Загрузка больших объемов данных для бизнес-аналитики

Старынин Валерий, BI-разработчик в Badoo

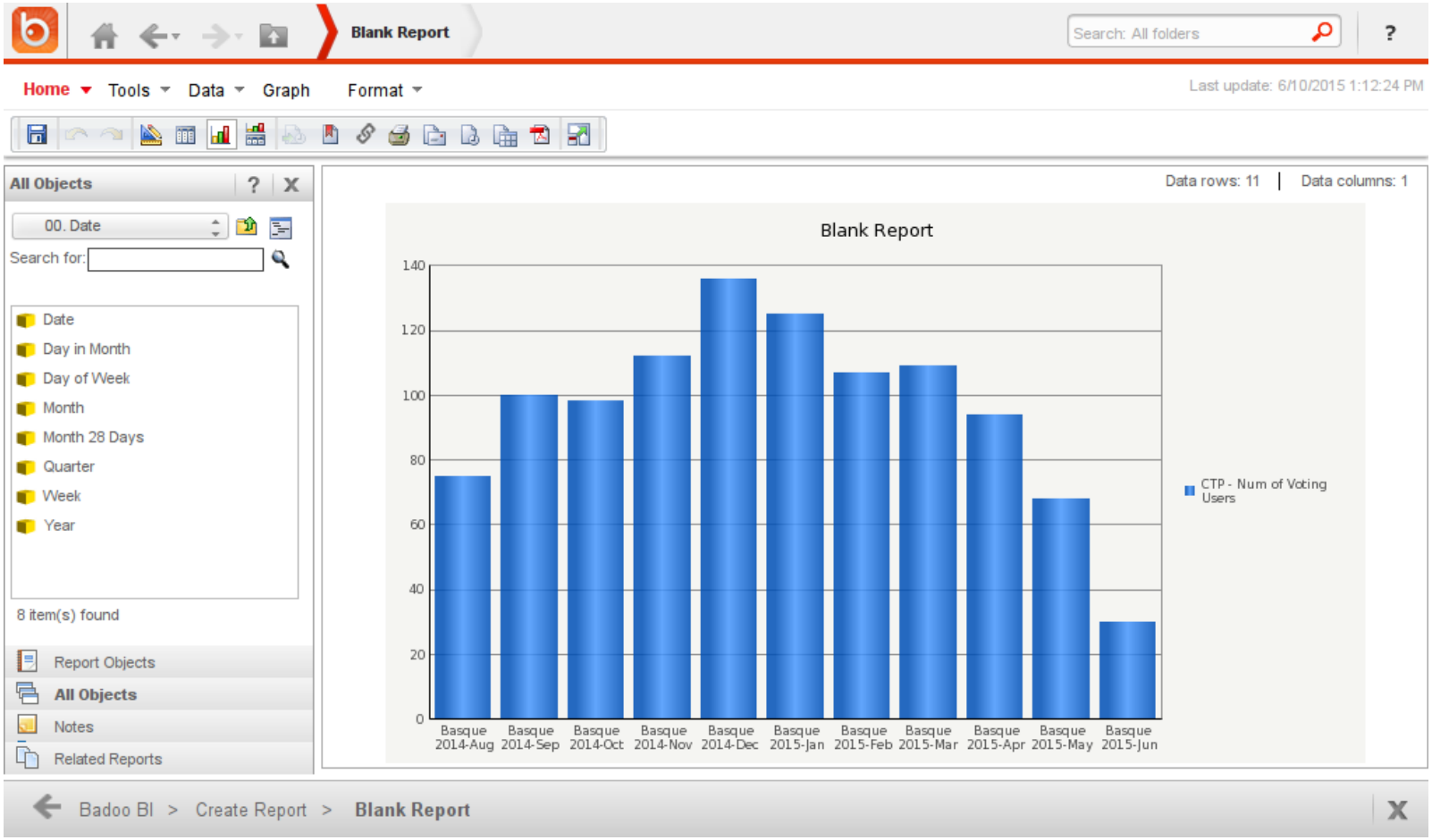
В докладе будет рассказано:

- Что делает BI в Badoo
- Какие задачи по сбору данных у нас есть
- Про наш инструмент загрузки данных: ETLMaster
- Tips & Tricks: как ускорить загрузку данных

