

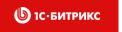


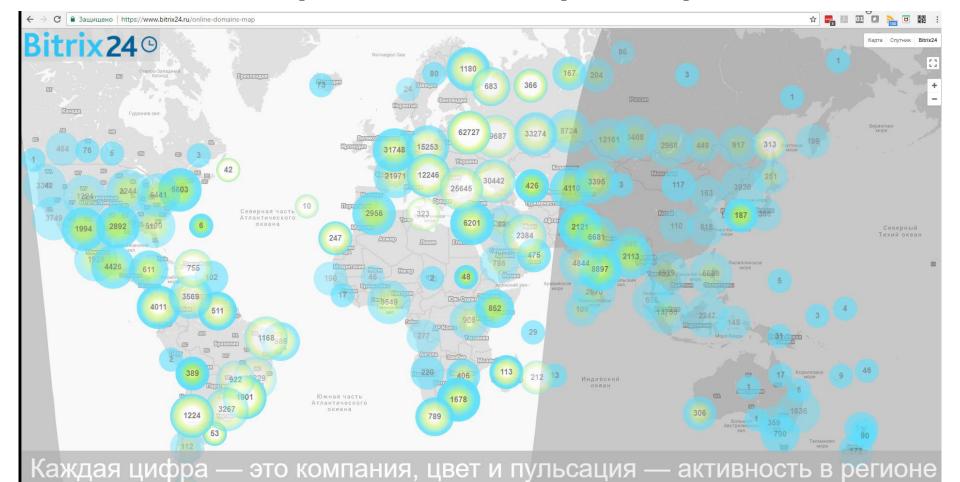


Зачем?

- 1. Удобные облачные объектные хранилища (HTTP)
- 2. Более 500 ТБ файлов клиентов в Amazon S3; > 500 млн. файлов
- 3. Катастрофа, удар молнии
- 4. Военные действия
- 5. Ошибки в приложении, суицид

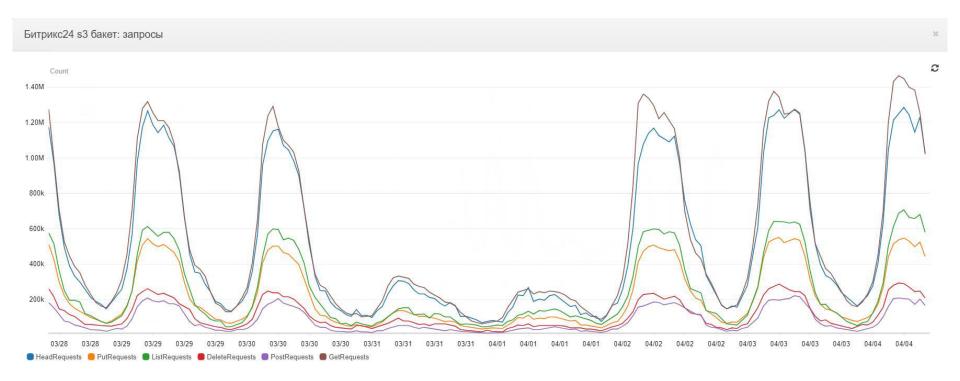
Клиенты Битрикс24 – на карте мира

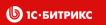




() 1С-БИТРИКС

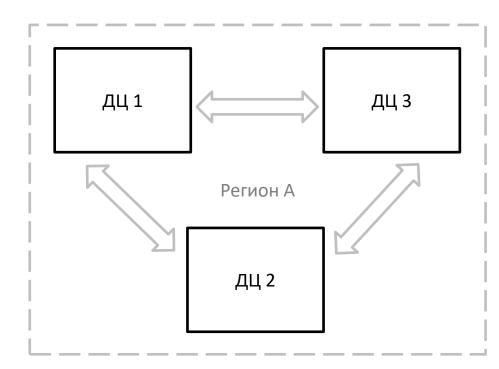
Amazon S3 – в Битрикс24





А вдруг ... катастрофа?

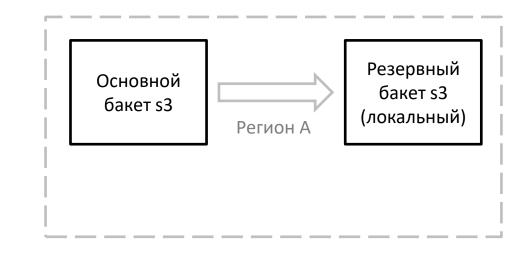
- 1. Amazon S3
- 2. Распределенная ФС с дублированием
- 3. Удар молнии в ДЦ Amazon и последствия
- 4. Открытые аналоги: Apache Hadoop





Ошибки в коде приложения

- 1. Копирование в локальный бакет s3
- 2. Ошибки в приложении (ошибка с «очисткой по префиксу»)





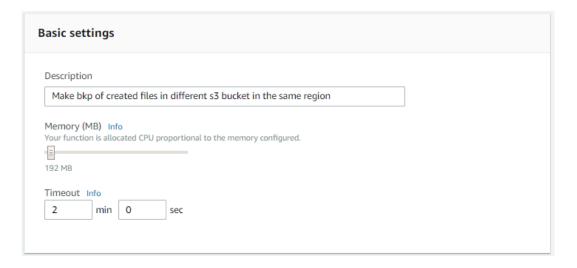
Техника копирования. Hadoop.

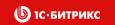
- 1. Выгружаем список названий файлов бакета s3 в текстовые файлы
- 2. >500 миллионов файлов
- 3. Запускаем Hadoop MapReduce jobs для параллельного копирования файлов из бакета A в бакет Б Несколько дней, PHP.

