Кликстрим в Авито

Аналитика в высоконагруженном проекте

Дмитрий Хасанов

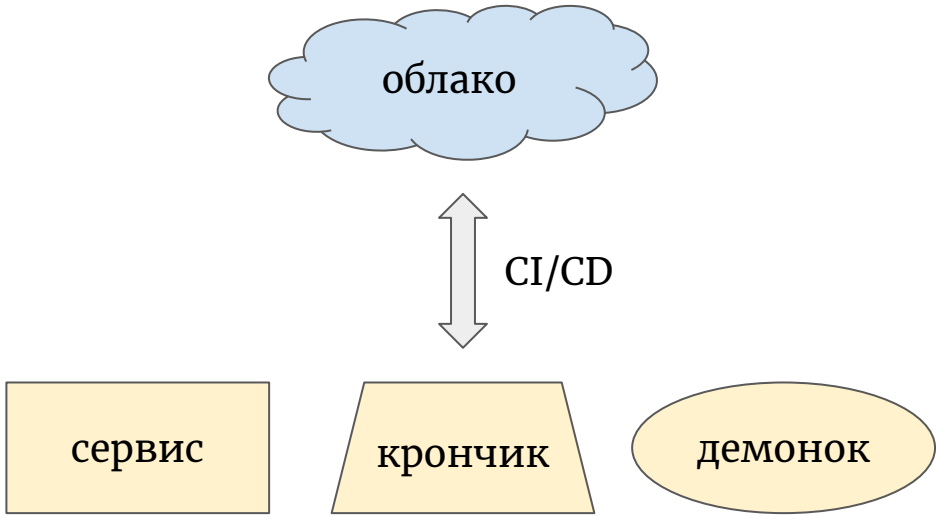
Обо мне

- фриланс
- классифайды
  n1.ru, auto.ngs.ru, avito.ru
- хакатоны
  Hack. Sleep. Repeat: goo.gl/bzMjjC

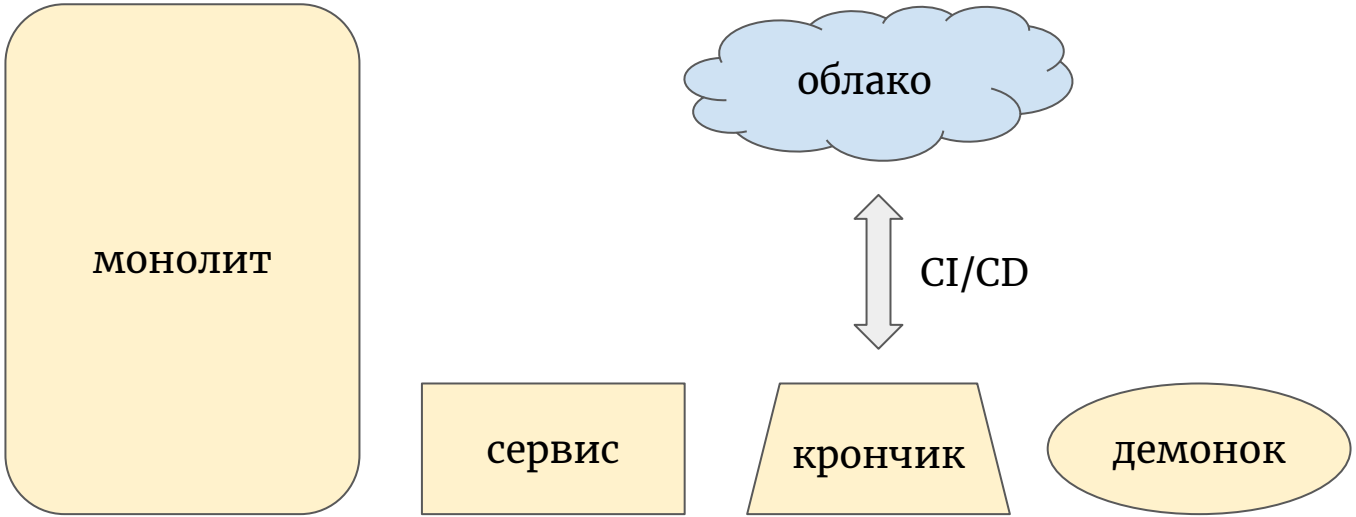
Авито

- топ 5 сайтов России по посещаемости
- аудитория 35 млн в месяц
- третья по стоимости компания Рунета
- мноооожество объявлений
- не только Авито

