

# Хранение данных в приложении

Станислав Протасов, 2014

# Сначала вопросы

- Технические характеристики данных:
  - Количество данных (KB, MB, GB, TB, ...)
    - а также наши возможности хранения
  - Качественные характеристики
    - средний размер единицы хранения
    - оценка диапазонов размеров
    - тип данных
      - сжимаемость
      - mime-type
      - локализуемость и кодировки
      - структурированность (xml, json, yaml, aligned, ...)

# Сначала вопросы

- Динамика данных:
  - Количественная (сколько будет через 3 месяца, год, 5 лет)
  - Качественная
    - Будет ли изменяться размер единицы хранения?
    - Будет ли меняться структура данных?

# Сначала вопросы

- Важность и назначение данных
  - Сервисные данные (зачем мы их храним?)
  - Бизнес-критичные данные (в т.ч. отчёты об ошибках и СЕР)
  - Данные пользователей и о пользователях
  - Разделяемые или публичные данные (внешняя модерация)

# Сначала вопросы

- Операции с данными
  - Данные бывают промежуточные и «окончательные»
  - Предполагаемые операции
    - 90/9/1
    - Сложная аналитика
      - языки и конструкторы запросов
      - OLAP
  - Бэкапы и восстановление из бэкапов
  - Публикация данных
  - Определение пользователей данных, их потребностей и способа доступа

# Из чего выбирать?

- **RDBMS** (тысячи их)
  - UI
  - Широта и качество поддержки в языках
  - Покрытие потребностей (даже в Enterprise бывает достаточно SQL Express) вроде репликации, шардинга, распределённых транзакций и т.п.
  - \* in-memory
- **NoSQL**
  - Key-Value (BerkleyDB)
  - Distributed Tabular (MapReduce: HBase, ...) и не только Tabular
  - Object-oriented (db4o, ZODB, ...)
  - Document-oriented (OO+. MongoDB)
  - \* in-memory
- **File System**
  - **Personal** RAID/SSD/Tape/...
  - Personal and Public **Cloud FS** (PCS, Amazon, Azure, ...)
- **DIY**

# Какие бывают данные

- **Семантически**

- Blobs [FS, Cloud Storage]
  - static content [CDN]
  - dynamic content [FS, (geo) sharding]
  - raw content [any]
- Objects, Documents [RDBMS+ORM, OO/DO NoSQL]
- Tables [RDBMS, Tabular NoSQL, CSV]
- Events [RDBMS+OLAP, Tabular NoSQL]
- Logs/Backups [FS, Cloud Storage]

# Какие бывают данные

- **Размер данных** влияет на следующие характеристики:
  - объём и тип занимаемого дискового пространства (см. OLAP)
    - Выбирать между ценой за своё железо и облачные эластичные хранилища
    - *При 64K секторах хранение 1K-сущностей становится невыгодным*
  - трафик (чтение-запись) и его характер
    - Стоимость трафика (например, условия VPS), настройки сетевого оборудования
  - характеристики дисков с точки зрения скорости и надёжности
    - Серверное железо + RAID vs Public/Personal Cloud



# Какие бывают данные

- **По типу доступа**
  - Последовательный, поточный или произвольный доступ
    - Хэш-таблицы и in-memory
    - Ленты и файлы
  - Доступ преимущественно по чтению/записи
    - Cassandra (write) vs Redis (read)
    - Входящий vs Исходящий трафик (Cloud)

# Действия с данными

- **Запись**
  - Крупные блоки [FS, Cloud]
  - Мелкая и частая запись [Key-Value, Indexed RDBMS]
- **Чтение**
  - Одинаковые запросы
    - Синтаксически [SQL prepared statements]
    - Идентичные [cache – программный, диска, DB] – Автоматизация
  - Совсем разные запросы
    - Язык запросов конструктор запросов
- **Конкурентность**
  - **Монопользовательский режим**: embedded DB, слабые требования к транзакциям
  - **Многопользовательский режим**
    - ACID
    - Механизмы удалённого доступа

# Владение данными

- **Монопольное владение**
  - Ответственность перед клиентами
- **Разделяемые (с разной долей ответственности)**
  - Вопросы интеграции
    - Интерфейсы
    - Выгрузки
  - Вопросы формата
  - Вопросы поддержки
- **Общий доступ**
  - Создание сервисов
  - Разработка и версионирование API
  - Вопросы экспорта и импорта данных

# Сжатие данных

- **Экономия** дискового пространства ведёт к **падению скорости** чтения-записи:
  - Отказ от индексов
  - Отказ от выравнивания (compressed tables)
  - Сжатие данных (PPMD, LZMA, компрессия в MSSQL)
  - Database shrinking
  - Compressed folders (практика - 50% на тексте)
  - Sharding

# Динамика данных

- **Борьба с ростом таблиц**
  - Выгрузка, архивирование, удаление или агрегация исторических данных
- **Создание и хранение бэкапов**
  - Контроль за свободным местом в хранилище бэкапов
  - Полный vs инкрементальный: место против надёжности
  - Обязательная фиксация сферы ответственности

# Доступность данных

- Вопросы к репликации данных
- Вопросы к надёжности и «портфолио» DBMS, железа, облачных сервисов
- Вопросы транзакций и грязного чтения

# Выводы

- При создании новой системы важно задать себе вопросы по характеру хранимых данных, прежде чем выбирать хранилище
- Принципиально может быть использован любой тип хранилища. Важно найти компромисс для требований решаемой задачи. RDBMS уже давно не панацея
- Кроме места, следует обращать внимание на скорость обращения, объём трафика и надёжность сервиса
- Важно оценивать не только текущий объём данных, но и долгосрочную динамику «в худшем случае»