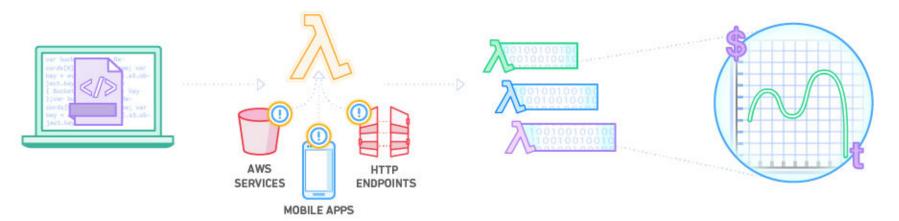




- 1. Микросервисы
- 2. Ограничение по памяти и времени выполнения
- 3. Расчет стоимости, сравнение с арендой облачного сервиса
- 4. Программная модель, языки программирования

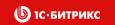






Upload your code to AWS Lambda Set up your code to trigger from other AWS services, HTTP endpoints, or in-app activity

Lambda runs your code only when triggered, using only the compute resources needed Pay just for the compute time you use



Example: Image Thumbnail Creation

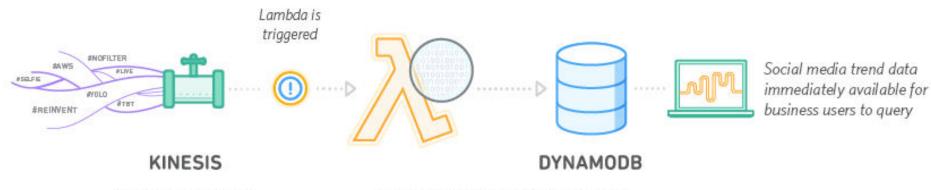


Photo is uploaded to S3 Bucket

Lambda runs image resizing code to generate web, mobile, and tablet sizes



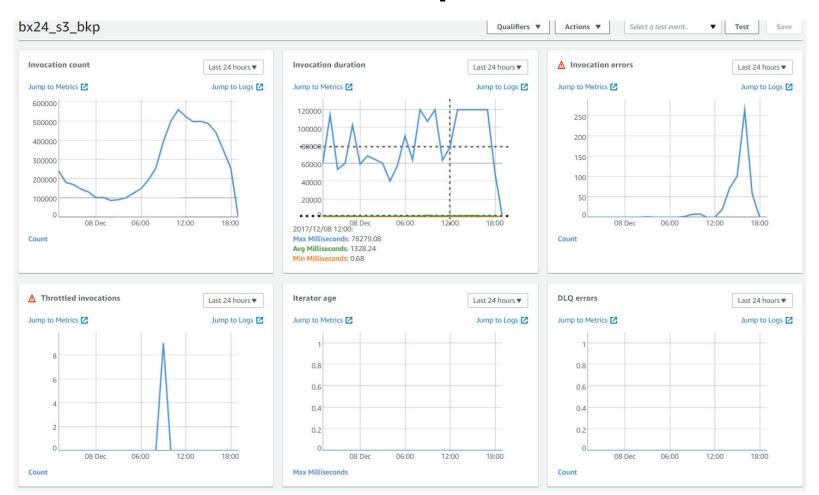
Example: Analysis of Streaming Social Media Data

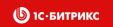


Social media stream is loaded into Kinesis in real-time. Lambda runs code that generates hashtag trend data and stores it in DynamoDB

Amazon Lambda – мониторинг

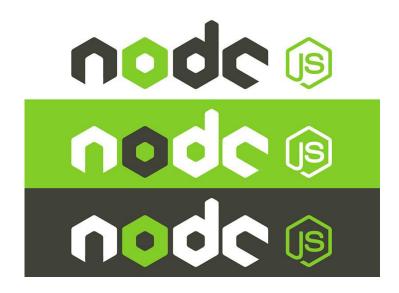


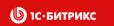




Amazon Lambda – версия 1, Node.JS

- 1. Кратко про технологию, асинхронность
- Пакеты (npm)
- 3. Контингент
- 4. Просто
- 5. Быстро
- 6. Глючит окружение в Амазоне
- 7. Спонтанное падение воркеров
- 8. Теряются события





Node.JS, npm и ... потроха



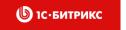


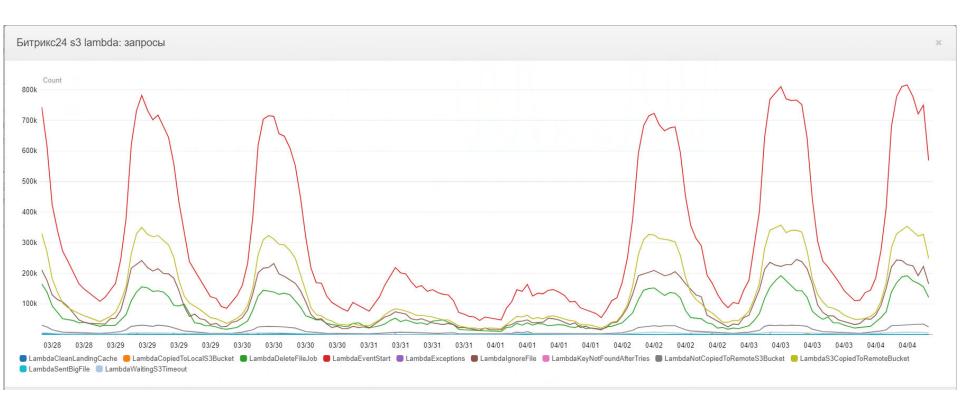
Amazon Lambda – версия 2, Java

- 1. Строгая типизация, отточенные промышленные стандарты
- 2. Великолепные среды разработки
- 3. Скорость
- 4. Много кода
- Продолжают теряться события, но гораздо меньше



Amazon Lambda – аналитика







Amazon Lambda – подводные камни

- 1. Heт SLA
- 2. Мониторинг
- 3. Спонтанные падения воркеров (Node.JS)
- 4. Версии и обновление сервисов, непрозрачность документации
- 5. Потеря пакетов, сверка и синхронизация остатков
- 6. Есть ли альтернатива?