Сервисная архитектура Авито



Сервисно-монолитная архитектура Авито



Кликстрим

- клик
- просмотр
- оплата
- сообщение в мессенджере
- поисковый запрос
- рекламный аукцион

Готовые решения

- Google Analytics
- Яндекс Метрика

Кликстрим

Событие

Хранилище

Отчёт

Кликстрим

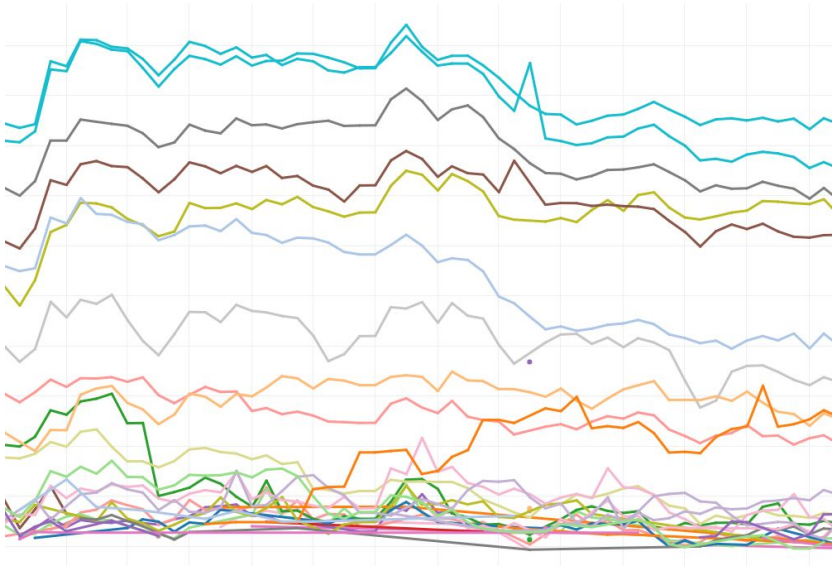
Событие

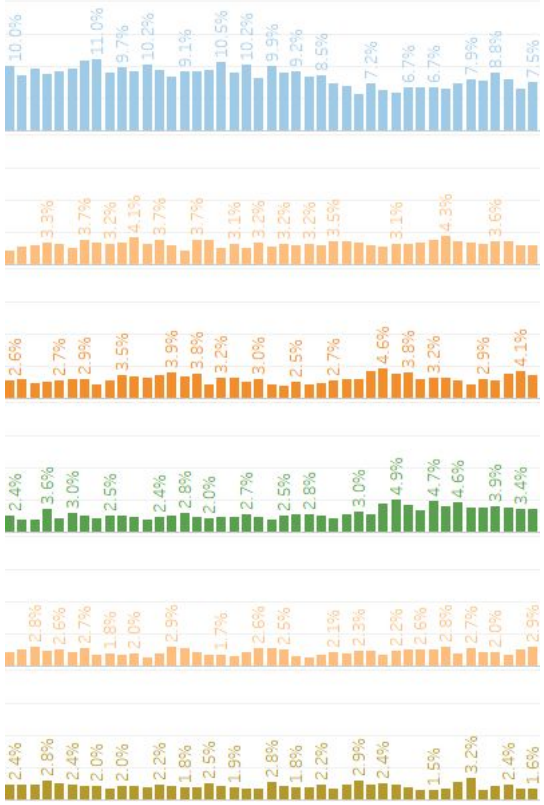
⇩

Хранилище

⇩

Отчёт





Кликстрим

Событие

Хранилище

Отчёт

```
“clickstream_event”: {
     “event_id”: 100,
     “user”: {
          “id”: 413,
          “email”: “user@example.com”
     },
     “geo”: {
          “latitude”: 20.220,
          “longitude”: -10.110
     },
     “user_agent”: “fluffy_browser”,
     “time”: “2018-04-11 22:37:05”
}
```