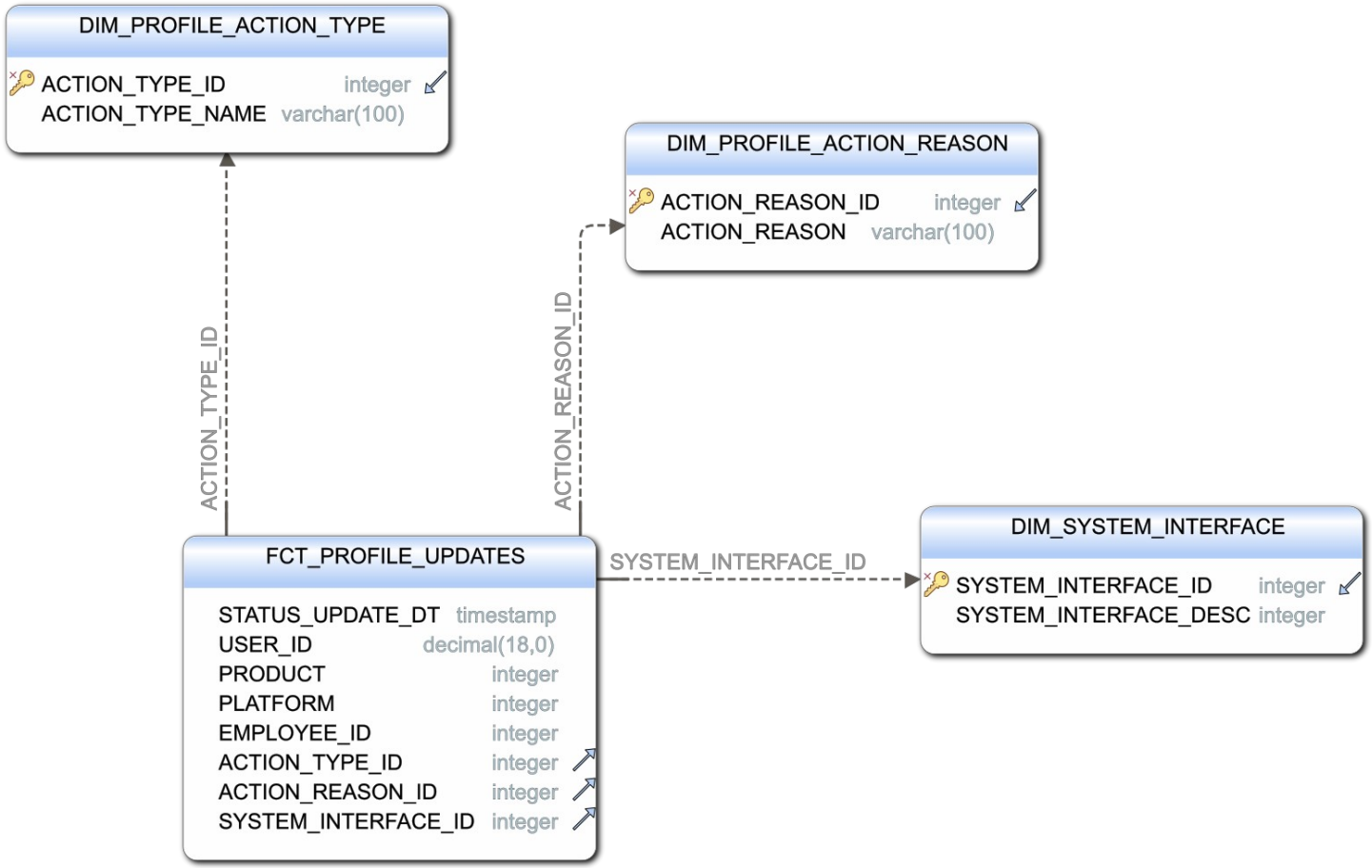
Badoo это

- Социальная сеть для поиска новых друзей
- 250 млн. пользователей
- Работаем во всех странах мира
- Мобильные приложения под Android, iOS, Windows, BlackBerry. А так же Wap и HTML5 версии
- 2,5 датацентра: в Европе, Америке и Азии
- Более 3 000 серверов

Что мы делаем в BI?

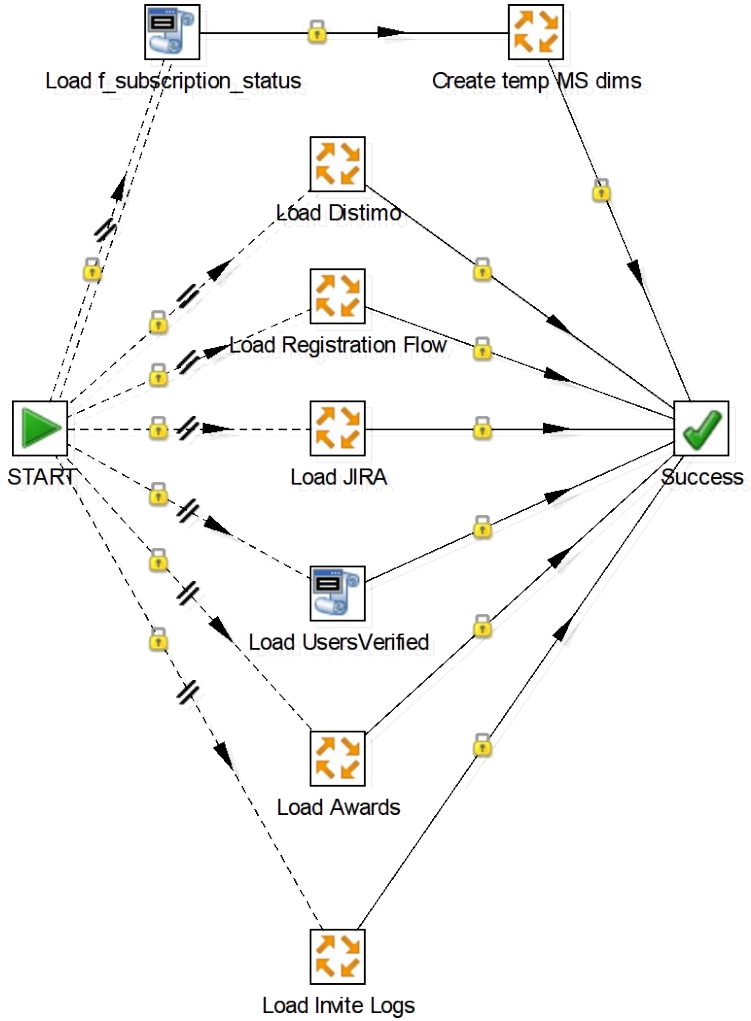


Аналитическая БД



Как мы загружали данные

- 1 запуск в день, загрузка до начала рабочего дня
- Куча скриптов
- Жесткий порядок выполнения
- XML



Требования к системе

- Зависимости а не последовательность действий
- Текстовый конфиг (хранить в git'е)
- Замена cron
- Загрузка данных из STDIN
- Поддержка используемых БД
- Повтор выполнения при сбое

Дополнительные требования к системе

- Простой перезапуск заданий и перезагрузка данных
- Параметризация SQL
- Мониторинг ресурсов
- Интерфейс для просмотра статуса и управления запуском
- Изоляция dev-окружений

