

XII международная конференция  
CEE-SECR / РАЗРАБОТКА ПО

28 - 29 октября, Москва



Семантическое ядро рунета – высоконагруженная  
content-based рекомендательная система  
реального времени на базе Amazon Kinesis/Lucene

Александр Сербул  
ООО “1С-Битрикс”

# О чем поговорим?

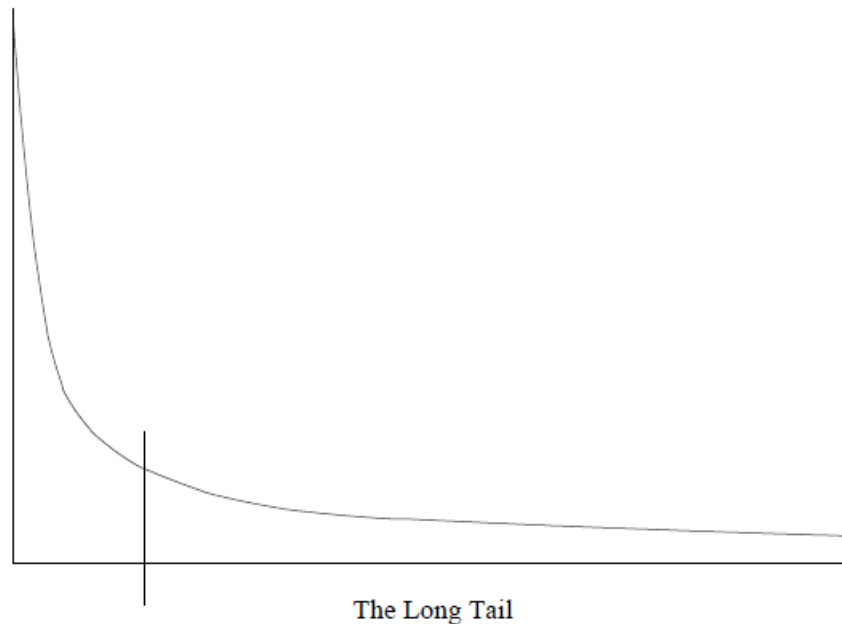
- Типы и алгоритмы рекомендательных систем – кратко
- Как мы собираем данные о действиях клиентов
- Архитектура нашей рекомендательной системы
- Статистика использования
- Советы как писать свои рекомендательные системы

# Типы и алгоритмы рекомендательных систем

- Релевантный контент – «угадываем мысли»
- Релевантный поиск
- Предлагаем то, что клиенту нужно как раз сейчас
- Увеличение лояльности, конверсии

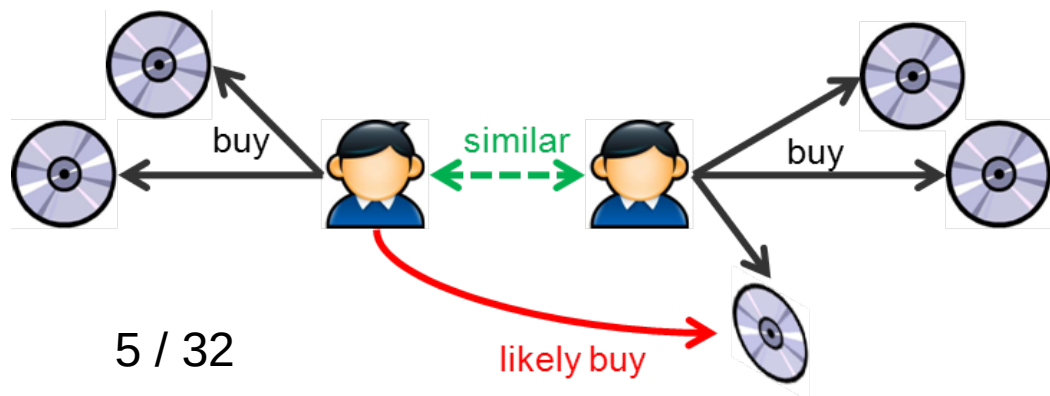
# Объем продаж товаров

- Best-sellers
- Топ-продаж...
- С этим товаром покупают
- Персональные рекомендации



