ХРАНИЛИЩА ДАННЫХ НА СЛУЖБЕ BI

Александр Крашенинников · Алексей Еремихин

badoo_tech

Алексей
Еремихин

Александр
Крашенинников

Руководим разработкой BI в Badoo

badoo_tech


>550 000 000
people all over the world
use our apps

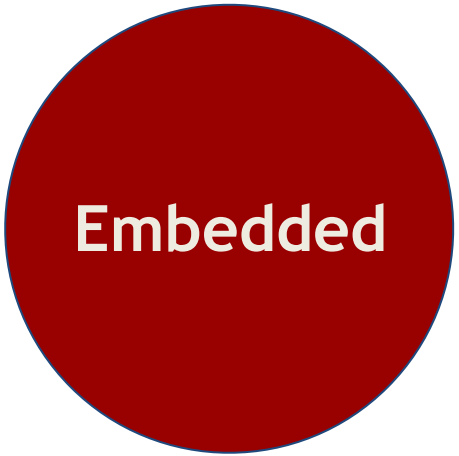
План доклада

- Что такое BI
- Хранилища данных
- Задача ETL
- Обработка потока событий
- Cross-database bridge

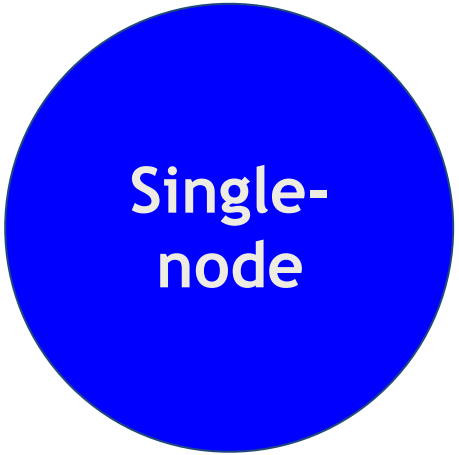
Что такое BI

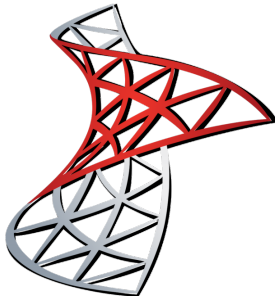
- Люди делающие графики из данных
- BI обеспечивает работу с данными
- А данные в хранилищах