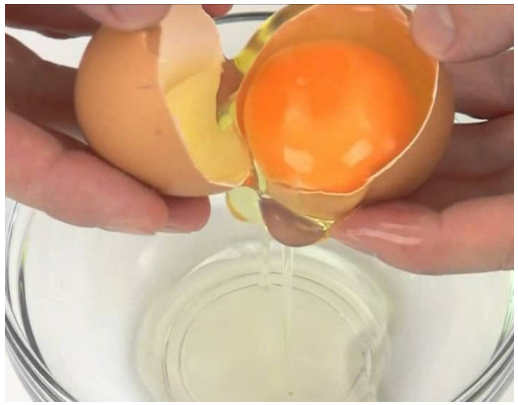
Альтернативы

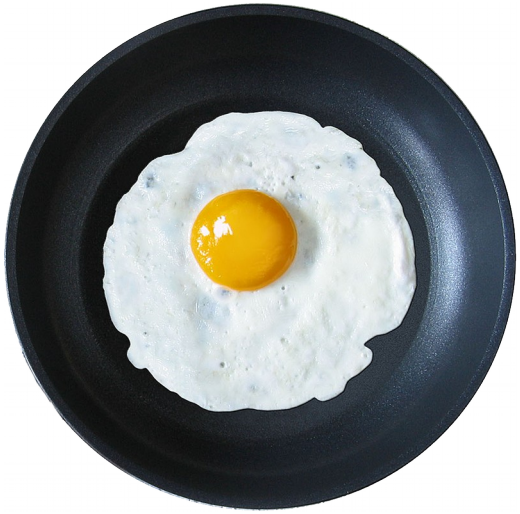
Не удовлетворяют нашим требованиям!

- Плохая поддержка БД
- Нет текстовых конфигов
- Нет запуска по зависимостям

Наше решение!

ETLMaster





Extract Transform Load

Интерфейс

NODES / LOAD / CPU	8 / 12.2 / 81.6%
RAM	186036.8 MiB
HDD READ / HDD WRITE	0 MiB / 1.9 MiB
NET	179.9 MiB
OCCUPIED SPACE	71.2%

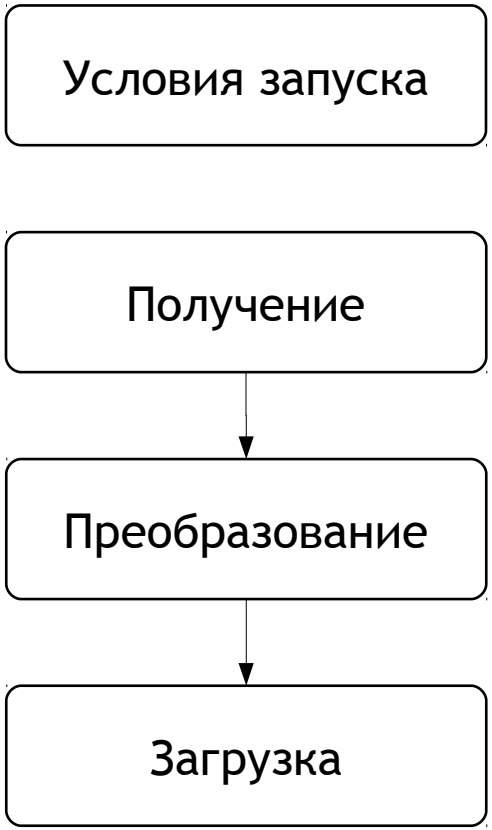
Server time	2015-06-18 13:08:05
ETLMaster status	**Running (pid: 12841)**
Pending tasks in queue	4
Completed tasks in 24h	802
Warnings and errors in 24h	5 WRN / 40 ERR

Queue | Config | Task Log | Errors | Signal | Master Log | Delay reasons | Reloads

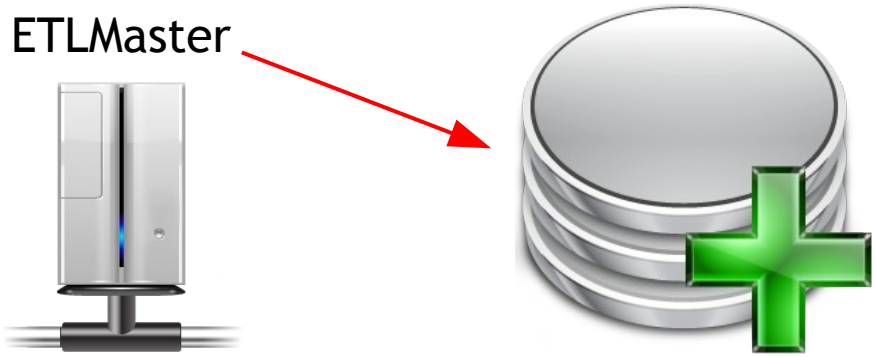
ID	Task name	Class	Status	Attempt	Created TS
			All		
209324	dim_profile.daily:ToProd		Completed	1	2015-06-18 02:24:41
209321	dim_profile.daily:KeepExisting		Completed	1	2015-06-18 02:19:40
209320	dim_profile.daily:Deduplicate		Completed	1	2015-06-18 02:18:21
209319	dim_profile.daily:AddDummies		Completed	1	2015-06-18 02:18:19
209058	dim_profile.daily:Load		Completed	1	2015-06-18 00:45:00
207382	dim_profile.daily:ToProd		Completed	1	2015-06-17 02:34:41
207372	dim_profile.daily:KeepExisting		Completed	1	2015-06-17 02:27:48
207369	dim_profile.daily:Deduplicate		Completed	1	2015-06-17 02:26:25