Кликстрим

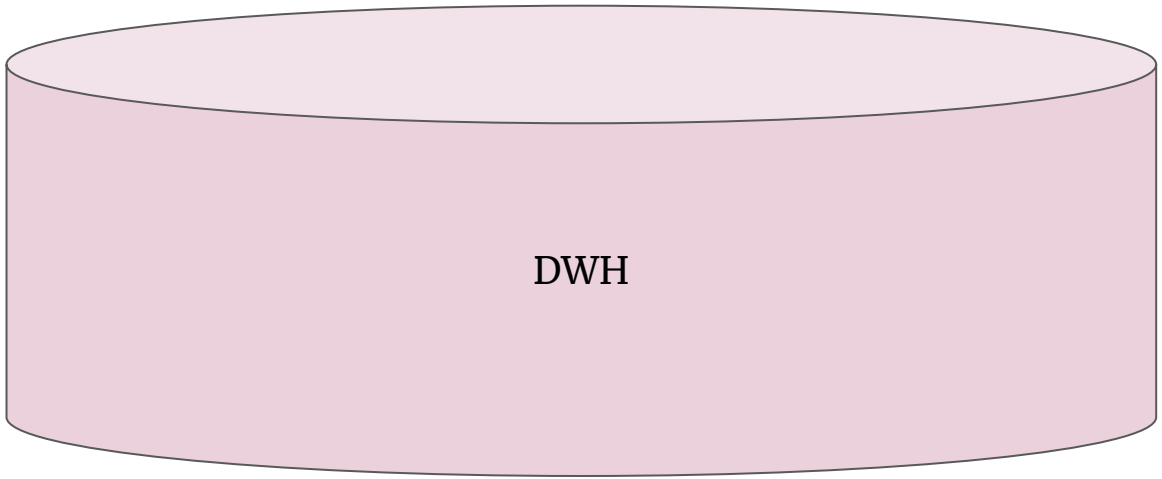
Событие

⇩

Хранилище

⇩

Отчёт



Кликстрим

Событие

Хранилище

Отчёт

Простая реализация

Проект А

```
event = {
    ‘field_one’: ‘val_1’,
    ‘field_two’: ‘val_2’,
    ‘time’: unixtime()
}
transport.send(event)
```

Проект Б

```
event = {
    ‘fieldOne’: ‘val_1’,
    ‘field.TWO’: ‘val_2’,
    ‘time’: date()
}
customTransport.send(event)
```

Простая реализация

Плюсы:

- быстро получаем отчёт
- легко реализовать

Минусы:

- бардак
- затраты ресурсов
- трудно переиспользовать код

Источники событий

Бэкенды:

- монолит: php
- сервисы: go, python, php
- кроны, демоны: go, python, php, shell

Источники событий

Бэкенды:

- монолит: php
- сервисы: go, python, php
- кроны, демоны: go, python, php, shell

Фронтенды: js

Источники событий

Бэкенды:

- монолит: php
- сервисы: go, python, php
- кроны, демоны: go, python, php, shell

Фронтенды: js

Мобильные приложения: swift, java

Источники событий

Бэкенды:

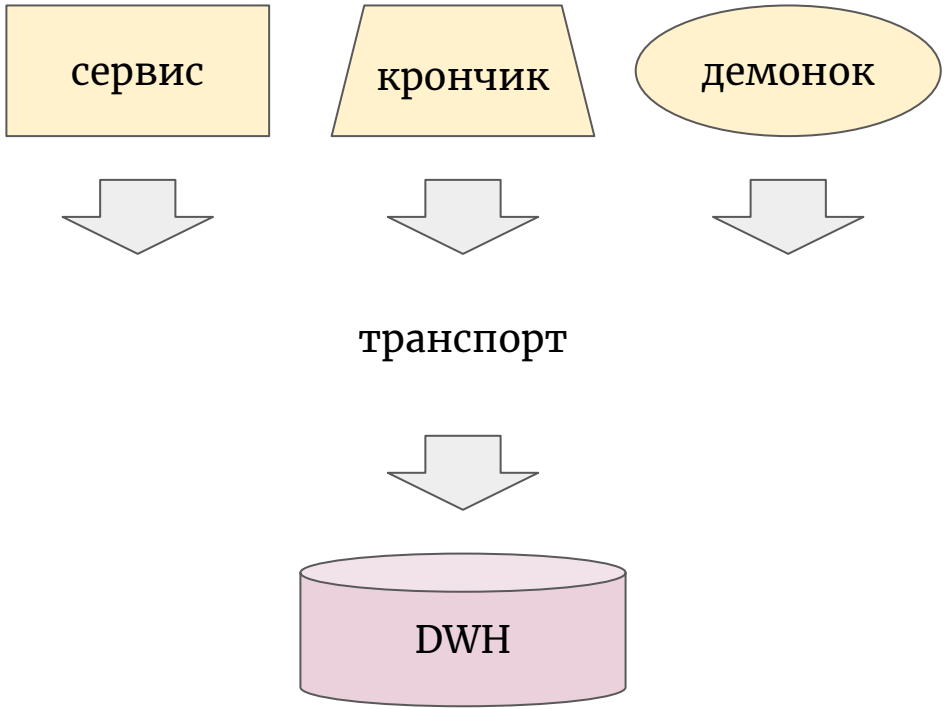
- монолит: php
- сервисы: go, python, php
- кроны, демоны: go, python, php, shell