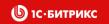
Amazon Lambda – dead letter queue (DLQ)

- 1. Если сообщение не доставлено в Lambda 2 раза
- 2. Попробуй его записать в DLQ SQS
- 3. Если не получилось (?) увеличь счетчик ошибок DLQ
- 4. Недоставленные сообщения нужно разбирать отдельно
- 5. Кейс ошибка при обновлении Lambda



Amazon Lambda – сервис поддержки

- 1. Что делать с «большими» файлами, которые сильно удорожают использование Lambda?
- 2. Сервис «поддержки» Amazon Lambda:
- копирование больших файлов
- до-отправка DLQ записей
- отложенное удаление
- ревалидация по внутренним файловым событиям ядра Битрикс24



Сервис поддержки – структура кода

- → bigdata-bx24-s3-bigfile-remote-backuper
- ~ ® src/main/java
- ~ com.bitrix24.backup.s3
- AwsUtils.java
- BigFileDownloadUploadTask.java
- BigFileQueueReader.java
- BigFilesRemoteBackuper.java
- Bx24PHPEngineEventFileInfo.java
- Bx24PHPEngineEventProcessingTask.java
- Bx24PHPEngineEventQueueReader.java
- DerbyRegistry.java
- DftEventProcessingTask.java
- DftTableReader.java
- DlqEventProcessingTask.java
- DlgFileInfo.java
- DlqQueueReader.java
- FileInfo.java
- HashMapRegistry.java
- Lambda.java
- ► # LocalS3FilesRegistry.java
- B S3BucketDeleter.java
- S3BucketDumper.java
- S3BucketStat.java
- 🤊 🖟 S3Differ.java
- S3KeyData.java
- [™] S3KeyDataProcessor.java
- S3KeyInventoryJsonDataSource.java
- SftEventProcessingTask.java
- SftTableReader.java
- src/test/java
- JRE System Library [JavaSE-1.8]
- Maven Dependencies
- src
- * target
- ≅ pom.xml

- 1. Apache Maven
- 2. Интенсивное использование "concurrency framework" и "collections framework"
- 3. Параноидальный код, продуманная работа с exceptions
- 4. Разные уровни логирования с slf4j + log4j
- 5. Юнит-тесты и частично интеграционные с Mockito.
- 6. В ближайший планах кластер с Apache ZooKeeper.



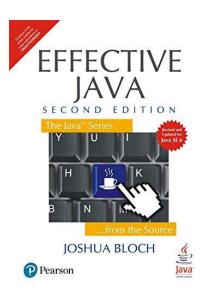
Сервис поддержки – техники программирования

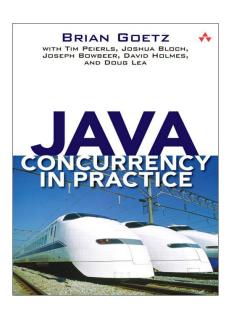
- 1. Готовые библиотеки, где можно: maven shade, commons-cli, commons-validator, commons-io, org.json, junit, org.mockito, com.opencsv.
- 2. Иммутабельные объекты, где возможно
- 3. Билдеры и фабрики, где удобно, вместо конструкторов. Инкапсуляция/делегирование вместо наследования.
- 4. Константность, где только можно через "final"
- 5. Инкапсуляция, где имеет смысл и удобно
- 6. Проверка входных параметров в публичных и ... остальных методах
- 7. Интенсивное использование Enums для перечислений и сопоставлений с образцом
- 8. Unit-тесты где можно и разумно



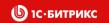
Книжки по программированию, которые стоит прочитать хоть раз









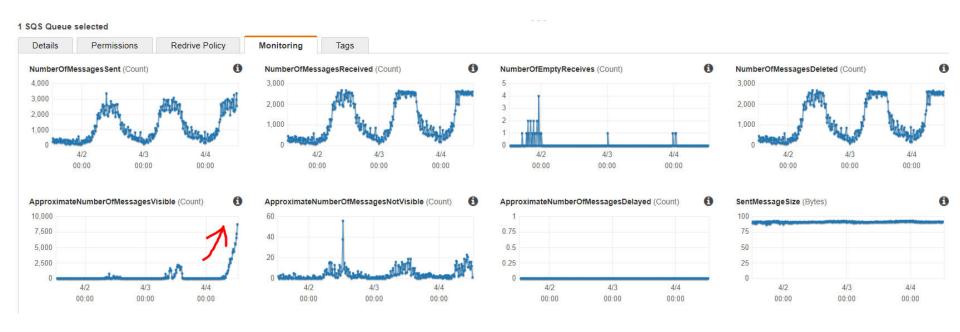


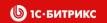
Сервис поддержки – копирование «больших» файлов

- 1. Читает SQS-очередь «больших» файлов, получает задание
- 2. Захватывает семафор над пулом копирующих потоков (ExecutorService)
- 3. Пытается скопировать файл в облако mail.ru напрямую через сокетсокет, без заливки на диск
- 4. Удаляет SQS-задание из очереди
- 5. Освобождает семафор