Пример задания

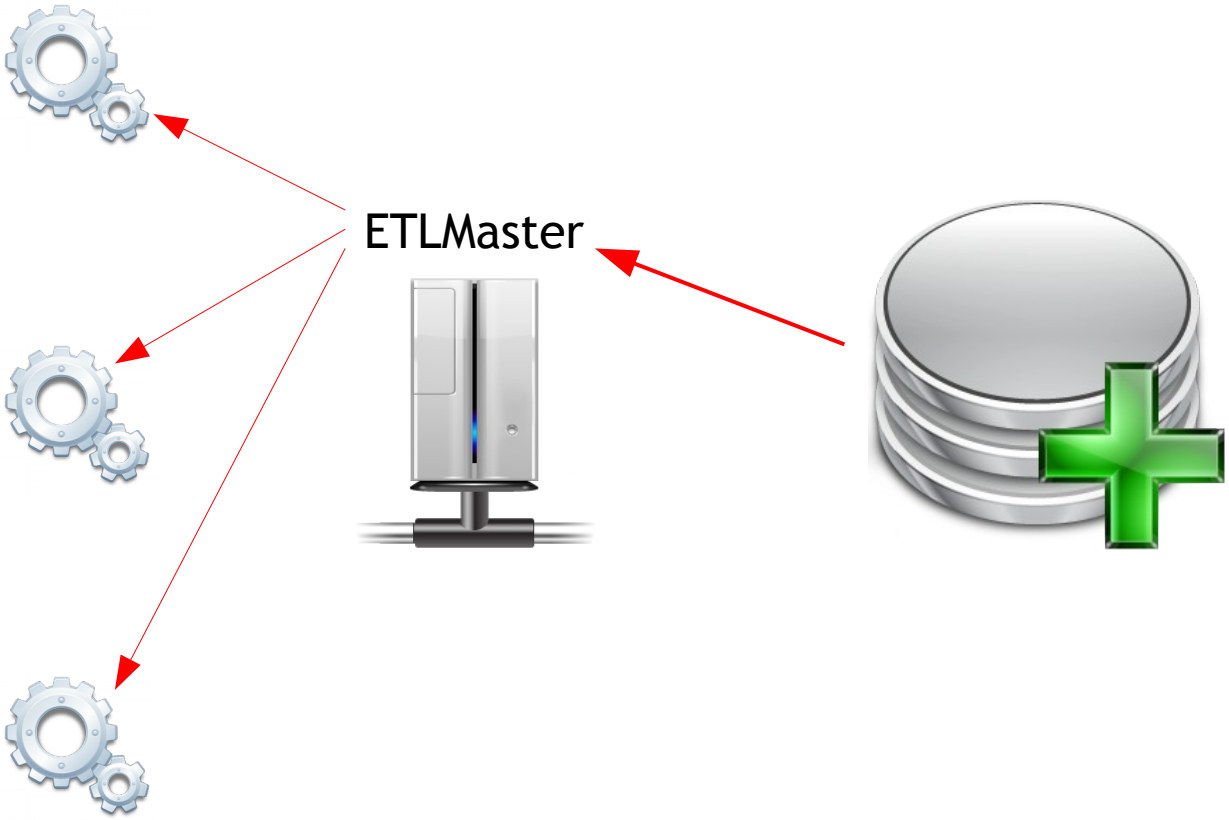
```
"autorun": {
  "timeshift": "00:45",
  "conflict": ["dim_profile.hourly"]
},
"tasks": [
  {
    "class": "LoadFromPhpDumperTask",
    "params": {
      "table_name": "staging_dim_profile",
      "dumper_name": "dumpDimProfileCli.php",
      "threads": "400"
    }
  },
  {
    "class":"SQLScriptTask",
    "params":{
      "file":"dim_profile_dedup.sql"
    }
  },
  {
    "class": "Stage2ProdAppendTask",
    "params": {
      "table_name": "dim_profile",
      "primary_key":"profile_user_id"
    }
  }
]
```



