

Реляционные СУБД не сдаются

или почему вам не нужен NoSQL

Николай Шестаков
ведущий программист DP Labs

Немного истории

- Рождение реляционной модели: 70-е
- NoSQL ворвался в нашу жизнь 2000-е

Немного истории

- Рождение реляционной модели: 70-е
- NoSQL ворвался в нашу жизнь 2000-е
 - И это ещё даже не модель данных

Так что же такое NoSQL?

- «Ключ-Значение» (Redis, MemcacheDB)
- Хранилище семейств колонок (Cassandra, HBase)
 - «Ключ-Значение», где значения – кортежи с заданным набором атрибутов (столбцов)
- Документные (mongoDB, RavenDB, Couchbase)
 - «Ключ-Значение», где значения – слабоструктурированные данные (XML, JSON...)
- Графы (InfiniteGraph, FlockDB)
 - Это вообще из другой оперы

А что такое вообще база данных?

А что такое вообще база данных?

- Базой данных является представленная в объективной форме совокупность самостоятельных материалов (статей, расчётов, нормативных актов, судебных решений и иных подобных материалов), систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины (ЭВМ)

А что такое вообще база данных?

- Данные структурированы!
 - Обеспечивает возможность поиска (язык запросов)
 - Разработчик работает на высоком уровне абстракции (абстрагирование – борьба со сложностью)
 - Ограничения целостности
 - Хорошая структурированность описывает законы реального мира без всякого программирования!

А что такое вообще база данных?

- NoSQL
 - ? Обеспечивает возможность поиска (язык запросов)
 - ? Разработчик работает на высоком уровне абстракции (абстрагирование – борьба со сложностью)
 - ??? Ограничения целостности
 - ??? Хорошая структурированность описывает законы реального мира без всякого программирования!

А что такое вообще база данных?

- NoSQL
 - ? Обеспечивает возможность поиска (язык запросов)
 - ? Разработчик работает на высоком уровне абстракции (абстрагирование – борьба со сложностью)
 - ??? Ограничения целостности
 - ??? Хорошая структурированность описывает законы реального мира без всякого программирования!
- NoSQL должен быть очень хорош, чтобы ради него отказаться от всего перечисленного!

Так чем же хорош NoSQL?

- Возможность обработки неструктурированных данных
- Горизонтальная масштабируемость (шардинг и Map-Reduce)
- Высокая скорость работы за счёт отсутствия JOIN
- Более быстрая разработка за счёт отсутствия ORM
- Инвесторы охотнее дают деньги проектам с NoSQL (O RLY?)

Неструктурированные данные

- Мы можем сохранить в нашу БД JSON!
- А потом делать запросы с фильтрами по JSON-атрибутам
- Если у СУБД есть Web-API, то можно запрашивать объекты напрямую из javascript!

Неструктурированные данные

- А почему наши данные не структурированы?
 - Источник данных такой – ничего не поделаешь
 - Нам лень продумывать схему, давайте зафигачим так, быстро сделаем сайт-прототип, заказчик будет в восторге, а схему додумаем потом
- Без схемы вы ходите по болоту: на клиент может прилететь что угодно. Как обрабатывать все возможные ошибки целостности данных?
- БД примет от вас любые данные, и не надейтесь на ваши валидаторы. Потом на продакшене заказчик не сможет правильно деньги посчитать – будете руками ошибки исправлять

Неструктурированные данные

Вообще говоря, слабоструктурированные данные можно хранить в РБД

- XML в MS SQL Server, Oracle
- JSON в MySql
- JSON в PostgreSQL
 - Запросы по атрибутам!
 - Индексы!!
 - Согласно некоторым тестам, уделывает MongoDB!!!

Шардинг и Map-Reduce

- Если данных очень много и вертикальное масштабирование не помогает
- Распараллеливание запросов с агрегацией по нескольким машинам
- Eventual Consistency – разумный компромисс ради высокой скорости

Шардинг и Map-Reduce

- А вы уже переросли возможности вертикального масштабирования?
 - Если ваш молодой человек дарит вам диск Аркадия Укупника, ваш молодой человек – Аркадий Укупник
 - Если вы делаете фейсбук или ВКонтакте, вы – Марк Цукерберг или Павел Дуров
 - Avito сидят на PostgreSQL, вы круче Avito?
- БД проектируется под запрос
- БД получается денормализованной

Шардинг и Map-Reduce

- Распределённые решения существуют и для РБД
 - Правда, они медленнее
 - Они достаточно сложны в реализации и администрировании
 - Полноценная транзакционность в них тоже не гарантирована
- РСУБД предоставляют много возможностей повысить эффективность за счёт денормализации, обеспечивая целостность

Высокая скорость за счёт отсутствия JOIN

- Вы так же можете не использовать JOIN в РБД (хотя это и глупо)
- У вас проблема с JOIN?
 - Может быть, стоит освежить знания по использованию индексов?
 - Может быть, просто ваша СУБД – ~~MySQL~~ отстой?
 - ОК, уговорили, давайте денормализуем
- Современные РСУБД за 40 лет научились делать JOIN
- При этом, для выполнения нового запроса с JOIN вам не надо переделывать структуру базы
- Если вы рассматриваете Map-Reduce как средство аналитики (для сводных отчётов), не лучше ли посмотреть в сторону OLAP?

Быстрая разработка за счёт отсутствия ORM

- Работали с Legacy-кодом из 90-х на Delphi или Visual Basic? Это результат быстрой разработки
- Такая быстрая разработка закладывает технический долг, размер которого находится в экспоненциальной зависимости от сложности концептуальной схемы системы
- Для прототипа сойдёт (о, круто, почти всё работает, давайте в бету, у нас же agile!)
- Для простеньких приложений тоже норм. Особенно если вы уже вложили 10 человекомесяцев в самописный фреймворк, ради экономии на ORM («Лучше день потерять, потом за 10 минут долететь!»)

Не забываем про другие плюсы РСУБД

- Ограничения целостности
- ACID
- Разграничение прав доступа
- SQL знают все, даже некоторые бизнес-аналитики
- Как там с NoSQL спецами на рынке труда?

РСУБД тоже не стоят на месте

- В 90-е был большой интерес к объектно-ориентированным БД. Реляционные системы взяли лучшее из ОО-модели. И победили их!
- PostgreSQL (и за ним другие) берут лучшее из NoSQL (обработка слабоструктурированных данных, шардинг)
- В большинстве проектов без РБД всё равно не обойтись, поэтому выбор обычно между РБД или РБД+NoSQL
- Возможно, мы ещё увидим, как NoSQL будут вытесняться РБД с NoSQL возможностями

Что делать?

- Используйте NoSQL

Что делать?

- Используйте NoSQL
- Только когда у вас нет другого выхода

Что делать?

- Используйте NoSQL
- Только когда у вас нет другого выхода
- И вы точно знаете, что делаете

Fight!

- Всё не так однозначно
- Для каждой задачи – свой инструмент
- Не существует серебряной пули
- Забивать гвозди микроскопом
- Где бы применить мой новый блестящий молоток?
- ...