Сервис поддержки – копирование «больших» файлов





Сервис поддержки – дозаливка событий из DLQ

- 1. Читает SQS-очередь «dead letter queue» с именами недоставленных Amazon Lambda файлов, получает задание
- 2. Захватывает семафор над пулом копирующих потоков (ExecutorService)
- 3. Пытается скопировать файл в облако mail.ru напрямую через сокетсокет, без заливки на диск
- 4. Удаляет SQS-задание из очереди
- 5. Освобождает семафор

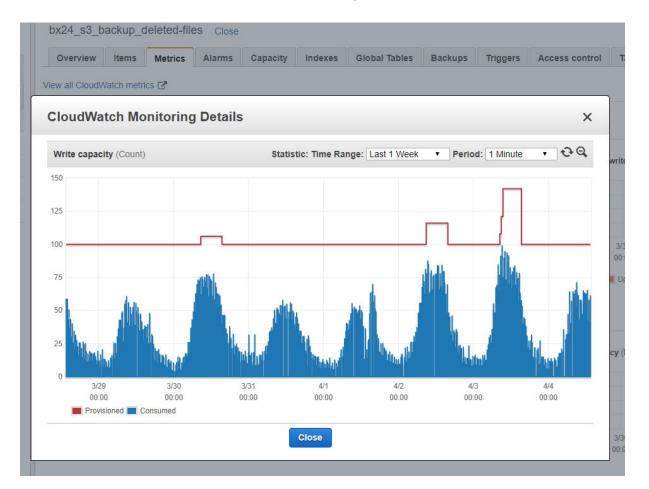


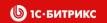
Сервис поддержки – отложенное удаление файлов

- 1. Читает Amazon DynamoDB таблицу (выборка событий месячной давности), получает задание
- 2. Захватывает семафор над пулом удаляющих потоков (ExecutorService)
- 3. Пытается удалить файл из бакета mail.ru
- 4. Удаляет задание из таблицы
- 5. Освобождает семафор



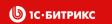
Сервис поддержки – отложенное удаление файлов



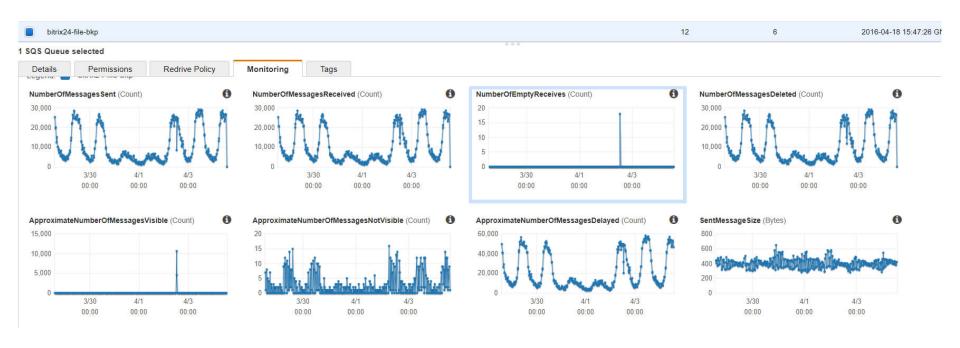


Сервис поддержки – корректировка файловыми событиями ядра Битрикс24

- 1. Читает Amazon SQS очередь файловых событий ядра Битрикс24, получает задание
- 2. Захватывает семафор над пулом копирующих потоков (ExecutorService)
- 3. Пытается скопировать файл в бакет mail.ru
- 4. Удаляет задание из SQS очереди
- 5. Освобождает семафор



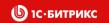
Сервис поддержки – корректировка файловыми событиями ядра Битрикс24





Проект по миграции >500 млн. файлов, >500 ТБ

- 1. Сканируются все записи в s3 Битрикс24 в один поток
- 2. Имена файлов записываются в DynamoDB таблицу на разные шарды для повышения скорости записи
- 3. Модуль в один поток читает записи с рандомной шарды и увеличивает номер шарды (при достижении предельной шарды начинает с рандомной шарды)
- 4. Часть чтений начинается с 0 шарды (чтобы захватить начальный «хвост»)
- 5. Модуль захватывает семафор и ставит задание в многопоточный ExecutorService
- 6. Задание на копирование выполняется в отдельном потоке. Успешное выполнение– удаление записи в DynamoDB таблице
- 7. Модуль освобождает семафор



S3->mail.ru – ресинхронизация

- 1. Список файлов >500 миллионов штук в двух s3 хранилищах
- 2. BerkeleyDB
- 3. SQLite
- 4. LevelDB
- 5. LMDB
- 6. Amazon Inventory
- 7. Варианты развития архитектур



S3->mail.ru: ресинхронизация

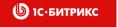
- 1. Читаем список файлов каждого портала в бакете s3 и сохраняем в конкурентной хэш-таблице в памяти (ConcurrentHashMap)
- 2. Читаем список файлов этого же портала в бакете mail.ru и пересекаем с хэш-таблицей в памяти
- 3. Пункты 2-3 делаются многопоточно
- 4. Оставшиеся файлы в памяти копируем в одном направлении: s3 –> mail.ru
- 5. Ведем детальное логирование
- 6. Запускаем раз в неделю