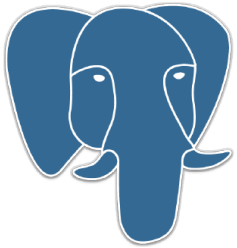
Хранилища данных: Embedded



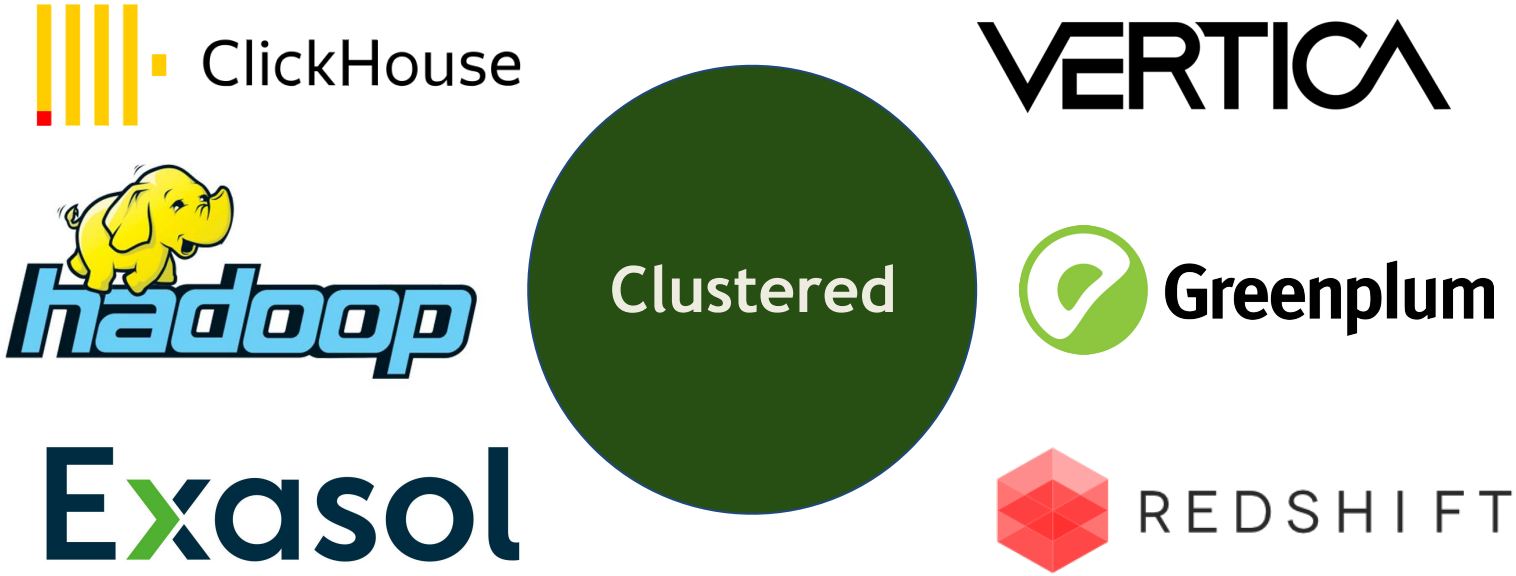
Хранилища данных: Single-node







Хранилища данных: Clustered

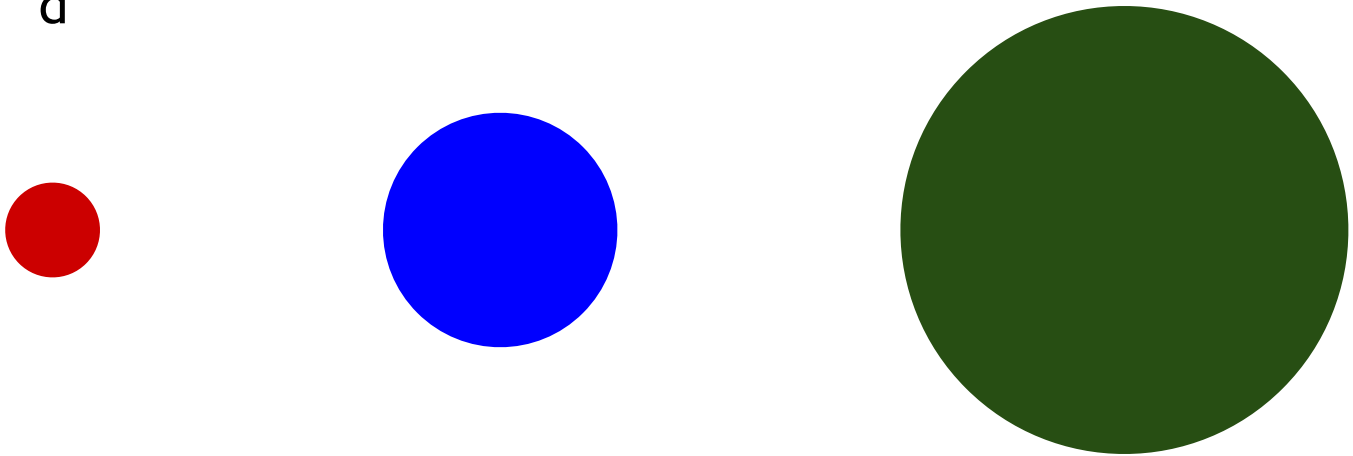


Размеры и сложность баз

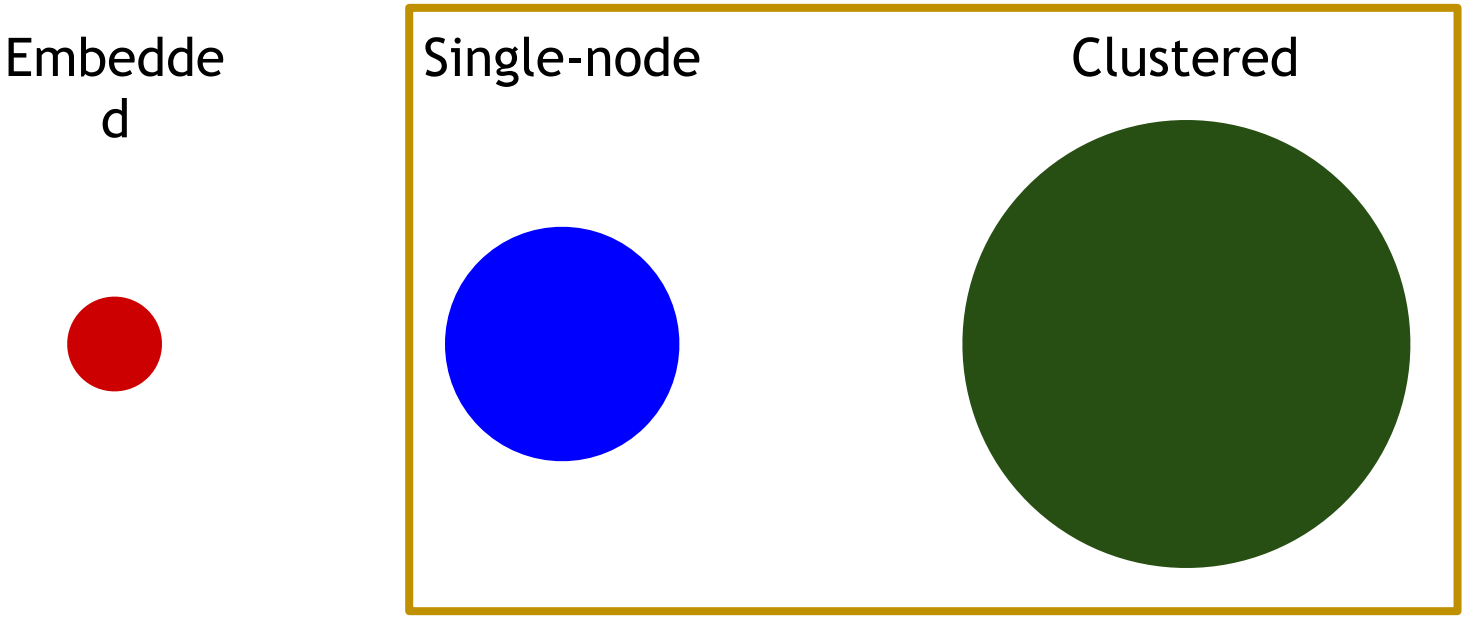
Embedded

Single-node

Clustered



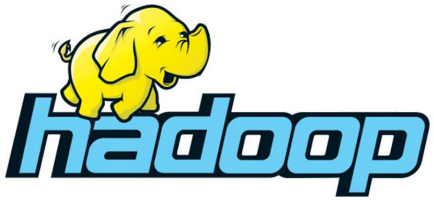
Размеры и сложность баз



Поговорим об этих базах

Single-node

Clustered



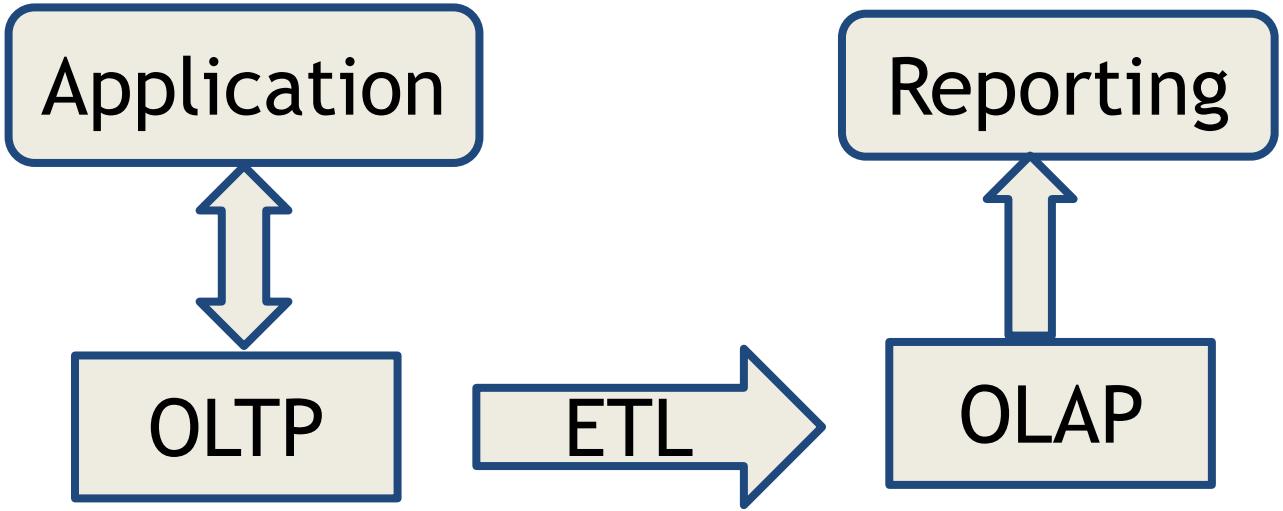
Классический ETL

Extract

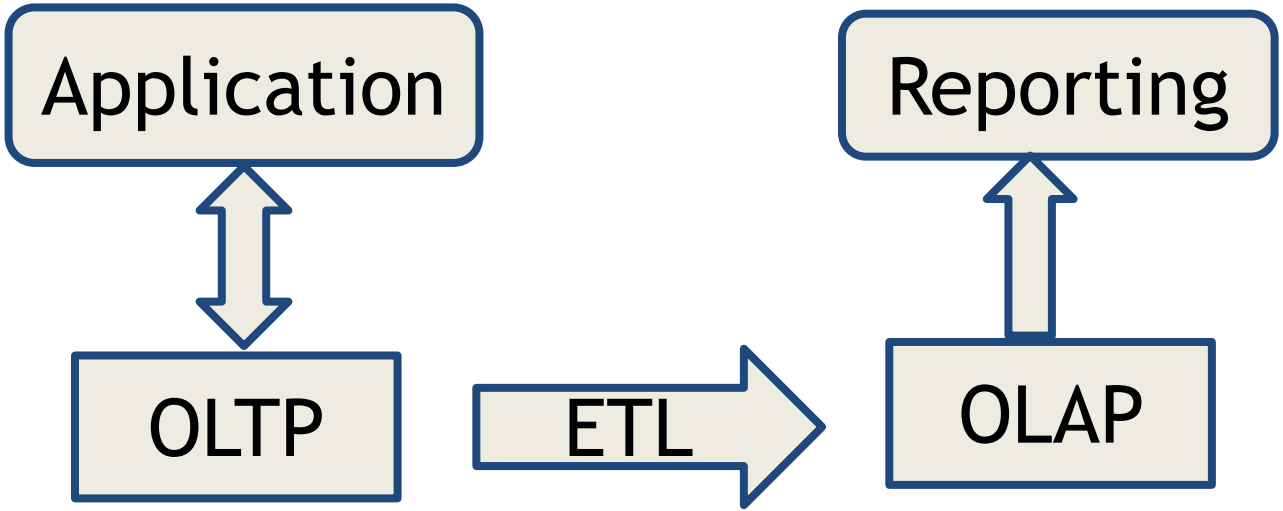
Transform

Load

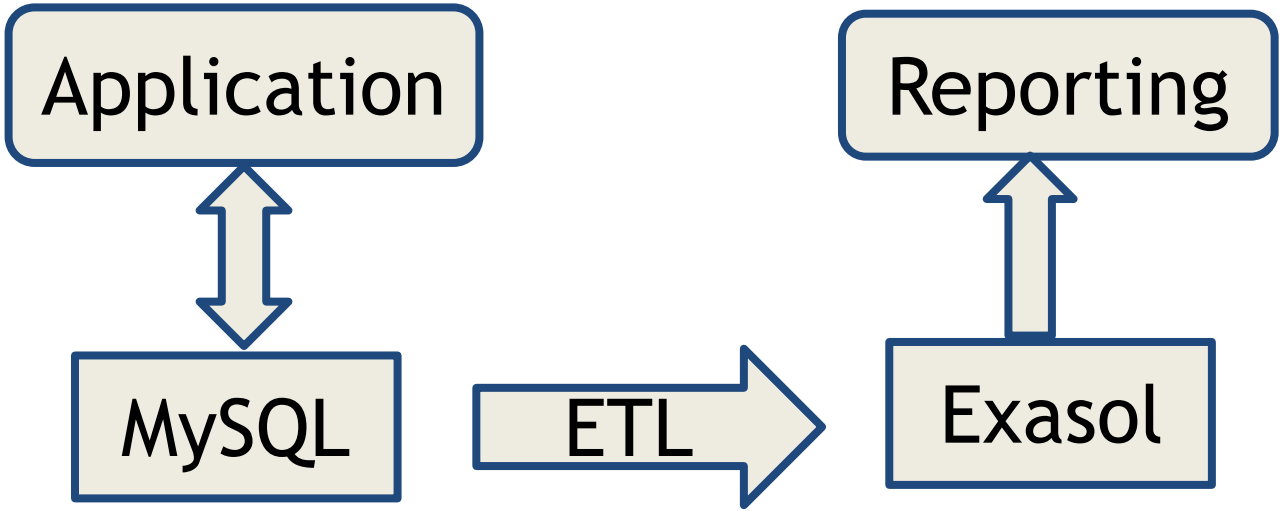
Что такое ETL



Непонятно!

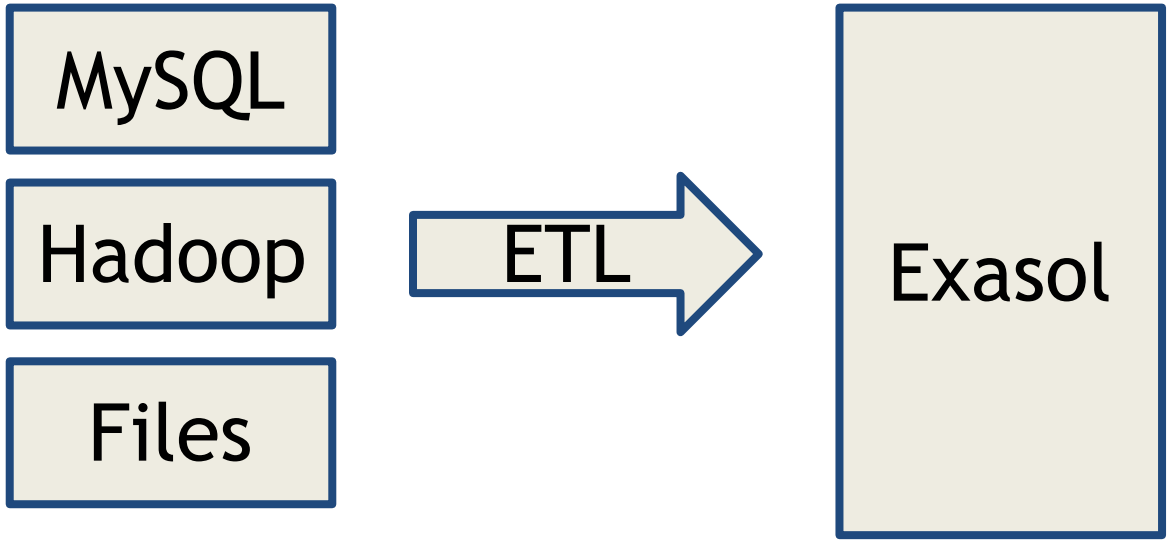


Непонятно!

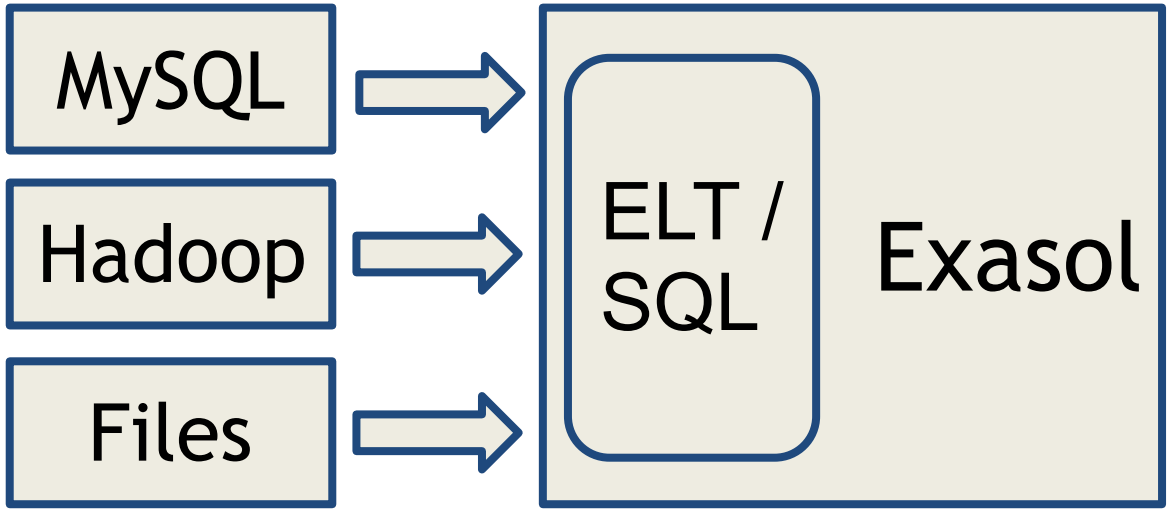


Что такое ETL?

