так как знает, что они все слиты воедино в новом значении), но должен сохранить все значения с более высоким номером версии (поскольку эти значения конкурентны данной входящей операции записи).

Tombstones решение

Решение конфликтов - обязанность клиентов. Удаляемую запись нельзя просто убрать из списка.

Необходимо сделать явную отметку об удалении.

Summary

- AP системы
- Проблемы master-master репликации
- Gossip: Scuttlebutt
- Репликация без master’a (dynamo-подобные БД)

Ваши вопросы?