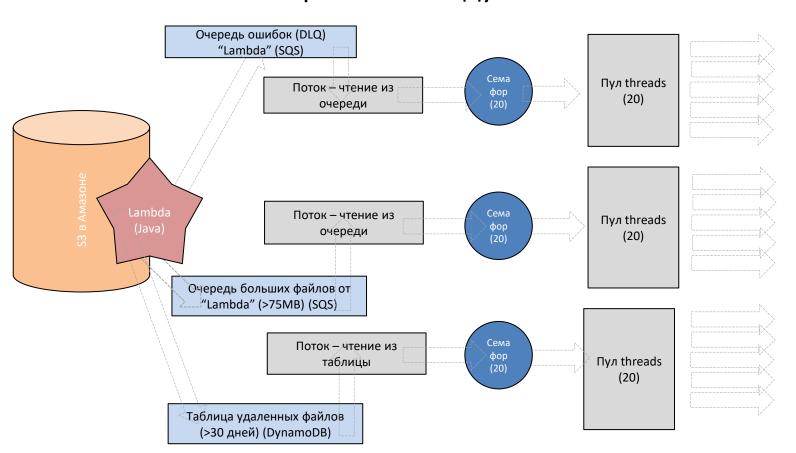


Ближайшие планы

- 1. Внедрить систему кластерной координации Apache ZooKeeper
- 2. Реализовать средства межмодульного взаимодействия: конфигурация, выбор лидера, очереди, блокировки, двух-фазный коммит
- 3. Быть готовыми масштабироваться на 2-100-1000 spot-машин!

Архитектура **репликатора файлов** клиентов Битрикс24 между континентами







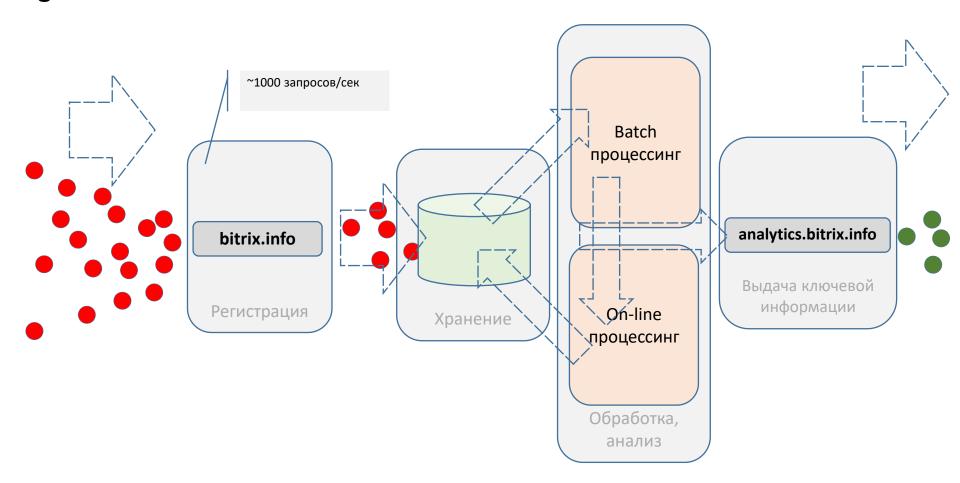


Сервис «Скорость сайта» клиента на Битрикс

- 1. Собираем хиты из Amazon Kinesis в Redis
- 2. Хит содержит метрики js navigation timing.
- 3. Храним последние 1250 хитов (redis list)
- 4. Удаляем 20% самых долгих
- 5. Рассчитываем медиану времени отображения страницы в кластере
- 6. Отдаем на карту, jsonp, RemoteObjectManager

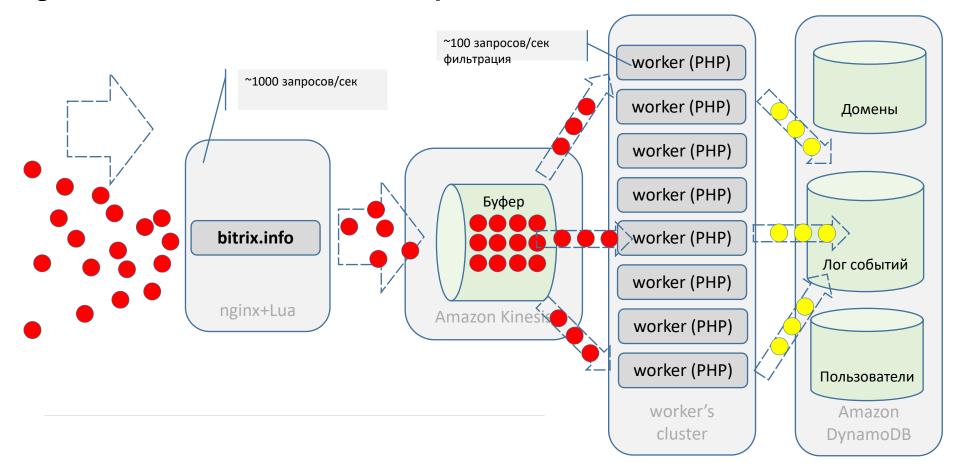


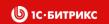
BigData – «под капотом». С птичьего полета.



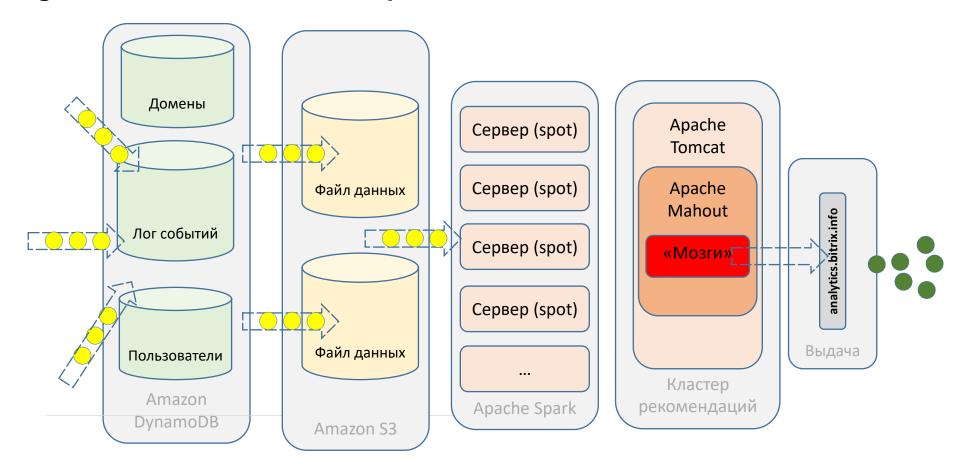


BigData – «под капотом». Регистрация событий.