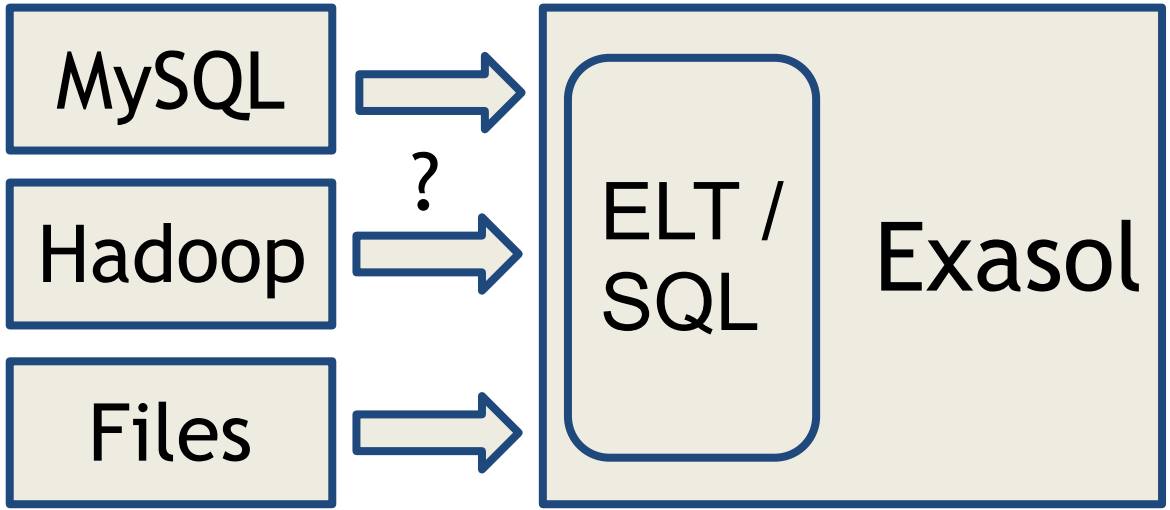
ETL большого проекта



ETL => ELT



Какой формат?



Формат TSV

- Почти как CSV (comma-separated), только tab-separated

Формат TSV

- Почти как CSV (comma-separated), только tab-separated
- Нет проблем с экранированием запятых и кавычек

Формат TSV

- Почти как CSV (comma-separated), только tab-separated
- Нет проблем с экранированием запятых и кавычек
- В BI табы и переводы строк встречаются редко

Формат TSV

- Почти как CSV (comma-separated), только tab-separated
- Нет проблем с экранированием запятых и кавычек
- В BI табы и переводы строк встречаются редко
- Удобно работать в командной строке (cut, awk, sed, sort,...)

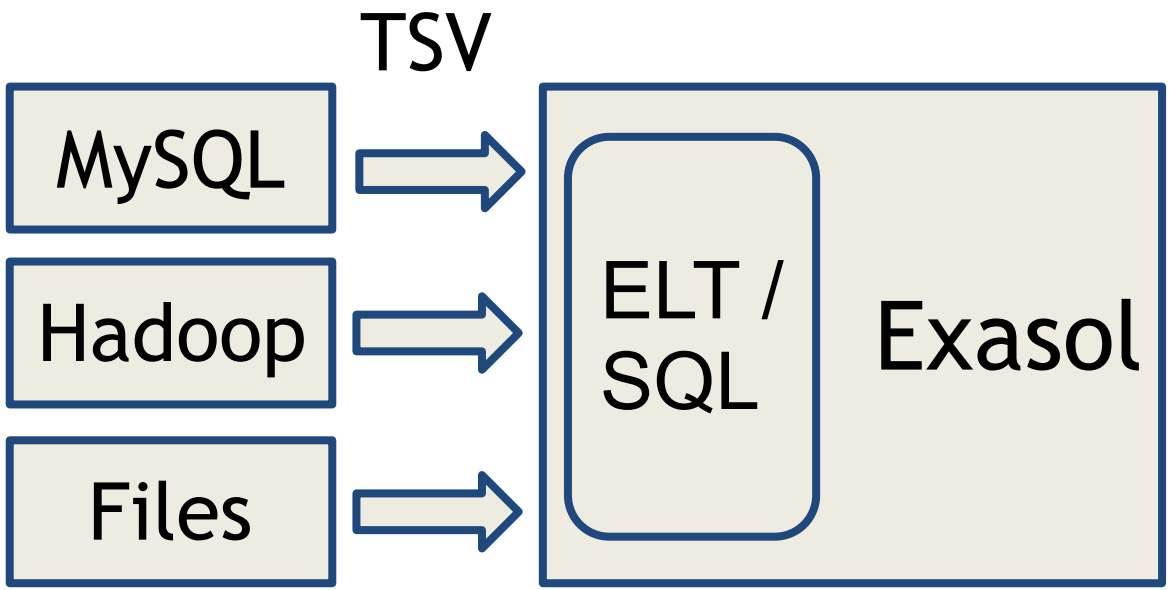
Формат TSV

- Почти как CSV (comma-separated), только tab-separated
- Нет проблем с экранированием запятых и кавычек
- В BI табы и переводы строк встречаются редко
- Удобно работать в командной строке (cut, awk, sed, sort,...)
- Поддерживается везде (CSV с набором опций)

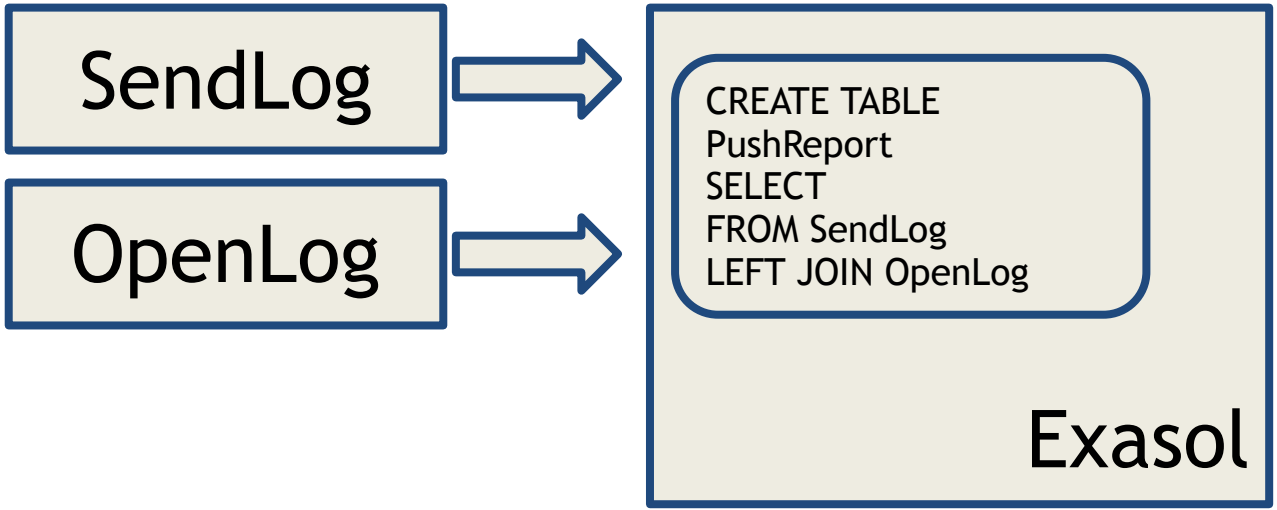
Формат TSV

- Почти как CSV (comma-separated), только tab-separated
- Нет проблем с экранированием запятых и кавычек
- В BI табы и переводы строк встречаются редко
- Удобно работать в командной строке (cut, awk, sed, sort,...)
- Поддерживается везде (CSV с набором опций)
- Удобен в реализации

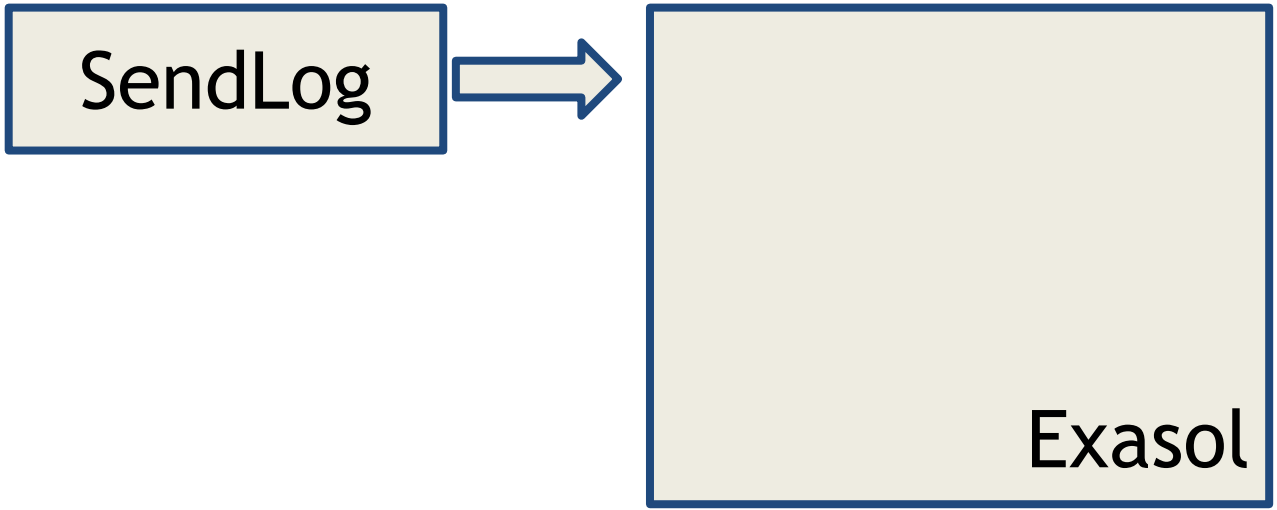
TSV



Пример: Push уведомления



Пример: Push уведомления



Передача MySQL -> Exasol

- Через временный файл

```
mysql -nsq -e
      'SELECT * FROM SendLog' > /tmp/
file
```

30 минут, 100Gb, 500M строк

```
mysql -nsq -e
      'SELECT * FROM SendLog' > /tmp/
file


exajload -sql
      "IMPORT INTO staging
        FROM LOCAL CSV FILE
       '/tmp/file'"
```

20 минут

50 минут

Передача MySQL -> Exasol

- Через временный файл
- Через pipe

```
mysql -nsq -e
      'SELECT * FROM SendLog'
```

```
mysql -nsq -e
      'SELECT * FROM SendLog' |
exajload -sql
      "IMPORT INTO staging
        FROM LOCAL CSV FILE
       '/dev/stdin'"
```

```
mysql -nsq -e
      'SELECT * FROM SendLog' |
exajload -sql
      "IMPORT INTO staging
        FROM LOCAL CSV FILE
       '/dev/stdin'"
```

```
set -o pipefail;
mysql -nsq -e
      'SELECT * FROM SendLog' |
exajload -sql
      "IMPORT INTO staging
        FROM LOCAL CSV FILE
       '/dev/stdin'"
```

Не забыть!!!

30 минут

Передача MySQL -> Exasol

- Данные с одной машины
- А что, если шардинг?

```
(
  mysql -nsq -h host1 'SELECT ....' &&
  mysql -nsq -h host2 'SELECT ....' &&
  mysql -nsq -h host3 'SELECT ....'
) | exajload -sql "IMPORT ... '/dev/
stdin"'
```

```
(
  mysql -nsq -h host1 'SELECT ....' &&
  mysql -nsq -h host2 'SELECT ....' &&
  mysql -nsq -h host3 'SELECT ....'
) | exajload -sql "IMPORT ... '/dev/
stdin"'
```

Последовательно 8(

MySQL => Exasol : multi pipe

fdlinecombine https://github.com/vi/fdlinecombine

Multiplex multiple input streams into stdout using \n as separators.