# Коллаборативная фильтрация

- Предложи Товары/Услуги, которые есть у твоих друзей (User-User)
- Предложи к твоим Товарам другие связанные с ними Товары (Item-Item): «сухарики к пиву»



	$M_1$	$M_2$	$M_3$	$M_4$	$M_5$	$M_6$
$U_1$	✓	✓	✓	✓		
$U_2$	✓		✓	✓	✓	
$U_3$		✓				✓

# Возможности коллаборативной фильтрации

- Персональная рекомендация (рекомендуем посмотреть эти Товары)
- С этим Товаром покупают/смотрят/... (глобальная)
- Топ Товаров на сайте

Apache Spark MLlib (als), Apache Mahout (Taste) + неделька

Объем данных

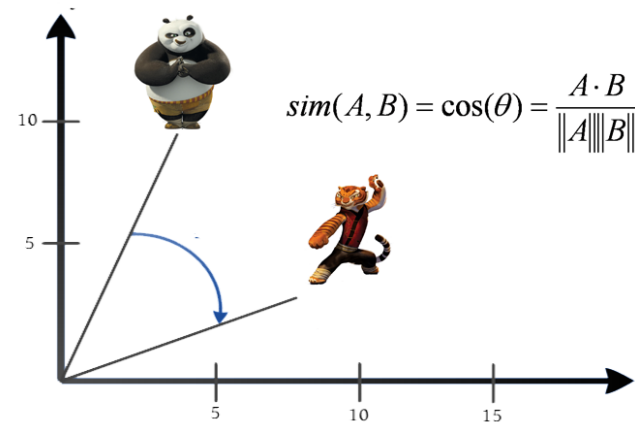
Объем модели, требования к «железу»

# Content-based рекомендации

- Купил пластиковые окна – теперь их предлагают на всех сайтах и смартфоне.
- Купил Toyota, ищу шины, предлагают шины к Toyota
- Поисковый «движок»: Sphinx, Lucene (Solr)
- «Обвязка» для данных
- Хранение профиля Клиента
- Реализация: неделька. Риски – объем данных, языки.

7 / 32

## Cosine Similarity



# Тюнинг рекомендательных систем

- Рекомендовать постоянно «возобновляемые» Товары (молоко, носки, ...)
- Рекомендовать фильм/телевизор – один раз до покупки
- Учет пола, возраста, размера, ...



8 / 32



**2016**  
**CEE-SEC(R)**



# Как собирать данные от клиентов?

- Как собирать?
- Куда собирать?
- Как обрабатывать?
- Поточковые алгоритмы...

# Технологии - RabbitMQ

<http://www.rabbitmq.com>

1. Очереди сообщений

на все вкусы

2. AMQP

3. Erlang

## 1 "Hello World!"

The simplest thing that does something



- > Python
- > Java
- > Ruby
- > PHP
- > C#
- > Javascript
- > Go
- > Elixir
- > Objective-C

## 2 Work queues

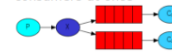
Distributing tasks among workers



- > Python
- > Java
- > Ruby
- > PHP
- > C#
- > Javascript
- > Go
- > Elixir
- > Objective-C

## 3 Publish/Subscribe

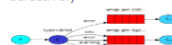
Sending messages to many consumers at once



- > Python
- > Java
- > Ruby
- > PHP
- > C#
- > Javascript
- > Go
- > Elixir
- > Objective-C

## 4 Routing

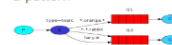
Receiving messages selectively



- > Python
- > Java
- > Ruby
- > PHP
- > C#
- > Javascript
- > Go
- > Elixir
- > Objective-C

