

NoSQL - новое слово в мире хранилищ данных

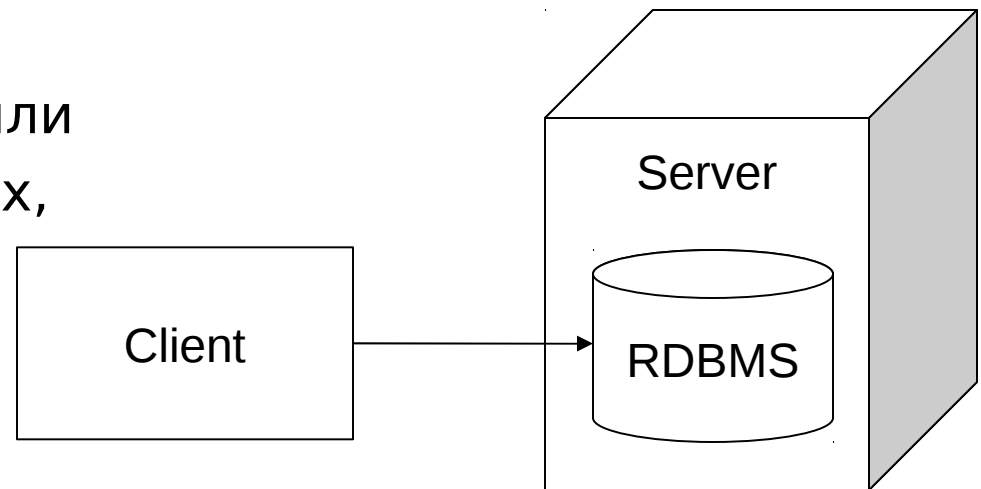
Котусев Святослав Юрьевич



<http://www.devconf.ru>

Рост объемов информации

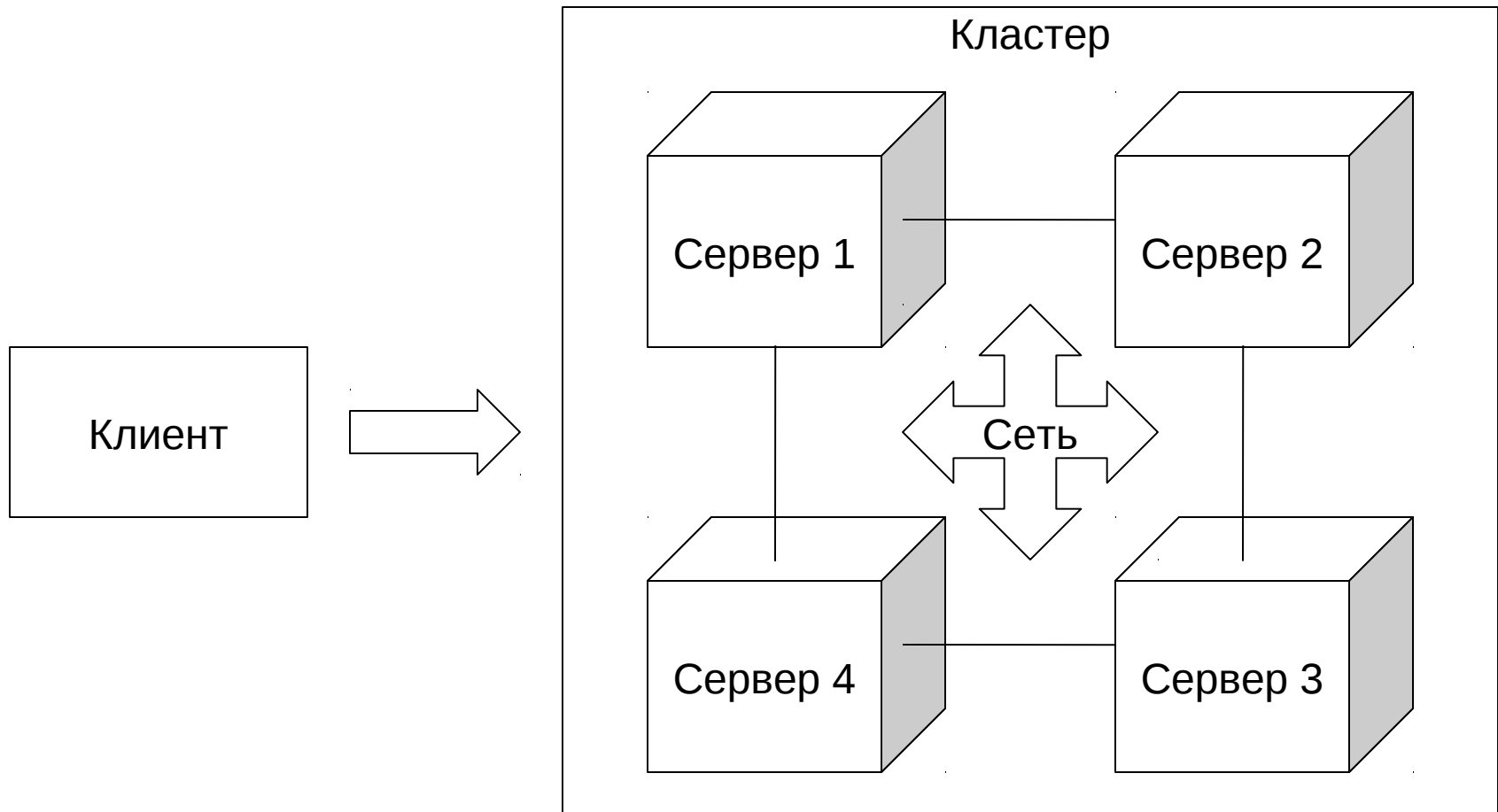
- В последние годы происходит взрывной рост объёмов информации
- Хранить, обрабатывать и предоставлять доступ к ней традиционными способами становится невозможным
- В последние десятилетия доминирующим способом хранения информации были реляционные базы данных, причём вся информация хранилась на одном физическом сервере