Does One Thing, But Does It Well.

```
fdlinecombine
    <(mysql -nsq -h host1 'SELECT ....')
    <(mysql -nsq -h host2 'SELECT ....')
    <(mysql -nsq -h host3 'SELECT ....')
    |exajload -sql "IMPORT ... '/dev/
stdin"'
```

```
fdlinecombine
    < (mysql -ns          t1 'SELECT ....')
    < (mysql -ns          t2 'SELECT ....')
    < (mysql -ns          t3 'SELECT ....')
    |exajload -sq          ORT ... '/dev/
stdin'''
```

Bash Pitfalls

MySQL => Exasol : multi pipe

Сделай сам!

PHP: proc_open()

Python: subprocess.Popen()

MySQL => Exasol : multi pipe

Сделай сам!
Универсальные альтернативы:
/dev/fd/%N
mkfifo - создать named pip
/dev/stdin, **/dev/stdout**

А почему бы не использовать GNU parallel?

“GNU parallel makes sure output from the commands is the **same output as you would get** had you **run the commands sequentially**. This makes it possible to use output from GNU parallel as input for other programs.”

GNU parallel

- Может мержить потоки (опция --line-buffer)

GNU parallel

- Может мержить потоки (опция --line-buffer)
- При этом жрёт много ресурсов

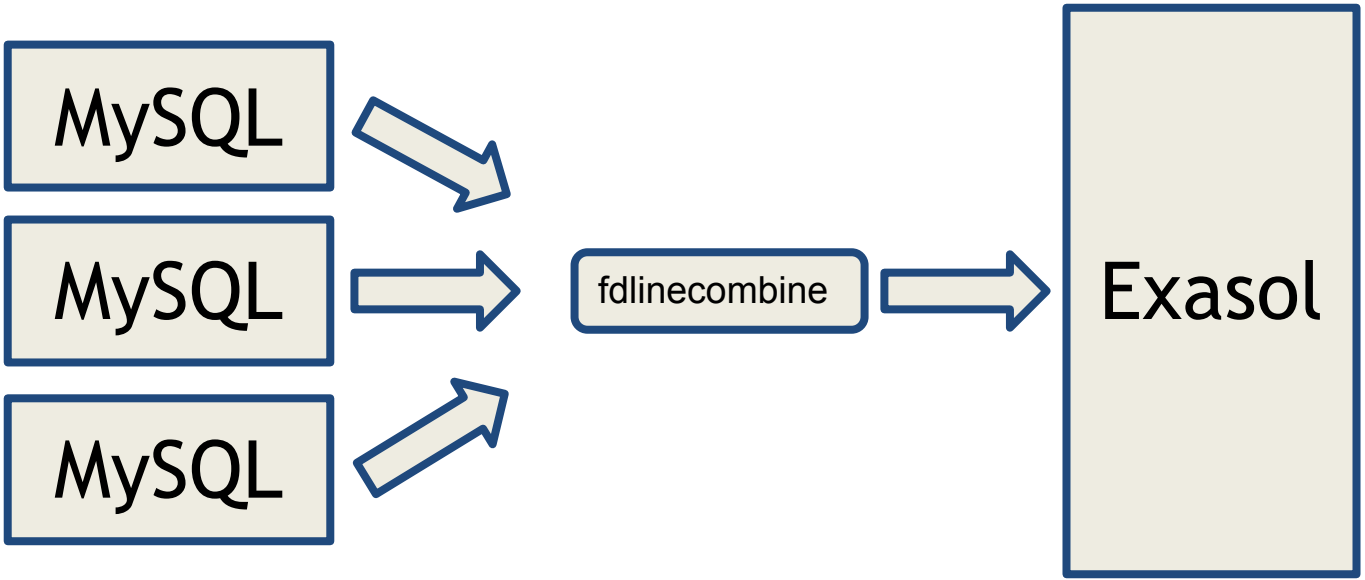
GNU parallel

- Может мержить потоки (опция --line-buffer)
- При этом жрёт много ресурсов
- Сложен в отладке

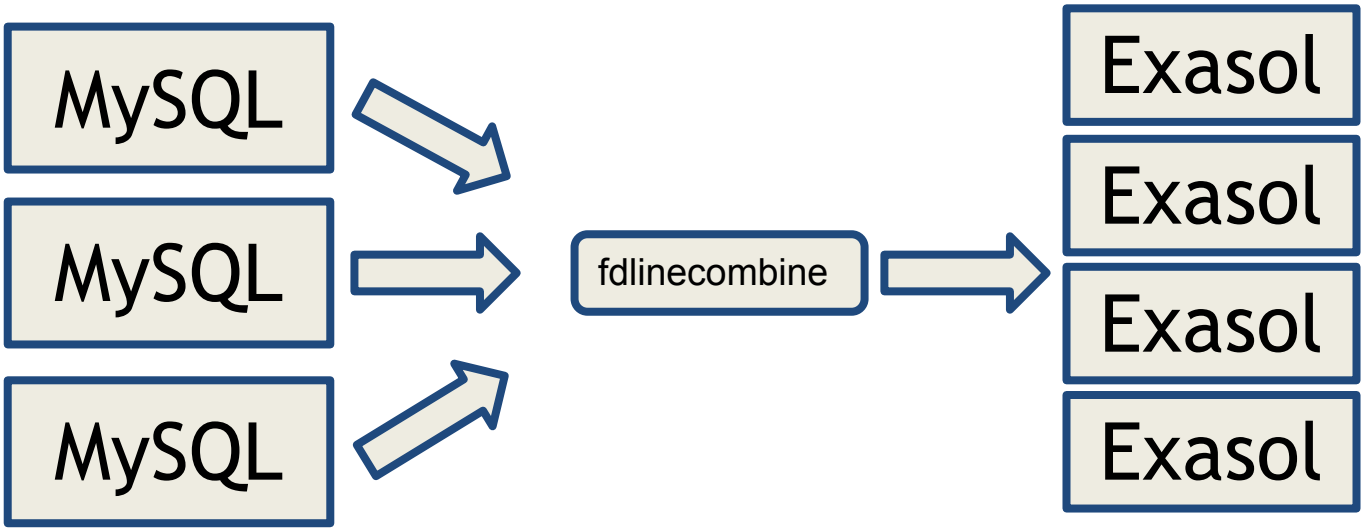
GNU parallel

- Может мержить потоки (опция --line-buffer)
- При этом жрёт много ресурсов
- Сложен в отладке
- Мы находили баги (теряли данные)