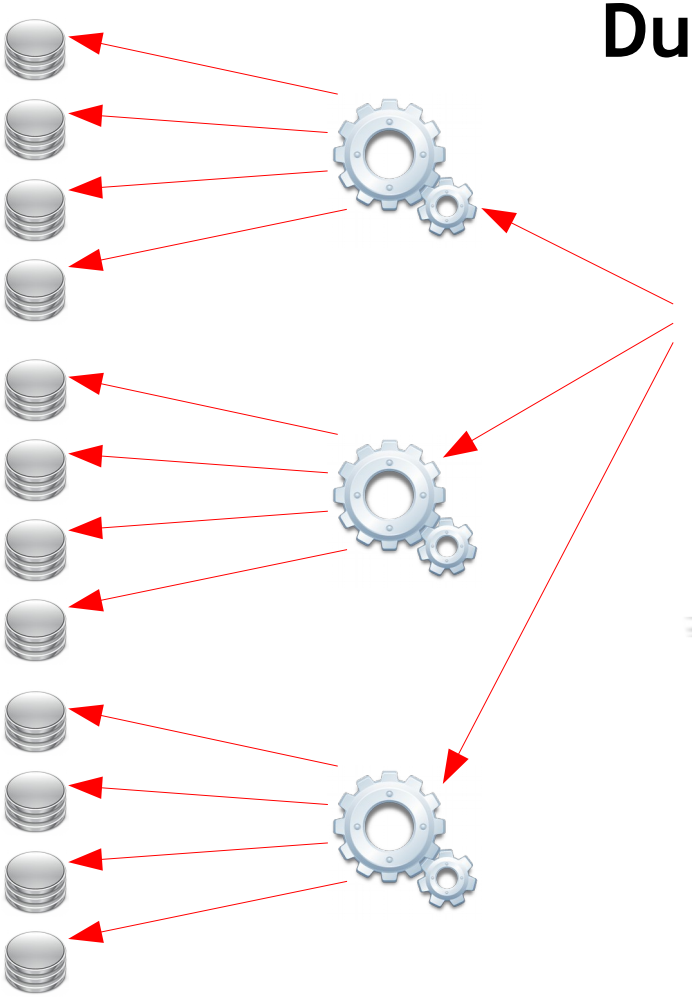
Dumper'ы



Dumper'ы

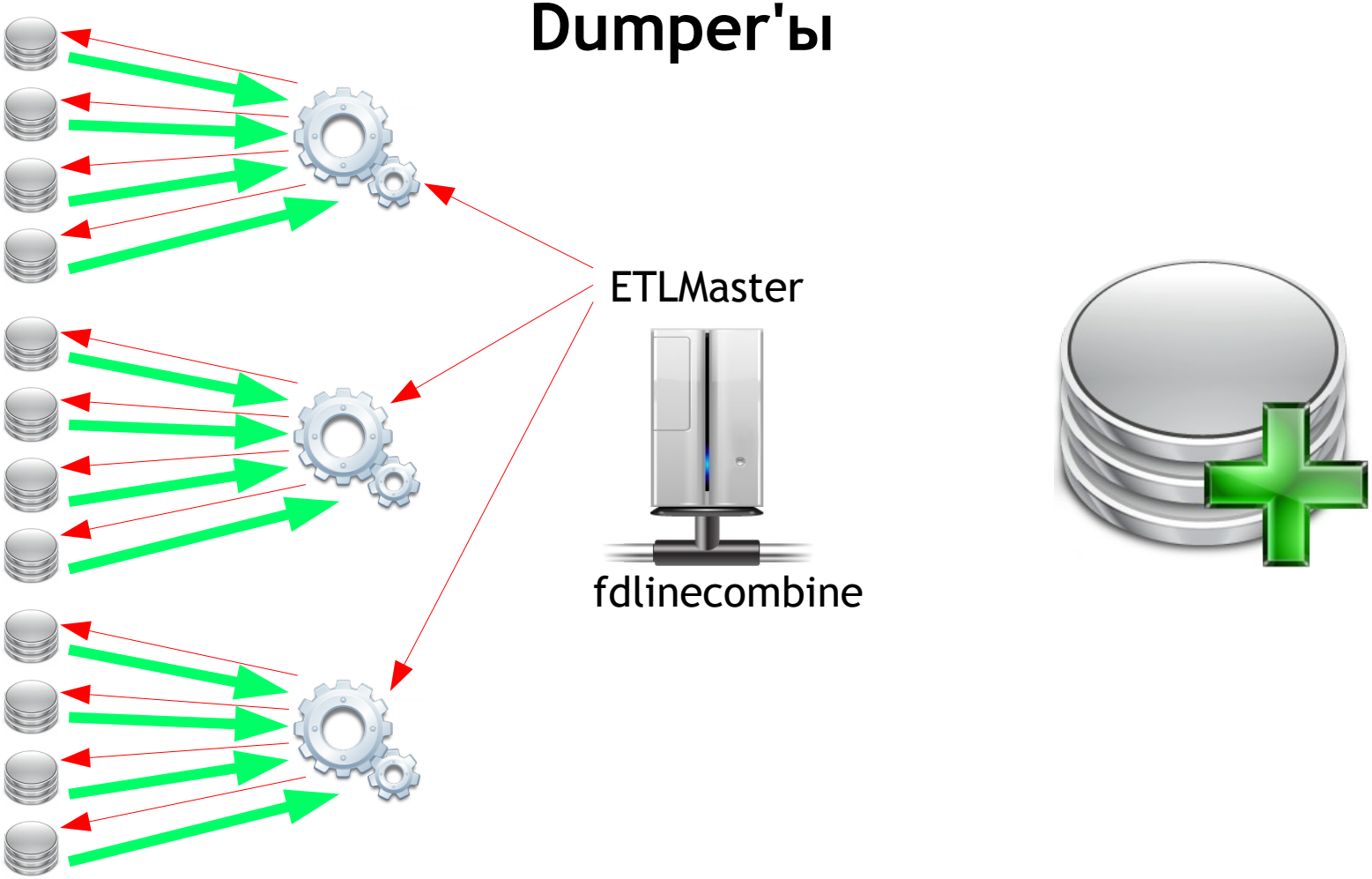


Dumper'ы

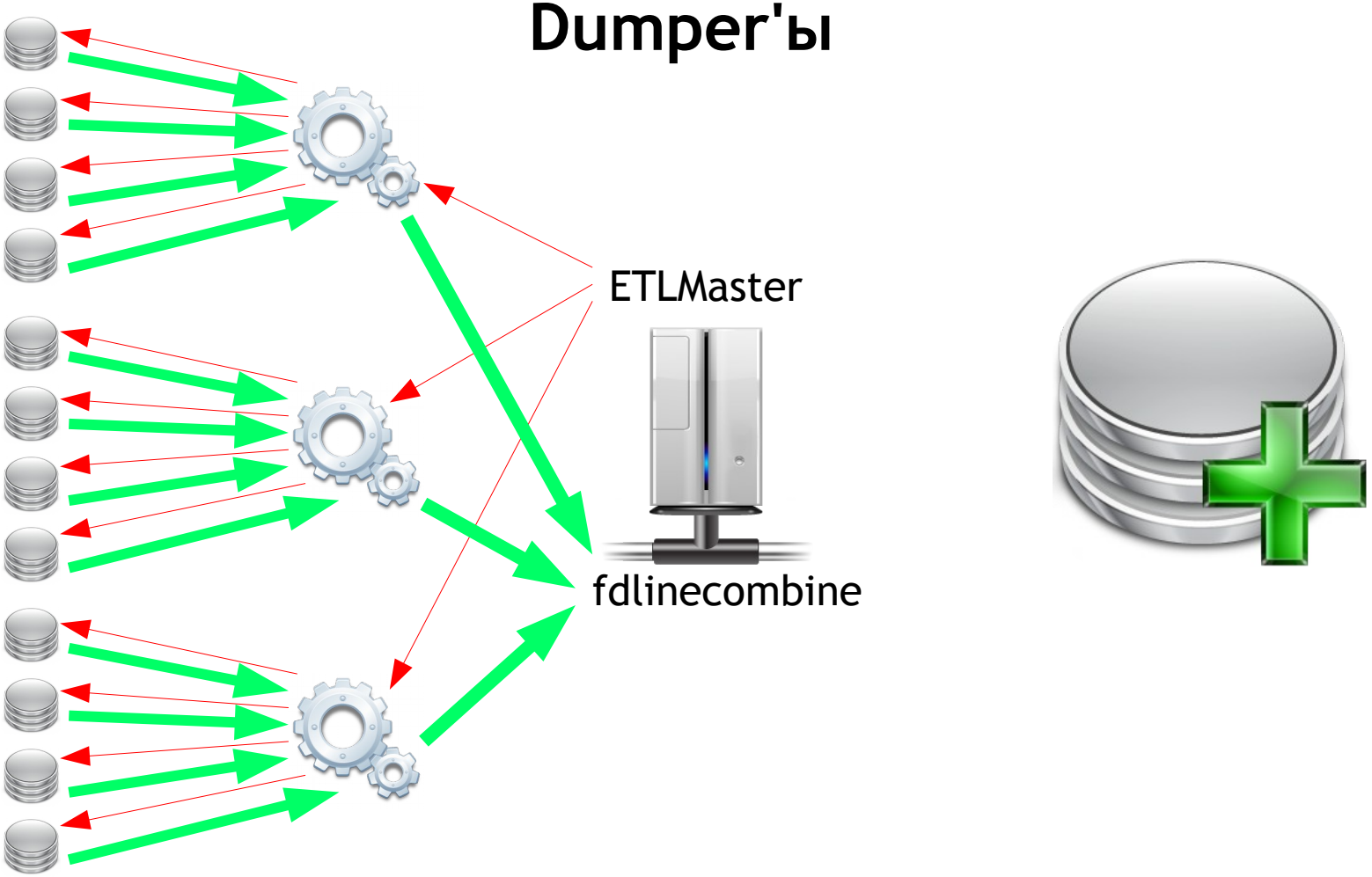


ETLMaster

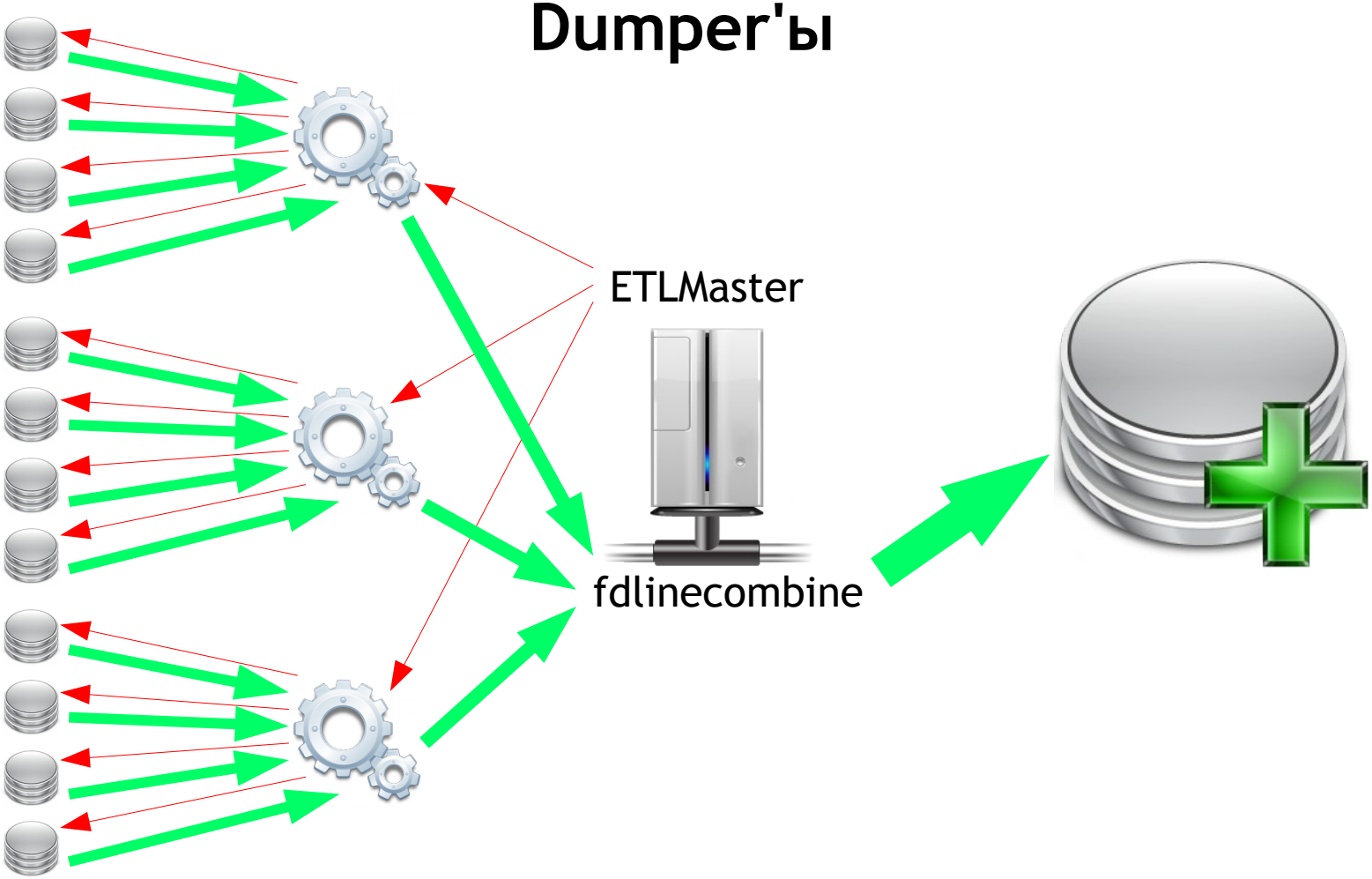
Dumper'ы



Dumper'ы



Dumper'ы

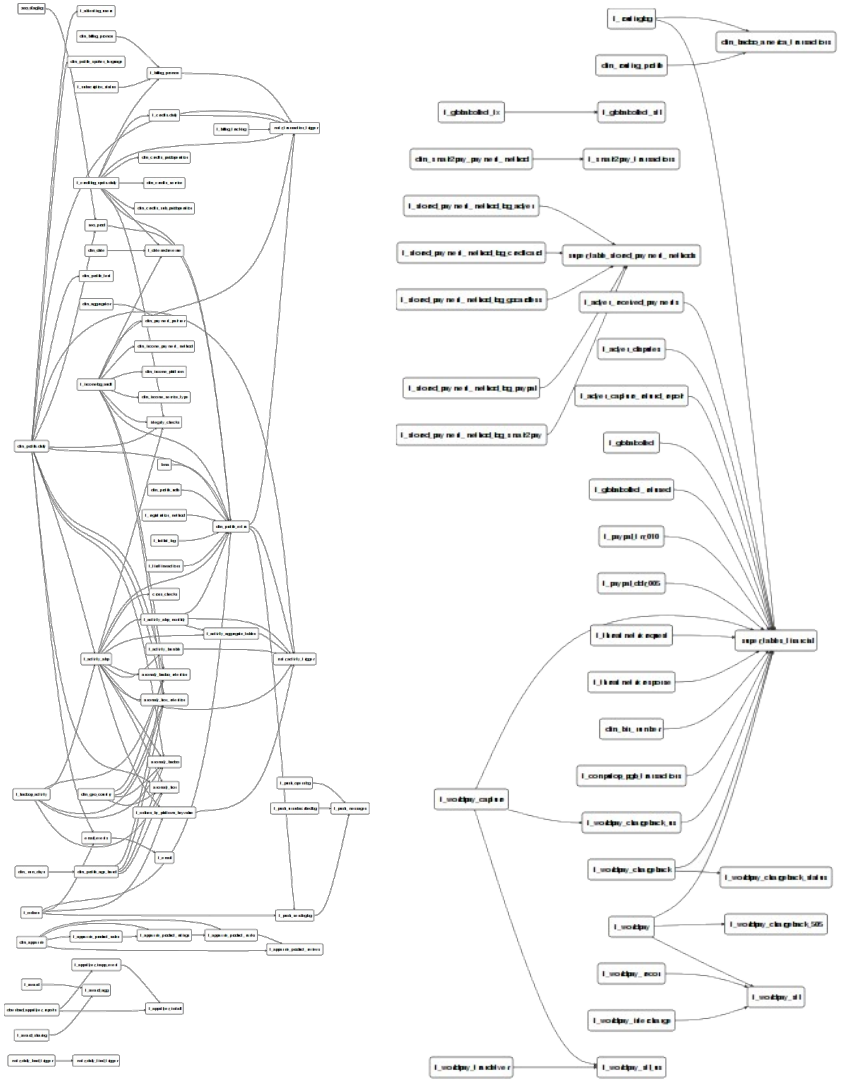


Возможности ETLMaster'a

- Ежедневная загрузка (старт по расписанию)
- Ожидание данных в источнике
- Ожидание выполнения зависимостей
- Возобновление загрузки после устранения проблем
- Перезагрузка данных за произвольный период с автоматическим запуском всех зависимых шагов

Пример зависимостей

```
"autorun": {
    "cron": "0 3 2 * *"
},
"wait": [
    {
        "name":"f_activity_abp",
        "type":"MaxOffset"
    },