Необходимость создания распределённых БД

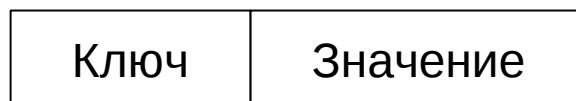
- Попытки создания распределённых реляционных баз данных имели лишь ограниченный успех
- Джоины, сложные запросы, триггеры и внешние ключи не могут быть эффективно реализованы если данных хранятся на разных серверах
- ACID транзакции не могут быть реализованы эффективно в распределённой среде

Распределённые базы данных

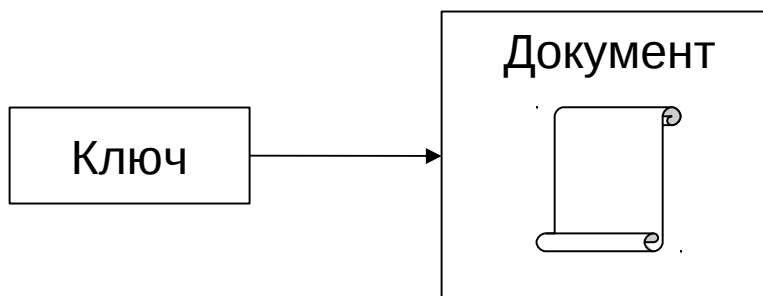


Модели данных в распределённых БД

- Ключ-значение



- Документные



- Табличные

	Столбец 1	Столбец 2	Столбец 3
Ключ 1	Значение 1	Значение 2	
Ключ 2	Значение 3		Значение 4

Требования к распределённым БД

- Целостность данных (Consistency)
- Высокая доступность (Availability)
- Устойчивость к отказам сети (Partition tolerance)

Только два из них могут быть удовлетворены одновременно!

Системы делятся на три вида:

- Системы не гарантирующие целостность данных
- Системы перестающие работать при отказе машины
- Системы перестающие работать при сетевом сбое

Репликация

Эти три вида систем соответствуют трём схемам репликации

- Асинхронная репликация
- Отсутствие репликации
- Синхронная репликация

Eventual Consistency

- Использование асинхронной репликации
- Отслеживании параллельных изменений
- Механизм разрешения конфликтов

Шардинг

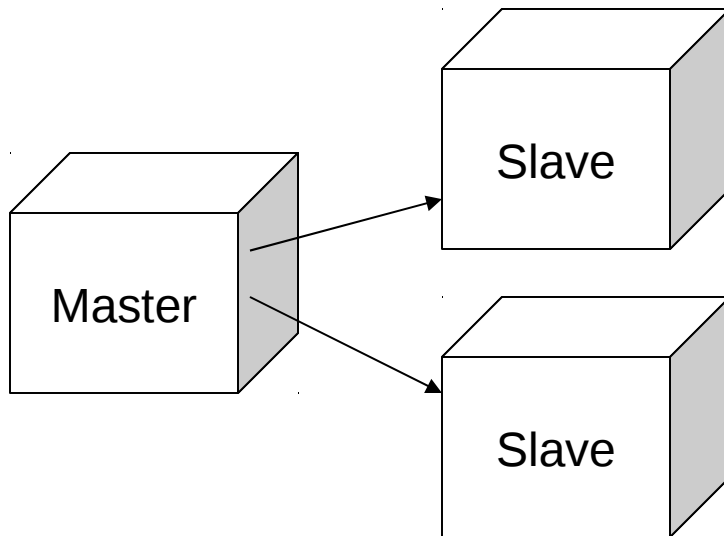
Шардинг это разбиение информации на части и хранение частей на разных физических машинах, он позволяет наращивать ёмкость базы данных путём добавления серверов к кластеру