Фронтенды: js

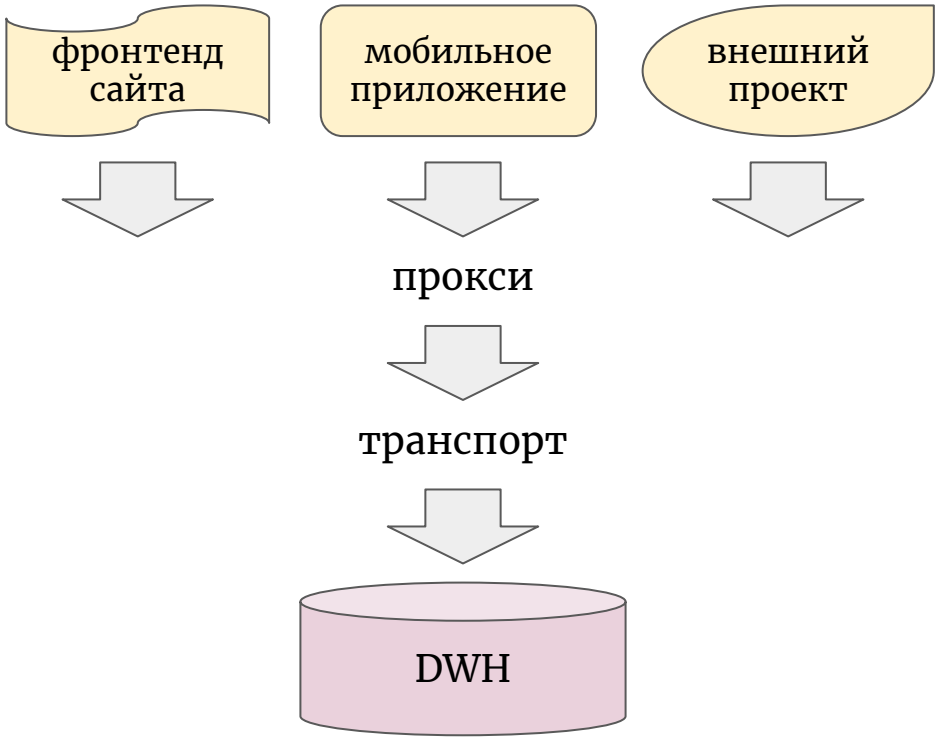
Мобильные приложения: swift, java

Внешние проекты: c#, java

Путь события



Путь события



Реестр событий

- окружения
- события
- поля
- метаинформация

Кодогенерация

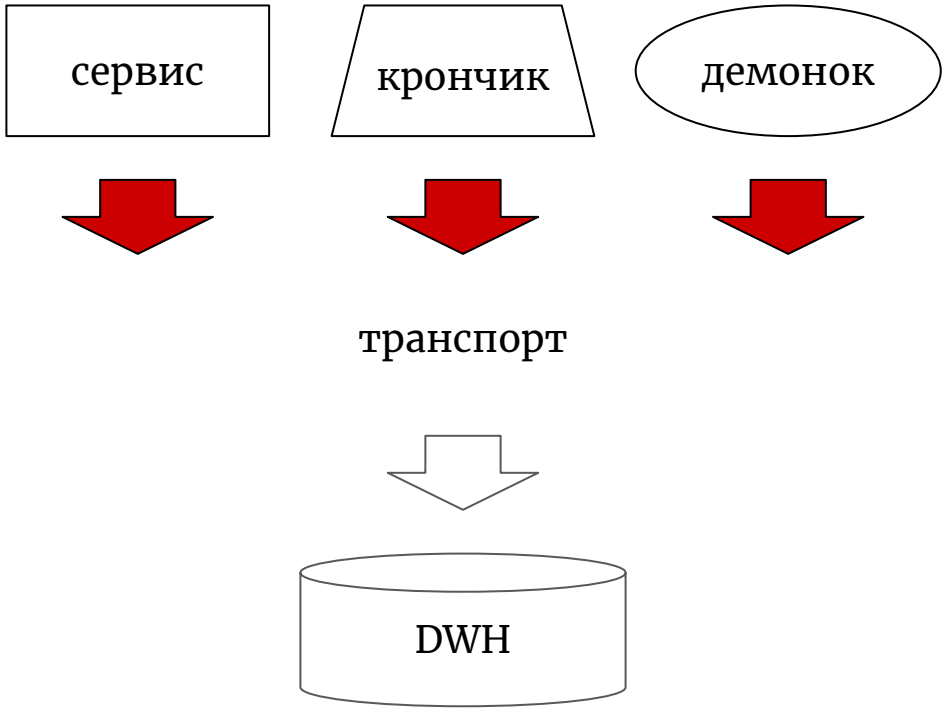
- лангпак php, go, python
- отправщик событий в DWH
- автодокументация

Лангпак

- геттеры, сеттеры
- общая логика

```
class EventOne:

def setFieldOne(string value):
        self.fields[“one”] = value

def getFieldOne() -> string:
        return self.fields[“one”]

def getTime() -> Timestamp:
        return time.now()

def getData() -> EventData:
        return self.fields
```

Отправщик событий



Транспорт

- NSQ: https://github.com/nsqio/nsq
- поддержка Fluent-протокола
- консьюмеры по запросу

Пример лангпака

```
package someproject

type SomeEventV0 struct {
	*event
}

func NewSomeEventV0() *SomeEventV0 {
	e := &SomeEventV0{
		event: new(
			420,
			`user@example.org`,
			`lindows`,
		),
	}

	e.required["user_id"] = struct{}{}
	e.required["email"] = struct{}{}
	e.required["os"] = struct{}{}
	return e
}
```

Версионирование событий

- нельзя ломать код на бою
- нельзя удалять
- новая версия события на каждое изменение