    {
        "name":"f_incomelog_audit",
        "type":"FinishedToday"
    },
    {
        "name":"f_credits.daily",
        "type":"FinishedToday"
    }
],
```



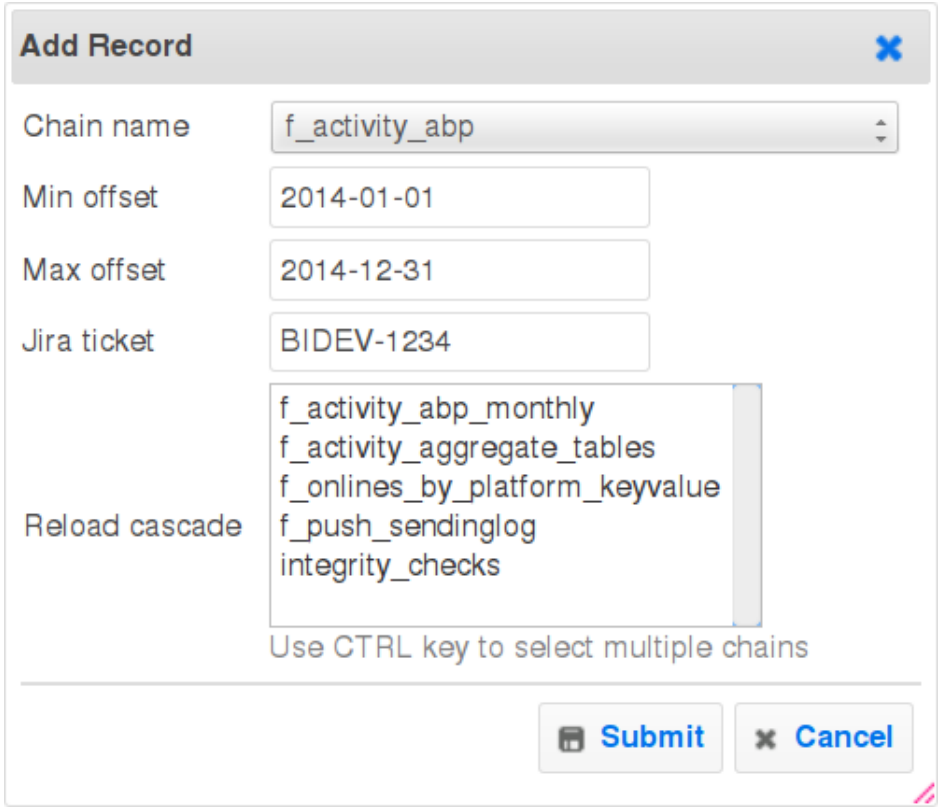
Какие стратегии загрузки применяем

- по периодичности — есть ежедневная и ежечасная загрузка
- по параллельности — данные можно грузить в 1 или несколько потоков
- инкрементально/перезагрузка всего/поддержание окна

Получение данных из данных

- Консолидация (объединение нескольких таблиц в одну)
- Агрегация (агрегация данных за месяц)

Перезагрузка и дозагрузка данных



Откуда мы собираем данные?

- MySQL — базы данных
- Шарды
- Данные из кода (конфиги)
- Hadoop
- Данные от сторонних компаний (api/sftp)

Сбор из MySQL, шардов и конфигов

- Собираем дамперами

Сбор из MySQL, шардов и конфигов

- Собираем дамперами
- В несколько потоков

Сбор из MySQL, шардов и конфигов

- Собираем дамперами
- В несколько потоков
- Указываем дамперу последовательность полей в результирующем потоке

Сбор из MySQL, шардов и конфигов

- Собираем дамперами
- В несколько потоков
- Указываем дамперу последовательность полей в результирующем потоке
- Получаем поток, готовый к загрузке

Сбор из MySQL, шардов и конфигов

- Собираем дамперами
- В несколько потоков
- Указываем дамперу последовательность полей в результирующем потоке
- Получаем поток, готовый к загрузке
- Иногда просто запускаем SQL, выдающий данные