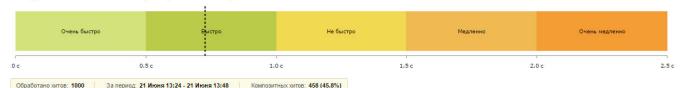
BigData – «под капотом». Обработка, анализ, выдача.



() 1С-БИТРИКС

Сервис «Скорость сайта»

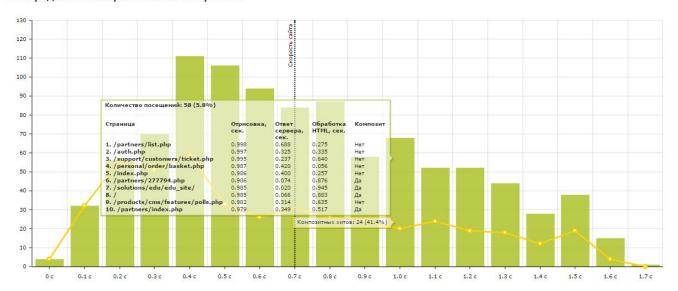




Скорость сайта — комплексный показатель комфортности работы с сайтом для посетителей.
Учитывает качество разработки сайта, качество хостина и доступность сайта по сели. Рассчитывается для 1000 последних посетителей вашего сайта.
Скорость сайта фактически показывает как быстро отобразился ваш сайт для большинства из этих 1000 посетителей.

Монитор производительности: 15.57 Композитный сайт: включено Ускорение сайта (CDN): включено

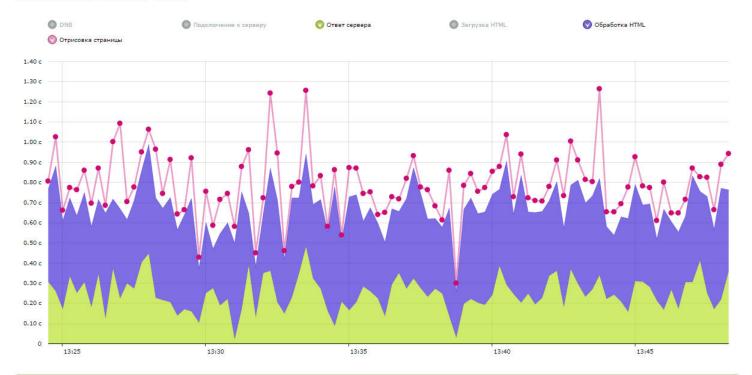
Распределение скорости сайта по времени



Сервис «Скорость сайта»

७ іс∙битрикс

Последние посещения сайта



Отрисовка страницы — время от начала перехода на страницу до появления её на экране. Именно по этому показатели считается Скорость сайта.

DNS — время выполнения запроса DNS для страницы.

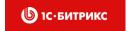
Подключение к серверу — сколько времени компьютер пользователя устанавливает соединение с сервером.

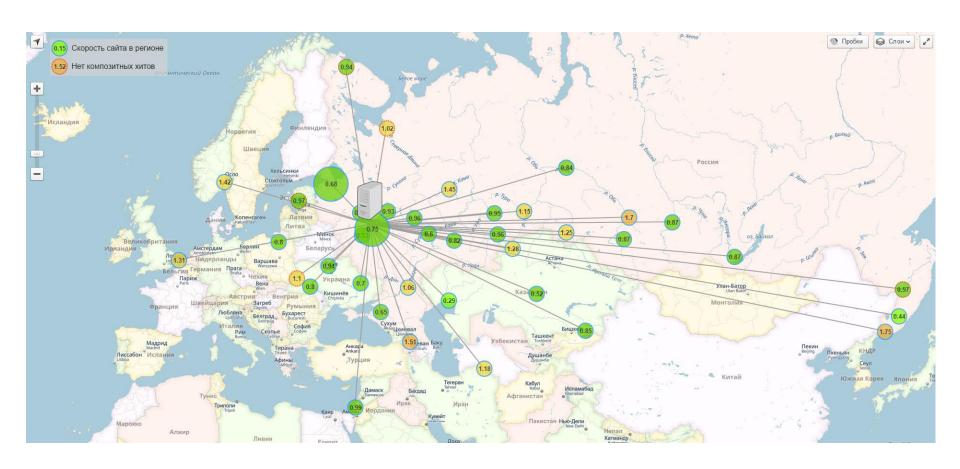
Ответ сервера — время обработки сервером запроса пользователя (включая время реакции сети для местоположения пользователя).

Загрузка HTML — время загрузки HTML страницы без ресурсов (картинки, CSS, Javascript).

Обработка HTML — время, в течение которого браузер обрабатывал содержимое страницы (синтаксический анализ HTML, CSS, обработка элементов JavaScript и отображение страницы) после загрузки её с сервера и до начала отрисовки.