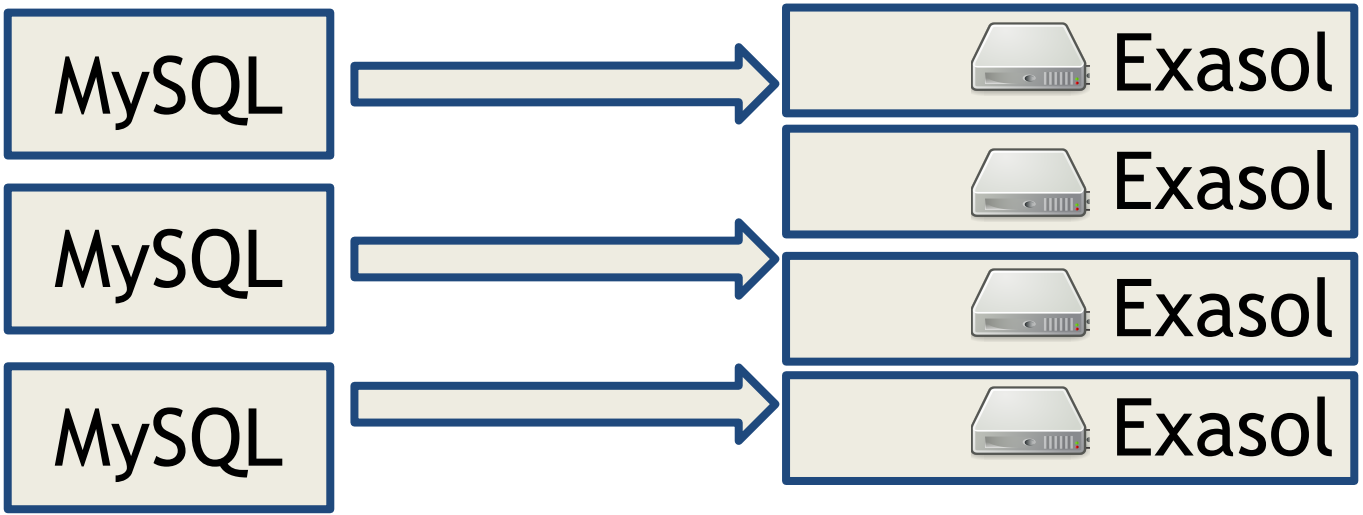
MySQL => Exasol



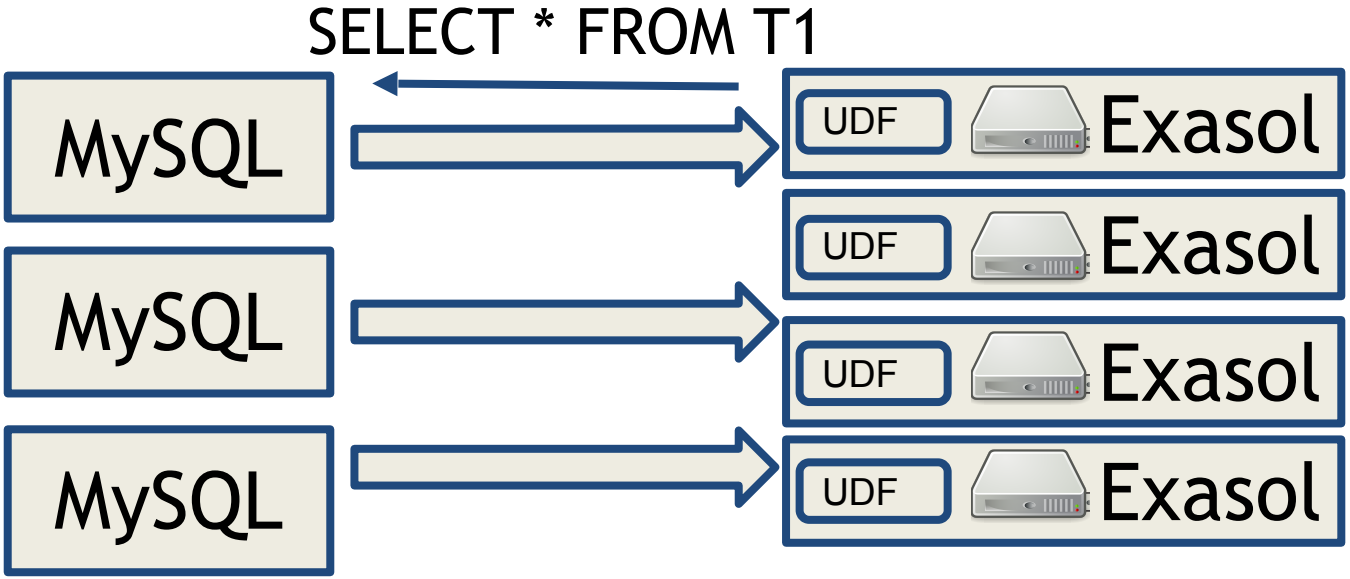
MySQL => Exasol



MySQL => Exasol



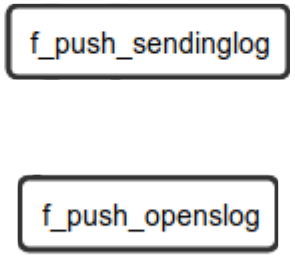
MySQL => Exasol



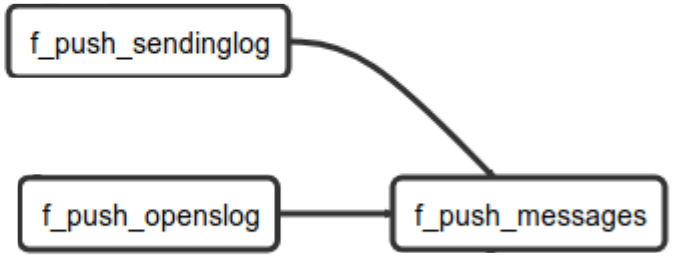
В 5 раз быстрее

ETL граф

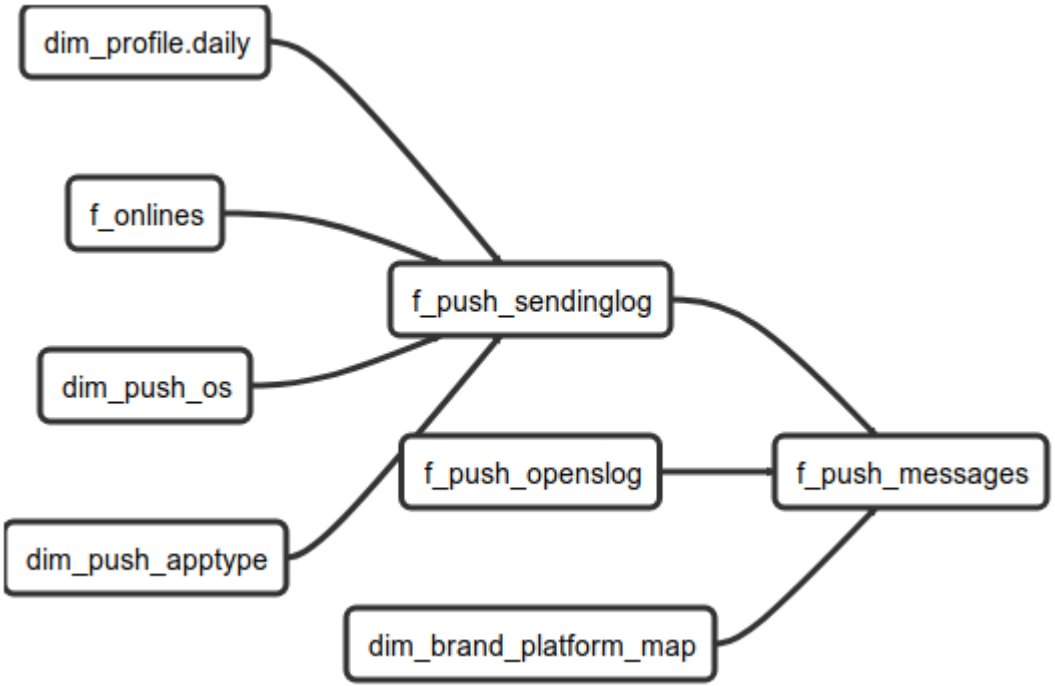
ETL граф



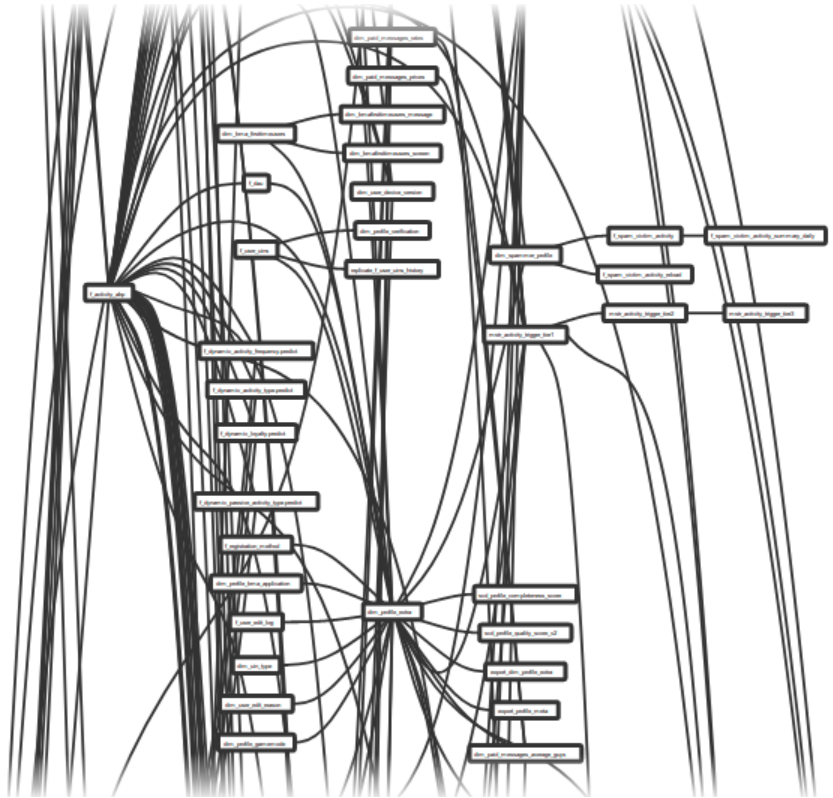
ETL граф



ETL граф


dim_profile.daily
f_onlines
f_push_sendinglog
dim_push_os
f_push_openslog
f_push_messages
dim_push_apptype
dim_brand_platform_map

ETL большого проекта



Обработка потока событий

Продуктовая разработка

- Вопросы про данные

Продуктовая разработка

- Вопросы про данные
  - Собираем больше информации

Продуктовая разработка

- Вопросы про данные
  - Собираем больше информации
- Эксперименты и новые метрики

Продуктовая разработка

- Вопросы про данные
  - Собираем больше информации
- Эксперименты и новые метрики
  - Настраиваем ETL
  - ALTER TABLE ….
  - Генерируем отчёты

Доколе!?

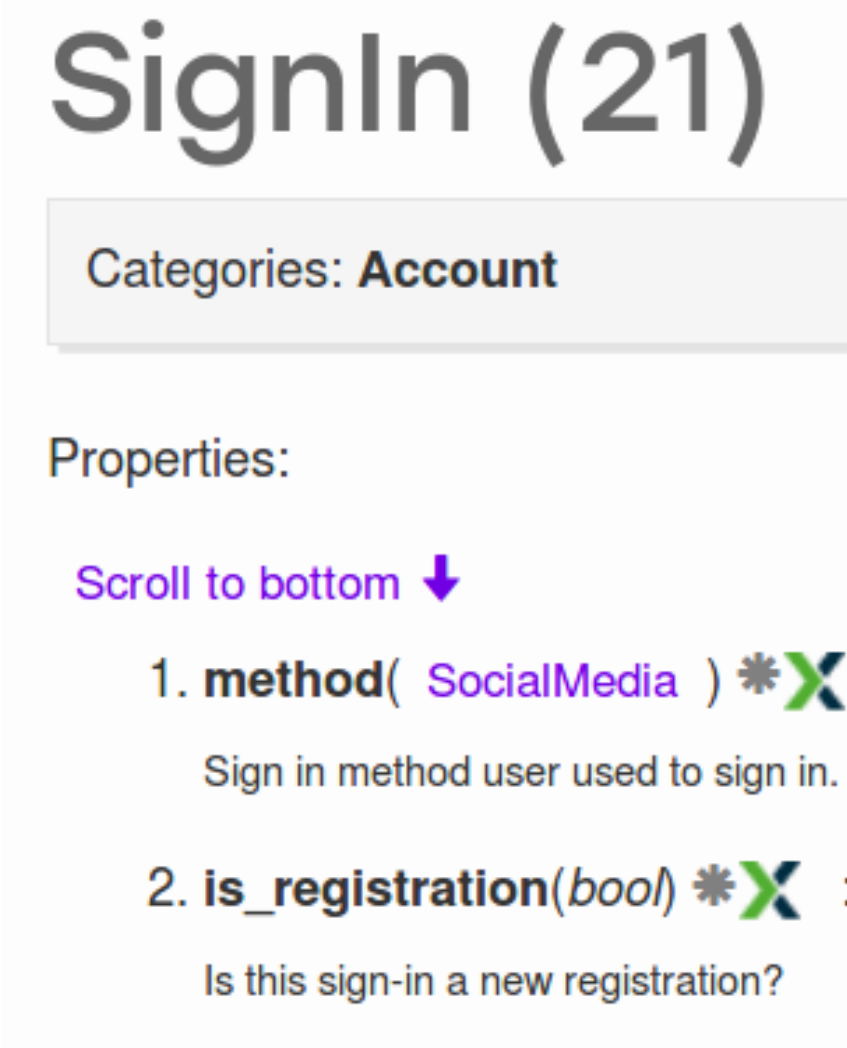
Автоматизируй это!