Версионирование лангпаков

- нельзя ломать код на бою
- новая версия на каждое изменение кода лангпака

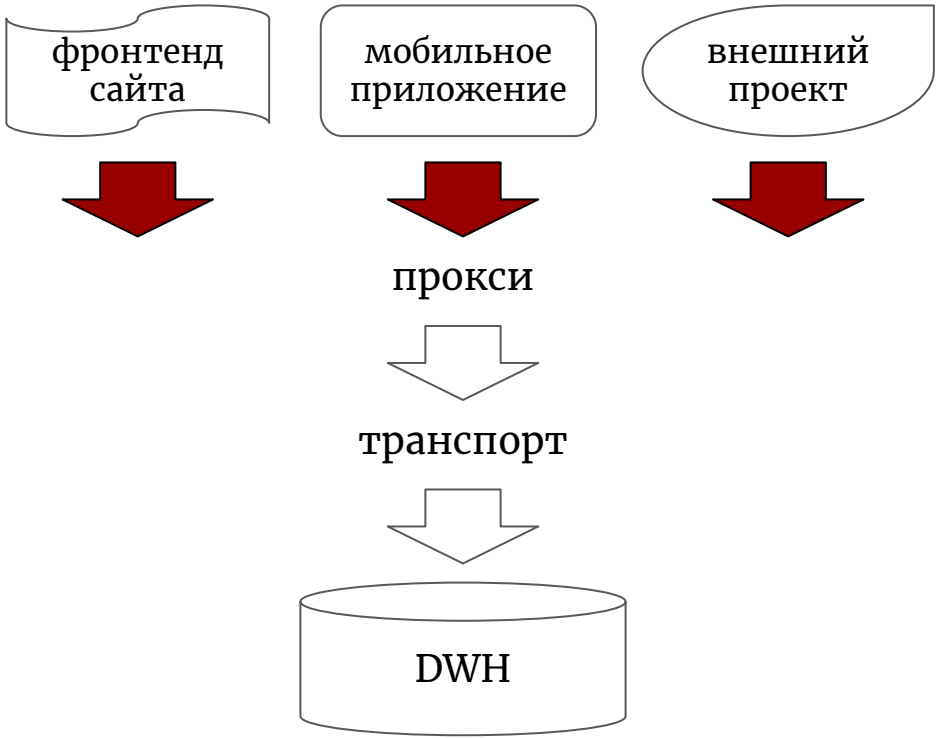
Получение лангпака

- php
  ```
  composer install reestr/langpack
  ```
- go, python
  ```
  curl \
      -d '{"environment":"bo"}' \
      -H "Content-Type: application/json" \
      http://reestr/langpack/go/0/ | tar x
  ```

Отправка событий

- php
  ```
  $clickstreamSender->send($event);
  ```
- go
  ```
  err := clickstreamSender.Send(event)
  ```
- python
  ```
  clickstream_sender.send(event)
  ```

Протохитрости



Протохитрости

Внешние проекты

```
message ExternalProjectEvent {
        int32 someIntField = 1;
        string someStringField = 2;
        bool someBoolFied = 3;
}
```

Протохитрости

Внешние проекты

```
message ExternalProjectEvent {
        int32 someIntField = 1;
        string someStringField = 2;
        bool someBoolFied = 3;
}
```

Мобильные приложения

```
message MobileAppEvent {
    int32 eventId = 1;
    int32 version = 2;
    int32 timestamp = 3;
    map<string,string> params = 4;
}
```

Объекты

```
event = {
    ‘scalar_field’: ‘some_string’,
    ‘object_field’: {
        ‘inner_scalar’: 3.14,
        ‘inner_object’: { … }
    }
}
```

Плюсы:

- наборы объектов

Комплексность:

- лангпаки
- админка реестра
- валидация

Административности

- идентификация владельцев событий
- отслеживание девиаций: технических, логических
- вывод из эксплуатации неиспользуемых событий
- интеграция в работающие проекты

Настоящее

- десятки источников
- более тысячи версий событий
- логируется около двух миллиардов событий в сутки

Будущее

- ускорение создания событий
- нативные пакеты с лангпаками для python и go
- статус отправки события
- доменная модель компонентов

Спасибо

pik4ez@gmail.com
github.com/pik4ez