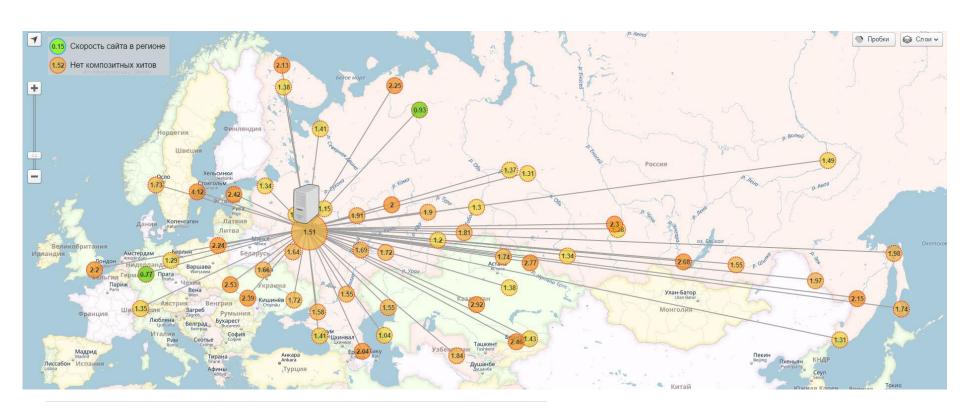
Сервис «Скорость сайта»





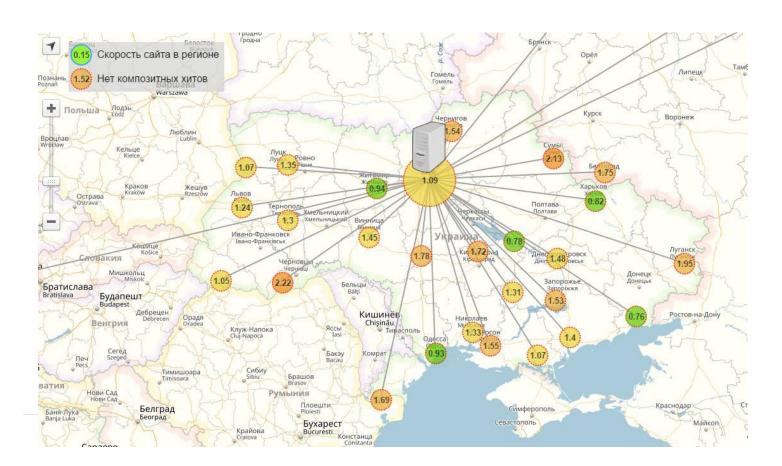
Сервис «Скорость сайта»





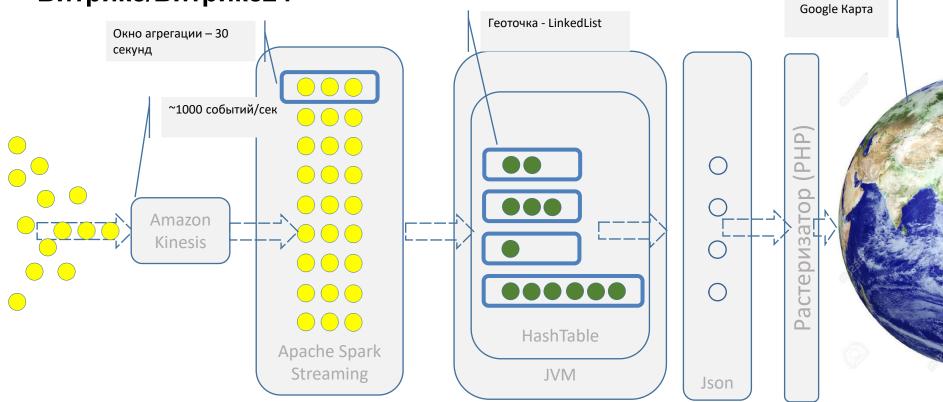
() 1С-БИТРИКС

Сервис «Скорость сайта»



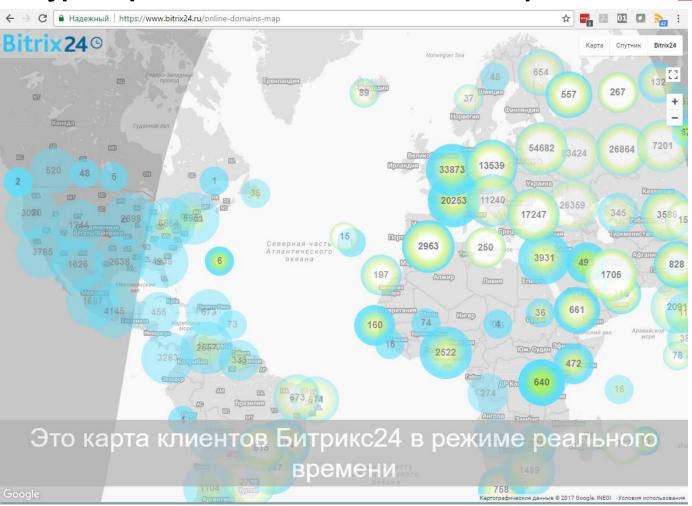


Архитектура карты активности клиентов Битрикс/Битрикс24



Архитектура карты активности клиентов Битрикс24

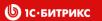






Кластерный анализ в Битрикс

- События использования инструментов на Битрикс24: задачи, вики, видеозвонки, календарь, чаты, поиск...
- Агрегация метрик по пользователям: Amazon Kinesis -> Amazon DynamoDB -> s3 -> Apache Spark
- Apache Spark Mllib, стандартный k-means не взлетает,
 O(n^3)
- «Иерархический k-means с усреднением», Scala, Spark
- На выходе 3-4 группы = типа использования продукта