Способы распределение запросов между серверами

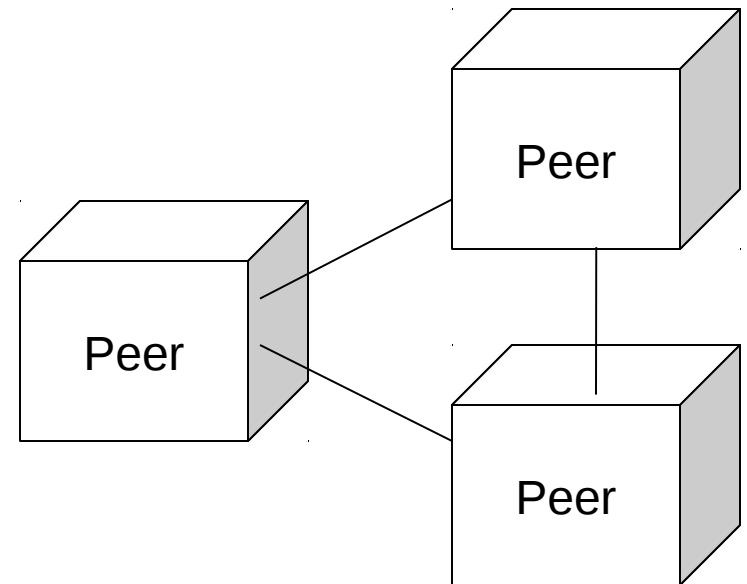
- Через промежуточный проксирующий сервер
- В клиентской библиотеке
- Внутри кластера

Архитектура

Централизованная



Децентрализованная



Существующие NoSQL решения

Обратим внимание на следующие свойства

- Модель данных
- Схема репликации
- Поддержка шардинга
- Централизация архитектуры
- Уникальные свойства и особенности

MongoDB

Модель данных – документная в формате JSON

Репликация – асинхронная

Шардинг – поддерживается через прокси

Архитектура – централизованная master/slave

Уникальные свойства и особенности

- Поиск по полям с использованием индексов
- Агрегирующие функции и MapReduce процедуры
- Хранит большие бинарные файлы через GridFS
- Бинарный протокол с клиентской библиотекой
- Шардинг не является полностью автоматическим

Redis

Модель данных – ключ-значение

Репликация – асинхронная

Шардинг – поддерживается через клиентскую библиотеку

Архитектура – децентрализованная

Уникальные свойства и особенности

- Асинхронная записи данных на диск
- Поддерживает “виртуальную память”
- Поддерживает хранение простых структур данных
- Поддерживает механизм publish/subscribe

Amazon Dynamo, Project Voldemort, Riak

Модель данных – ключ-значение

Репликация – асинхронная с Eventual Consistency

Шардинг – поддерживается через прокси и библиотеку

Архитектура – децентрализованная

Уникальные свойства и особенности

- Поддерживает различные бэкэнды (BerkeleyDB и т.д.)
- Riak поддерживает MapReduce

CouchDB

Модель данных – документная в формате JSON

Репликация – асинхронная с Eventual Consistency

Шардинг – не поддерживается

Архитектура – децентрализованная

Уникальные свойства и особенности

- RESTful интерфейс
- Использует хранимые на сервере Design документы
- Поиск при помощи механизма MapReduce
- Механизм выборки последних изменений базы
- Хранит бинарные файлы как аттачменты к документам
- Шардинг при помощи программы CouchDB Lounge
- Может выступать как веб сервер, сервер приложений и база данных одновременно!

Keyspace

Модель данных – ключ-значение

Репликация – синхронная

Шардинг – не поддерживается

Архитектура – децентрализованная

Уникальные свойства и особенности

- Обеспечивает гарантию целостности данных
- Поддерживает “грязное” чтение
- Атомарные операции (test-and-set и др.)
- Групповые операции чтения и записи
- Операции чтения ключей с одинаковым префиксом
- Использует клиентскую библиотеку

BigTable, Hbase, Hypertable

Модель данных – табличная

Репликация – асинхронная, обеспечивается GFS

Шардинг – поддерживается внутри кластера

Архитектура – централизованная, master/tablet серверы

Уникальные свойства и особенности

- Ячейки хранят версии данных, индексированных по времени
- Обеспечивает локализацию данных для близлежащих ключей
- Транзакции на уровне одной строки таблицы
- Использует GFS и Chubby
- Может быть источником данных для MapReduce

Cassandra

Модель данных – табличная

Репликация – настраиваемая

Шардинг – поддерживается внутри кластера

Архитектура – децентрализованная

Уникальные свойства и особенности

- Таблицы поддерживают столбцы и супер-столбцы
- Изменяемая поддержка целостности данных
- Поддерживает Eventual Consistency без пользовательского механизма разрешения конфликтов
- Создание индексов и поиск по значениям столбцов
- Шардинг с учётом расположения сервера
- Может быть использована как источник данных для Hadoop

Вопросы?