- Формальное, machine-readable описание данных
- Автоматизация доставки в хранилища
- Автоматизация репортинга

Формальное описание событий

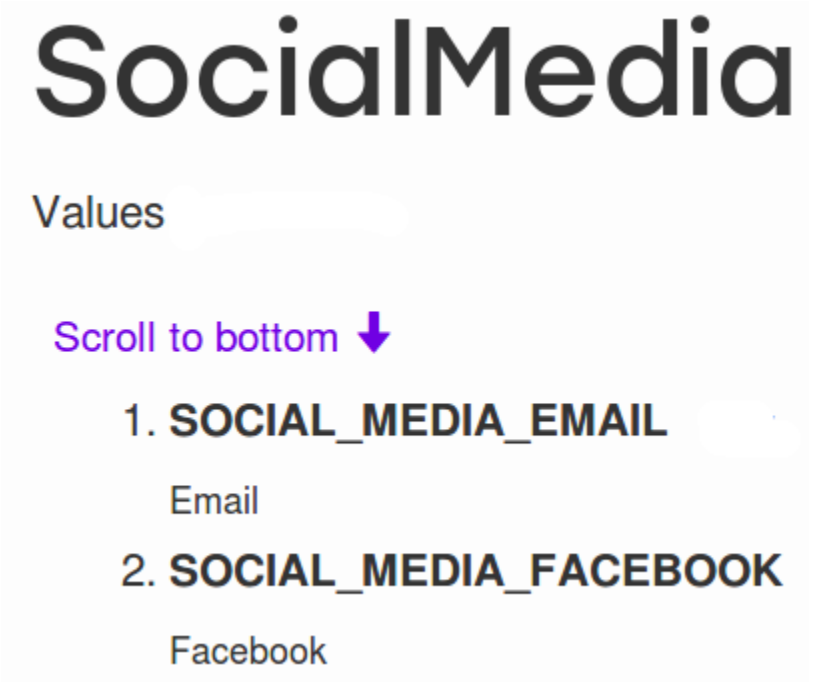
- Единый формат для всех платформ
- Строгая типизация атрибутов
- Понятно человеку и машине

Событие «Авторизация»



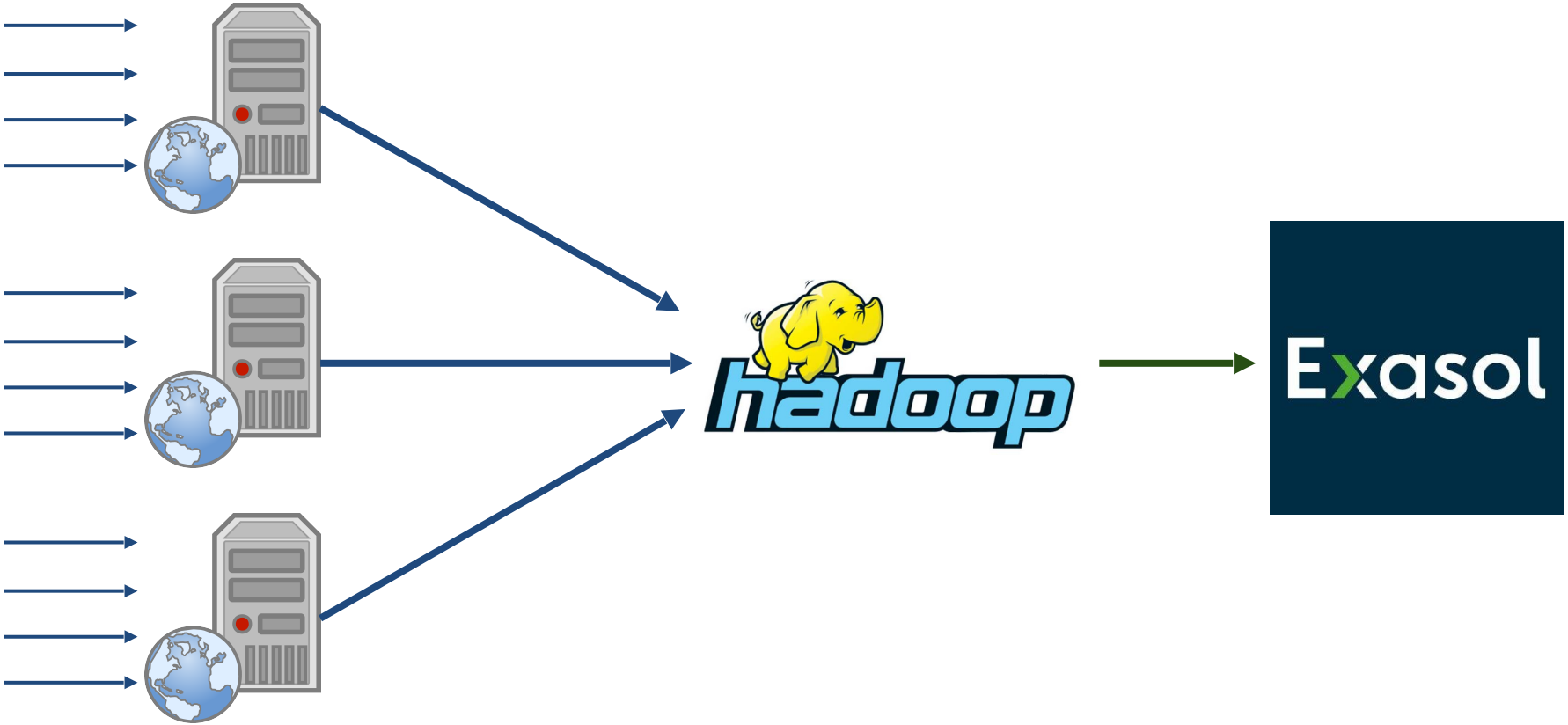
Событие «Авторизация»

ENUM «Способы авторизации»



Доставка в хранилища

- Сгенерировать события на backend
- Преобразовать
- Загрузить в хранилище


hadoop
Exasol

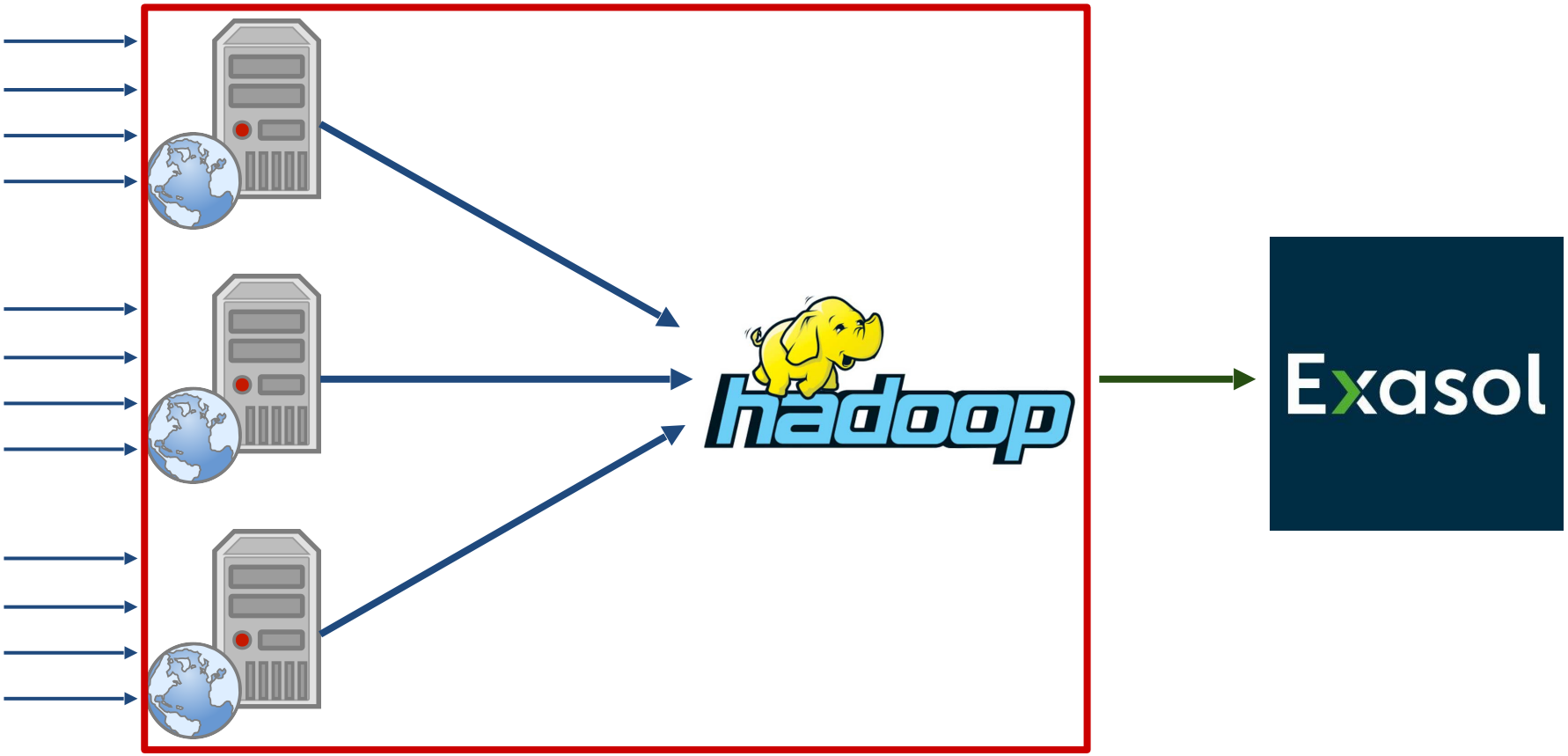

hadoop
Exasol

Схема в Hadoop (Hive)

- Единая таблица для хранения всех событий
- Хранение атрибутов в map<string, string>

Схема в Hadoop (Hive)

- Единая таблица для хранения всех событий
- Хранение атрибутов в map<string, string>
- Модный формат хранения ORC

Схема в Hive: преимущества

- Поддержка любых атрибутов событий
- Top-level reporting по всему объему данных
- Оперативное обновление

Схема в Hive: недостатки

- Большое количество партиций
- Overhead на хранение атрибутов в строках

Доставка в Hive

- Накопить данные в HDFS
- Сделать временную таблицу (JSON)
- INSERT OVERWRITE TABLE table_name SELECT …. FROM table_temporary