Загрузка из Hadoop

- В старых данных нет новых столбцов

Загрузка из Hadoop

- В старых данных нет новых столбцов
- В первой строке каждого файла — заголовок (список колонок)

Загрузка из Hadoop

- В старых данных нет новых столбцов
- В первой строке каждого файла — заголовок (список колонок)
- Для каждого файла формируется команда обработки на awk

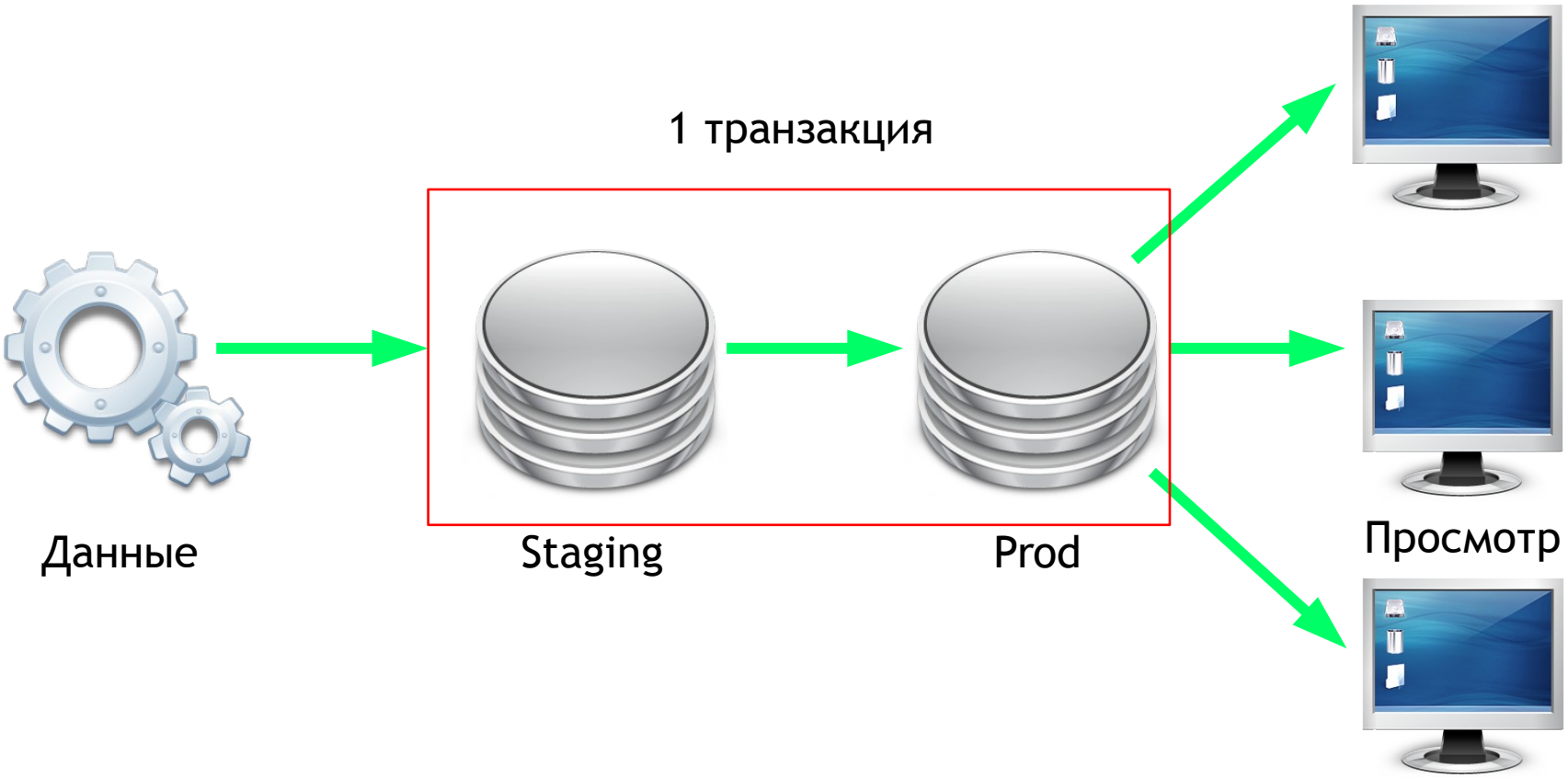
Загрузка из Hadoop

- В старых данных нет новых столбцов
- В первой строке каждого файла — заголовок (список колонок)
- Для каждого файла формируется команда обработки на awk
- Получаем готовый поток данных

Данные от сторонних компаний (api/sftp)

- Все данные нестандартизированы
- Приходится писать специфичные скрипты обработки
- Раньше приходилось обрабатывать отдельными скриптами, сохранять промежуточный результат и потом загружать; Теперь обрабатываем и загружаем сразу

Использование Staging—таблиц



Tips & Tricks

- Подгоняем данные под таблицу:
  - Изменяем порядок столбцов на требуемый
  - Подставляем отсутствующие колонки
  - Обрезаем строки по длине (только MySQL делает это сам)
- Обеспечиваем обновление данных за 1 транзакцию

Tips & Tricks

- Вычисления в MySQL — это быстрее, чем в PHP
- Выгрузка из MySQL быстрее всего — через passthru('mysql -s -q -e "SELECT * FROM DB.table"')
- Выгружаем из 1 таблицы в несколько потоков, когда упёрлись в CPU, а не в диск (т.е. почти всегда)

```
WHERE (id % 3)=0
WHERE (id % 3)=1
WHERE (id % 3)=2
```

Tips & Tricks

- ssh используем вот так:
  ssh -c arcfour -n -o BatchMode=yes -o StrictHostKeyChecking=no -T
- Сливаем потоки через fdlinecombine

Немного цифр

- Загрузка происходит за 4 часа
- Загружается 1,5 Tb данных
- 250 цепочек для запуска, примерно 600 задач
- 90% задач решается без написания кода на PHP
- 347 таблиц в аналитической базе

Спасибо за внимание!
Вопросы?

http://fb.com/BadooMoscow

http://vk.com/badoocom

http://twitter.com/BadooDev

http://habrahabr.ru/company/badoo