Коллаборативная фильтрация (Item-Item) в Битрикс

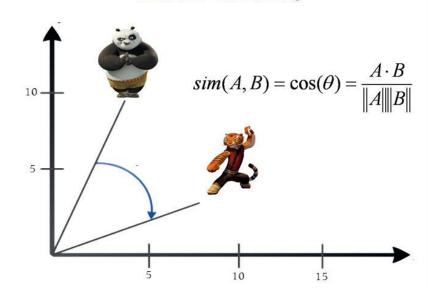
- Были модели по 10-20 миллионов Пользователей/Товаров
- Фильтрация, создание датасета для матрицы: s3->Apache Spark job.->Apache Mahout Taste
- Оставили модель 10 на 10 миллионов для «похожести» Товара на Товар
- Хак: популярный Товар на сайте через похожесть
- Проблема холодного старта



Content-based рекомендации – реализация, риски

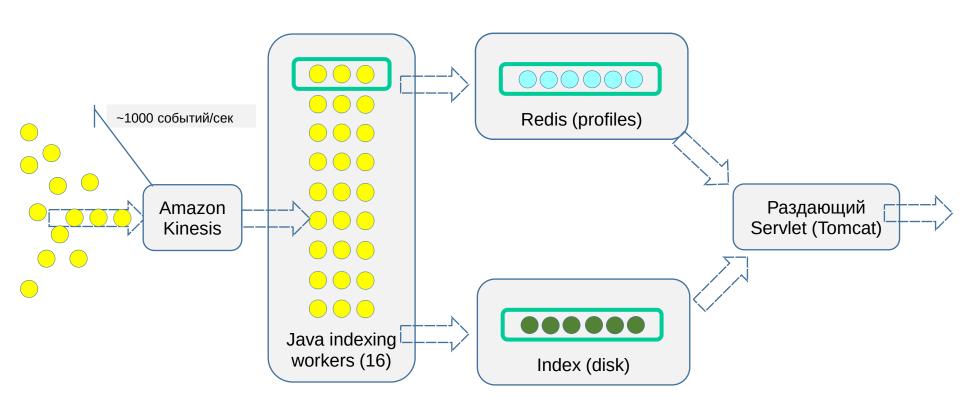
- Поисковый «движок»: Sphinx, Lucene (Solr)
- «Обвязка» для данных
- Хранение профиля Клиента
- Реализация: неделька. Риски объем данных, языки.

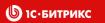
Cosine Similarity





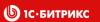
Content-based рекомендации – в Битрикс



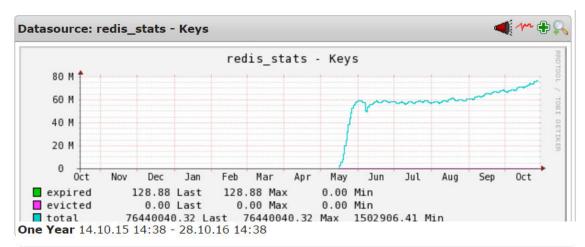


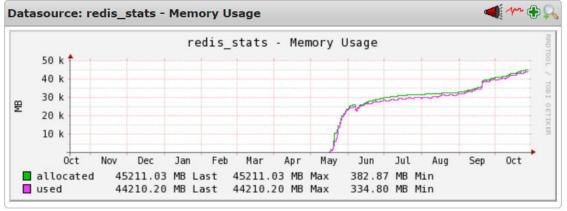
Content-based рекомендации – в Битрикс

- "Потребители": десятки тысяч интернет-магазинов
- "Поставщики": все сайты на Битрикс, больше 100к
- Тэги Профиля: название страницы, h1
- Индекс Товаров: название, краткое описание, разделы
- Индекс: гигабайты, сотни файлов в папке



Content-based рекомендации – в Битрикс





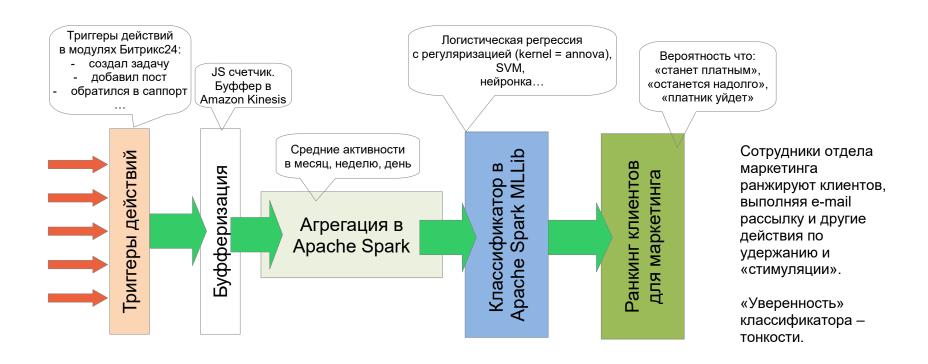


Классификация – компании-клиенты Битрикс24

- Кто из бесплатников станет платником?
- Кто из платников уйдет?
- Больше 2.5 миллиона зарегистрированных компаний
- Сбор счетчиков (десятки метрик) и агрегация в Apache Spark
- Классификация, Apache Spark MLlib, логистическая регрессия с регуляризацией
- Выгрузка моделей в админки для маркетинга, рассылки, конверсия на 2-3% выше
- Минус: небольшой охват с уверенным предсказанием



Классификатор клиентов Битрикс24





Спасибо за внимание! Вопросы?

Александр Сербул

@AlexSerbul

Alexandr Serbul serbul@1c-bitrix.ru