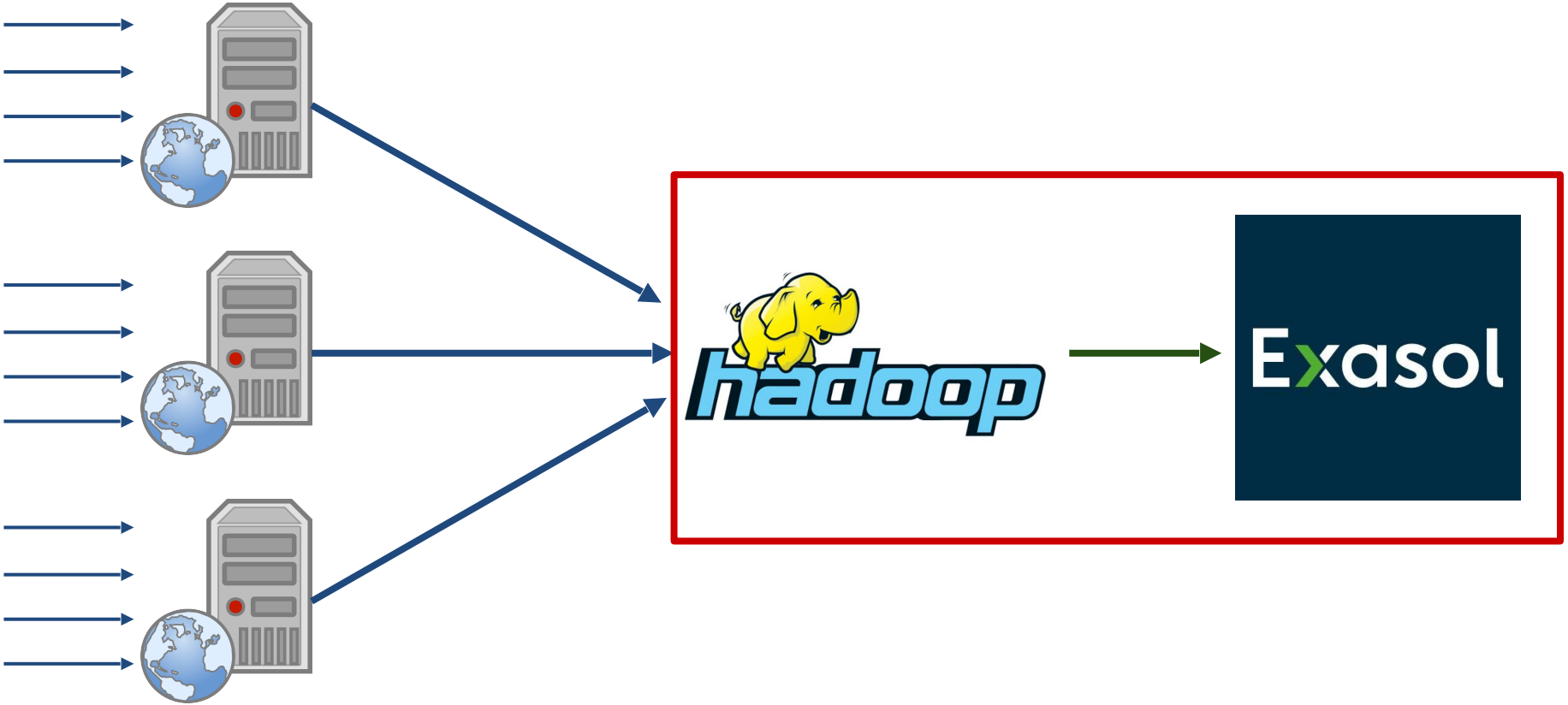

hadoop
Exasol

Схема хранения в Exasol

- Таблица на каждый тип события
- Каждому атрибуту – отдельная колонка
- Общая информация

  о событиях в отдельной таблице

Header

head_id	12345678901234
user_id	100500
gender	Male
country	Russia
platform	Android

Событие SignIn

head_id	method	job_id
12345678901234	Email	job_one
98765432109876	Facebook	job_two

Событие ViewProfile

head_id	profile_id	job_id
12345678901234	1567854	job_one
98765432109876	4589783	job_one

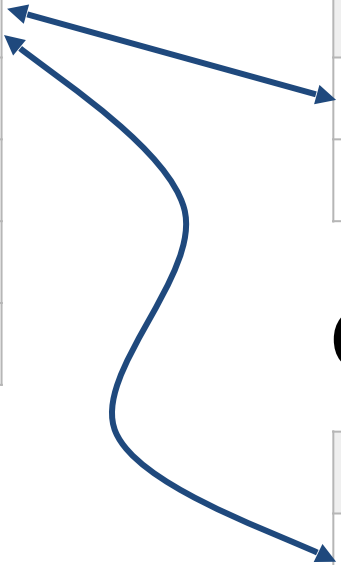


Схема в Exasol: преимущества

- Оптимизация хранилища под типы данных
- Быстрый репортинг по отдельным событиям

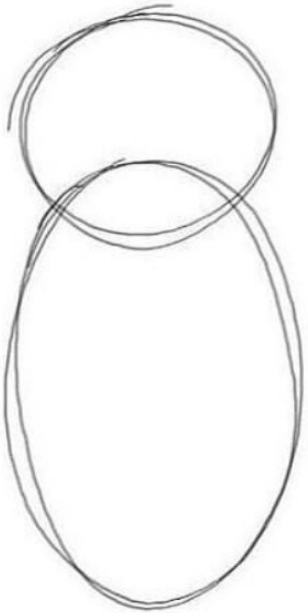
Схема в Exasol: недостатки

- Надо поддерживать согласованность конфига событий и схемы таблиц
- Нет сквозного репортинга по всем данным

Доставка в Exasol

- Взять данные из Hadoop
- Сформировать открыть транзакцию в Exasol
- Вставить данные
- COMMIT в Exasol

Baykuş nasıl çizilir



1. ADIM

ÖNCE IKI DAIRE CIZIYORUZ...

2. ADIM

SONRA KALAN DETAYLARI ÇIZIYORUZ.

Hadoop -> Exasol

- Объём исходных данных – до 150 Gb
- У каждого события – своя схема

Схема pipeline

- Выбрать данные из Hive в Spark
- Для каждого события получаем колонки из Exasol
- Формируем TSV-файлы в HDFS
- Импортируем в Exasol

```
IMPORT INTO data.click_event
FROM CSV AT
‘http://hadoop:50070/etl_xxxx/signin/'
FILE '1.csv.gz',
...
FILE 'XXX.csv.gz';
```

Hadoop WebHDFS

```
IMPORT INTO data.click_event
FROM CSV AT
‘http://hadoop:50070/etl_xxxx/signin/'
FILE '1.csv.gz',
...
FILE 'XXX.csv.gz';
```

Параллельная обработка событий одного типа

Преимущества

- Распределение дисковой нагрузки
  - HDFS
  - Spark Workers
- Node-to-node взаимодействие при IMPORT
- Идемпотентность - перезаливка части событий из job_id

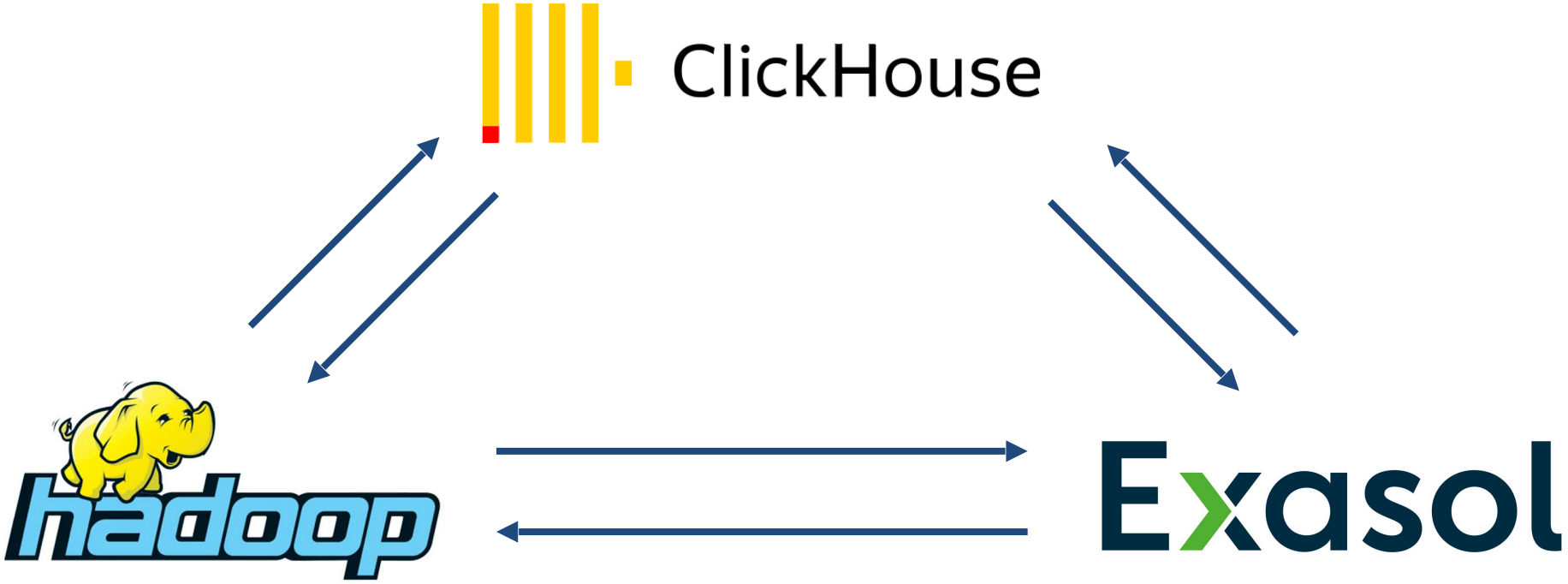
Cross-database bridges

Зачем строить мосты?

- Каждая база хороша по-своему
- Не все задачи можно выполнить внутри одного хранилища

Зачем строить мосты?

- Каждая база хороша по-своему
- Не все задачи можно выполнить внутри одного хранилища
- Требуется обмен данными между большими системами

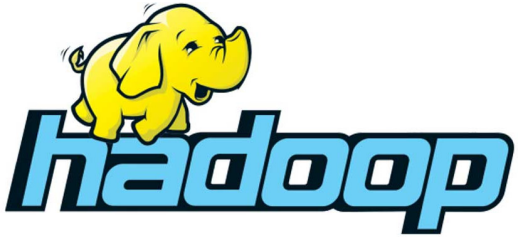

ClickHouse
hadoop
Exasol

ClickHouse

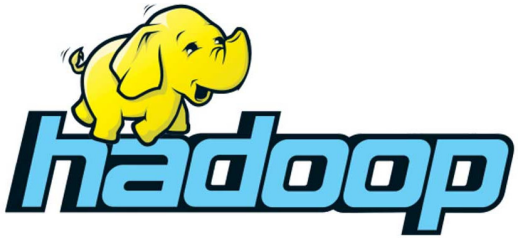
- Сверхбыстрая выборка в пределах таблицы
- Real-time вставка данных
- Можно делать мелкие вставки

ClickHouse

- Слабый JOIN
  - не все аналитические задачи решаются
- Важно думать о data locality
  - иначе не сделать JOIN больших таблиц



- Принимает данные в любых форматах
- Лёгкое масштабирование
- Удобное управление репликацией и retention
- Можно делать любые JOIN
  - хоть и не всегда быстро



- Надо думать о партицировании данных
- Плохо работает с маленькими вставками
- Высокое значение latency