## 5 Topics

Receiving messages based on a pattern



- > Python
- > Java
- > Ruby
- > PHP
- > C#
- > Javascript
- > Go
- > Elixir
- > Objective-C

## 6 RPC

Remote procedure call implementation

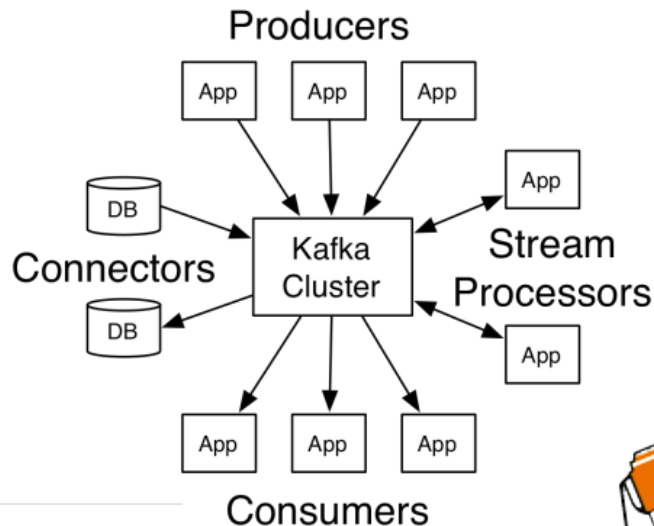
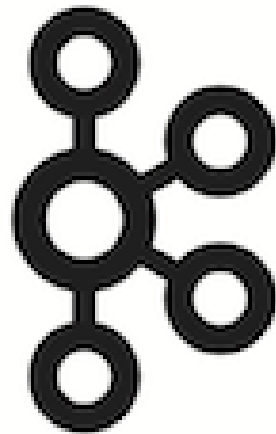


- > Python
- > Java
- > Ruby
- > PHP
- > C#
- > Javascript
- > Go
- > Elixir

# Технологии - Apache Kafka

<http://kafka.apache.org/>

1. "LinkedIn"
2. Не совсем очередь
3. Совсем не очередь!
4. Клиентское приложение «держит» курсор потока
5. Scala