Exasol

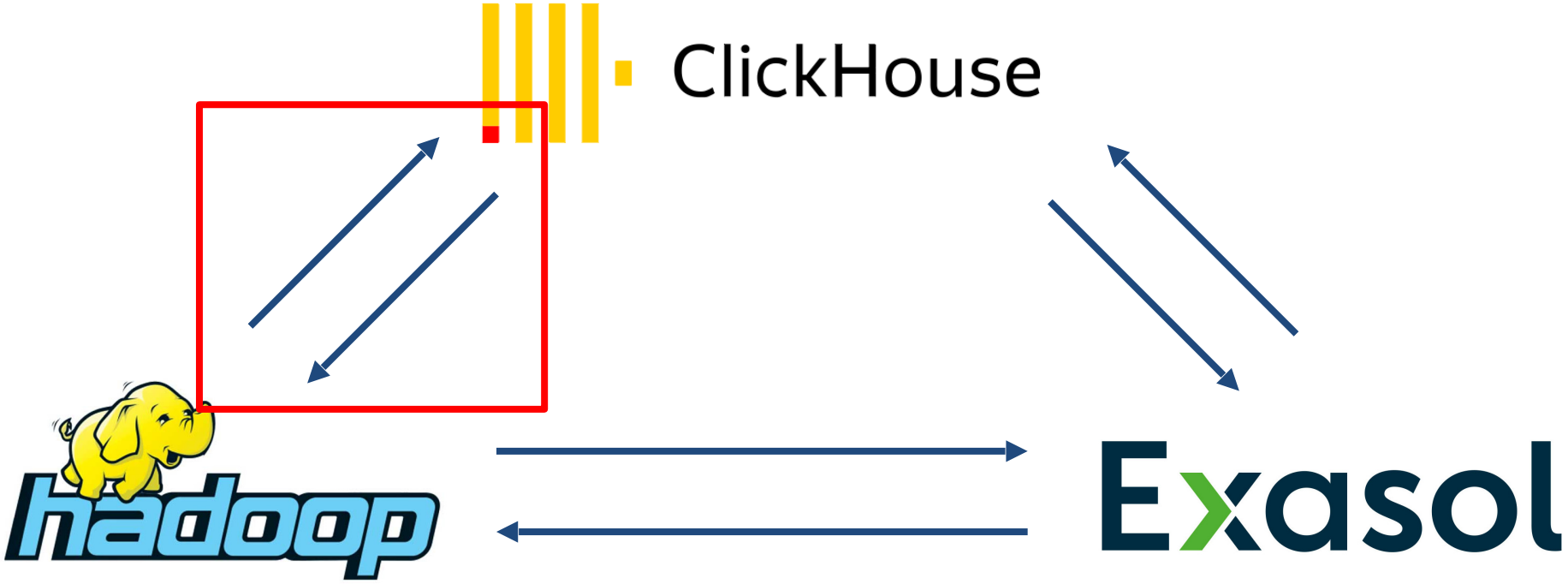
- Высокая скорость сканирования
- Любые аналитические операции
- Всё быстро

Exasol

- Дорого хранить исторические данные
- Вставка большими порциями и редко

Всё круто, и всё же – мосты?

- Ни одно хранилище не обладает 100% информации
- Будем брать лучшее, и доставлять недостающее


ClickHouse
hadoop
Exasol

Hadoop to ClickHouse

- Запрос в ClickHouse на импорт данных
- Используя Spark

```
SELECT * FROM
hdfs(
    'hdfs://hdfs1:9000/file',
    TSV,
    'id UInt64, number Float64'
)
```

```
SELECT * FROM
hdfs(
    'hdfs://hdfs1:9000/file',     --- URL файла
    TSV,  --- формат данных
    'id UInt64, number Float64'--- схема данных
)
```

Hadoop to ClickHouse via SQL

- Нативно, внутри ClickHouse
- Поддержка лишь одного файла за раз
  - в разработке поддержка globbing
- Требуется согласованность форматов данных
- Всё прокачивается через
  1 машину ClickHouse

Hadoop to ClickHouse via Spark

- Читаем данные в Spark
- Преобразуем
- Каждую партицию данных пишем через JDBC-коннектор + нестандартный метод sendRowBinaryStream

Hadoop to ClickHouse via Spark

- Не только SQL для преобразований
- Параллелизм обработки
- Пишем в любые ноды ClickHouse

Hadoop to ClickHouse via Spark

- Не только SQL для преобразований
- Параллелизм обработки
- Пишем в любые ноды ClickHouse
- Надо программировать
- Ограничено Spark + Java/Scala
- Накладные расходы от Spark

ClickHouse to Hadoop

- Запрос в ClickHouse на экспорт данных
- Через Spark

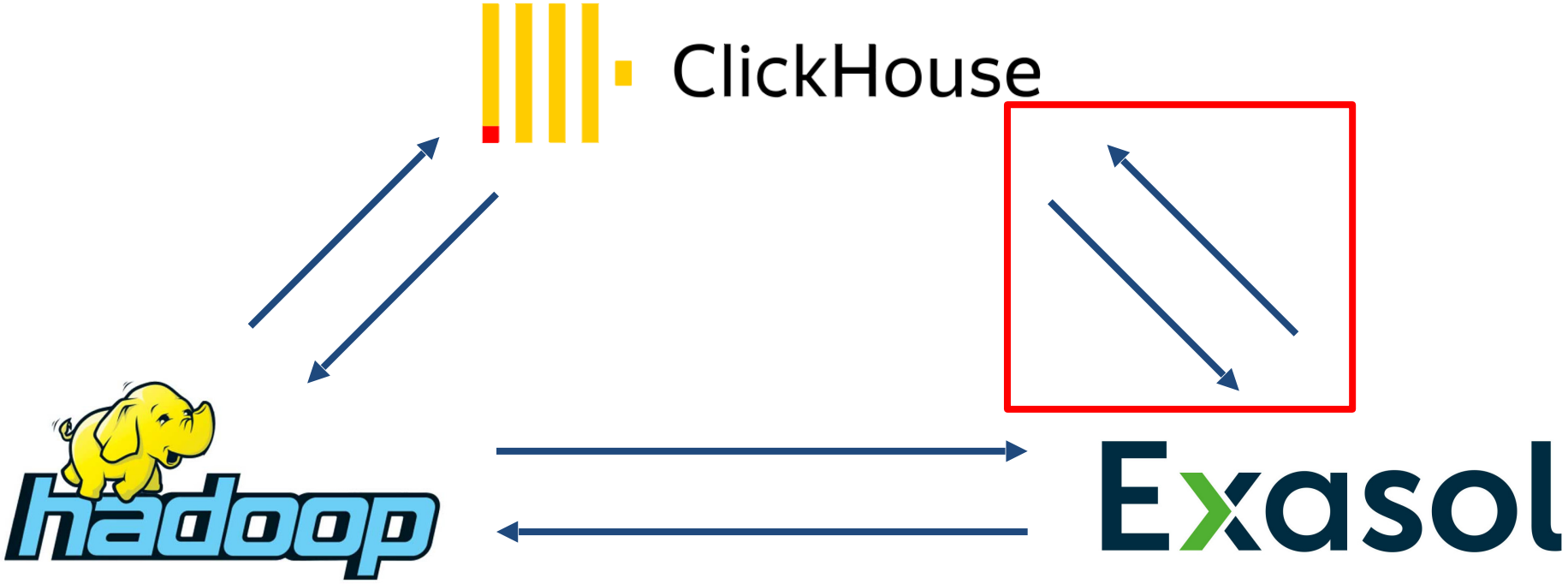
```
CREATE TABLE hdfs_table
(id Int32)
ENGINE = HDFS('hdfs://hdfs1:9000/file', 'TSV');

INSERT INTO hdfs_table
SELECT * FROM another_table;
```

ClickHouse to Hadoop via SQL

- Запись только в один файл
- Обрабатывается через 1 машину ClickHouse

ClickHouse to Hadoop via Spark

- JDBC + нестандартный метод executeQueryClickHouseRowBinaryStream
- Программное управление топологией данных


ClickHouse
hadoop
Exasol

ClickHouse to Exasol и обратно

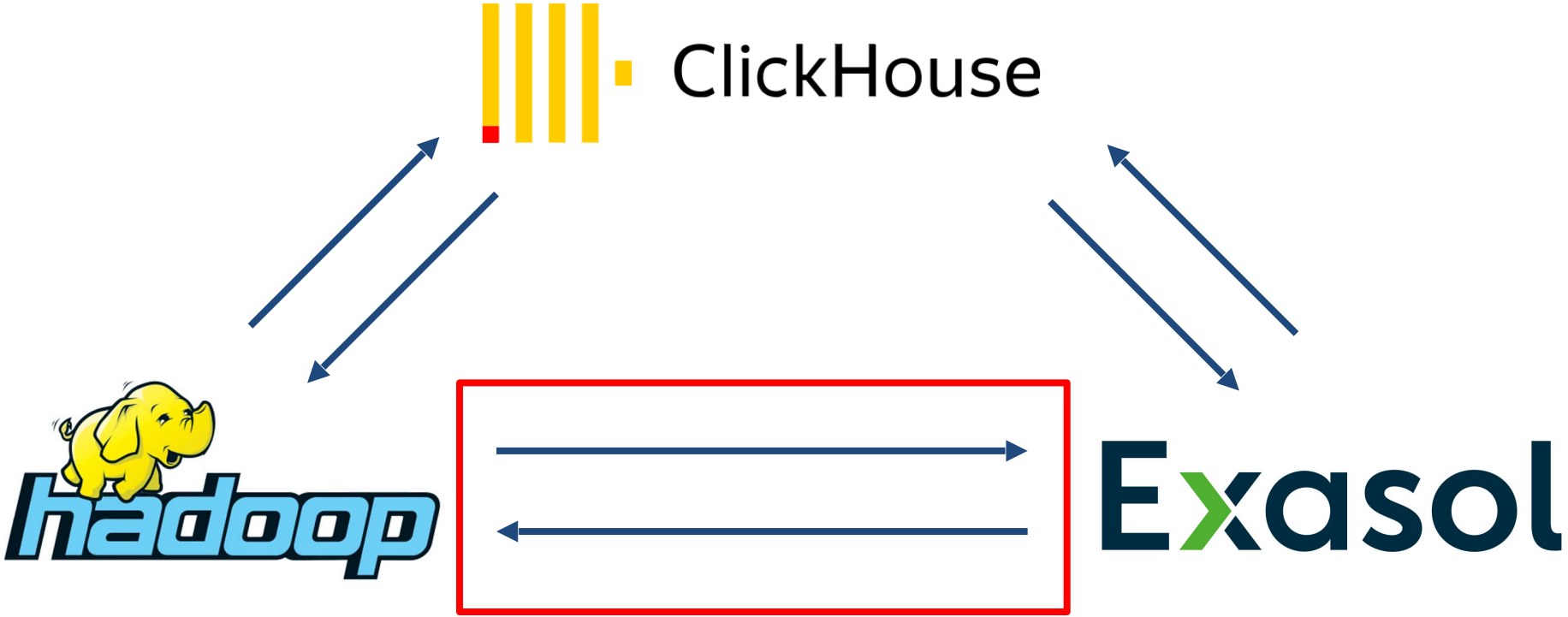
- Через IMPORT/EXPORT команды Exasol

```
IMPORT INTO exasol_table
FROM CSV AT
'http://clickhouse1:8123/'
FILE '?query=SELECT * FROM clickhouse_table'
```

```
EXPORT exasol_table
INTO CSV AT
'http://clickhouse1:8123/'
FILE
'?query=INSERT INTO clickhouse_table FORMAT CSV'
```

ClickHouse to Exasol и обратно

- Дёшево и просто
- Ограничено взаимодействием двух машин
  - исправить: через Spark
  - в связке с Exasol sub-connections


ClickHouse
hadoop
Exasol

Hadoop to Exasol

- WebHDFS + TSV + Exasol IMPORT query
- Через Spark + Exasol sub-connections

Exasol to Hadoop

- Exasol EXPORT query
  - более одного файла
  - только CSV
- Exasol UDF
  - высокая производительность
  - любые форматы
  - требует программирования

Выводы

- BI – это круто :)
- Подбираем хранилище данных под задачу
- Берём лучшее от разных систем
- В экспериментах рождается прогресс

СПАСИБО!

krash@corp.badoo.com
a.eremihin@corp.badoo.com

badoo_tech