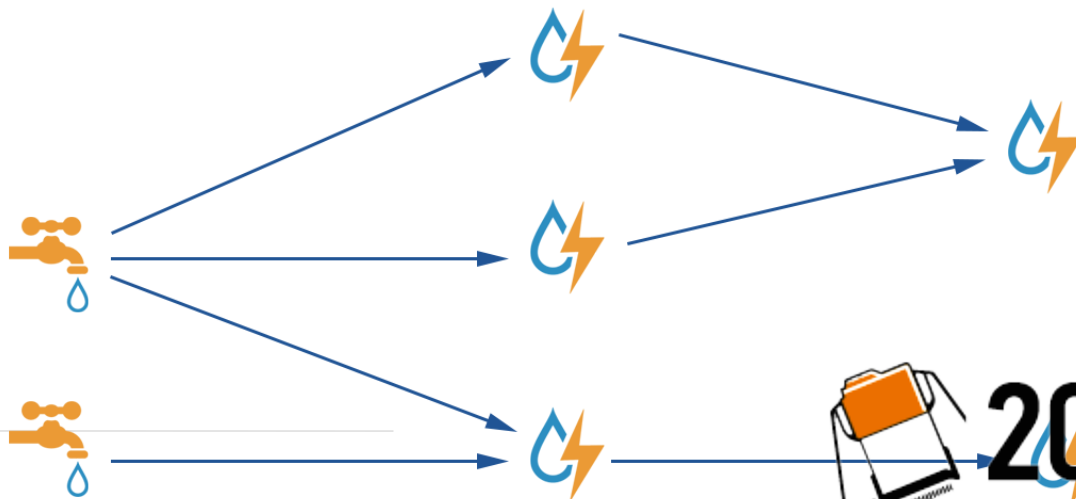
# Технологии - Apache Storm

<http://storm.apache.org>

1.Task parallel

2.Удобные, гибкие  
workflow

3.Clojure/JVM



# Технологии - Pinba

<http://pinba.org>



**Pinba**

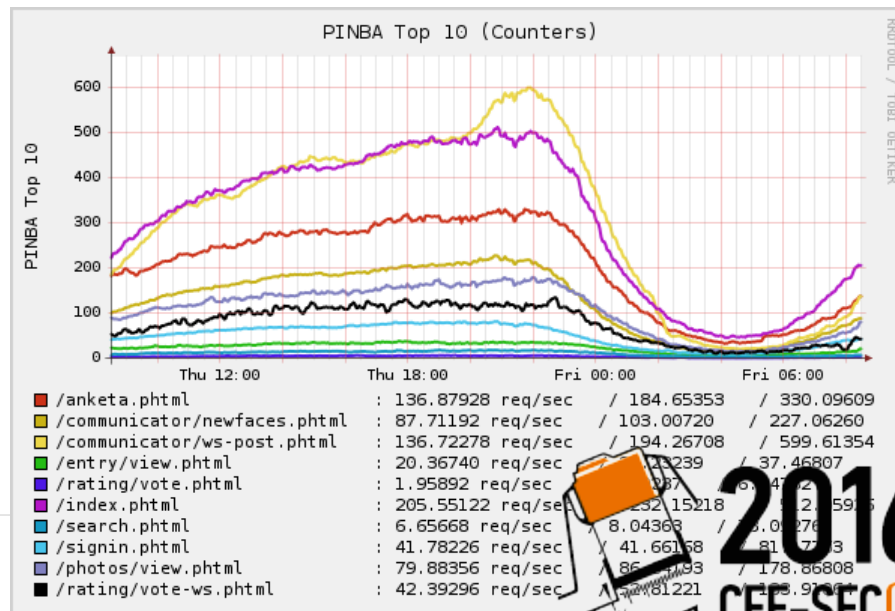
Агрегация внутри собств.

движка в MySQL

1. Интеграция с PHP

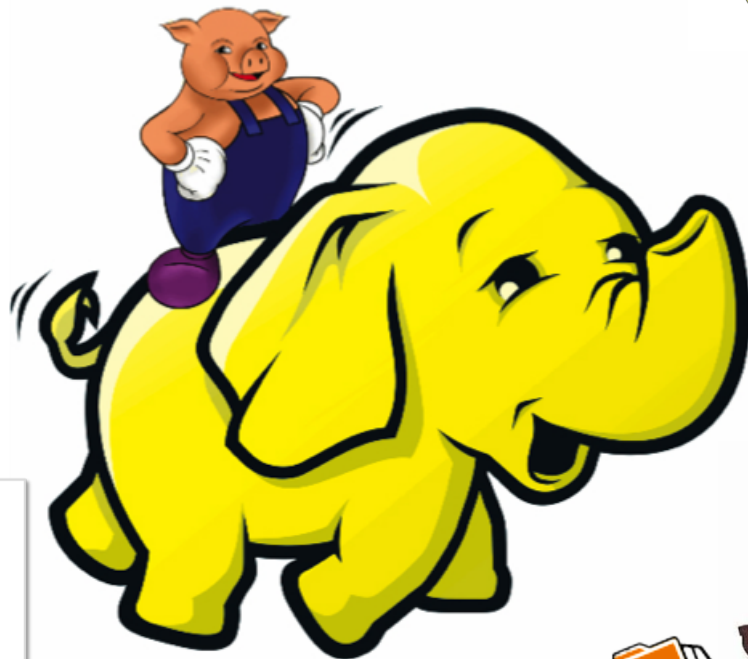
2. Быстро, удобно

3. Badoo.com



# Apache Hadoop

- Платформа:
  - - вычисления (MapReduce)
  - - файловая система (HDFS)
  - - “SQL-запросы” по данным



Sample Hadoop Applications

(Hive)



APACHE  
**HBASE**





More...

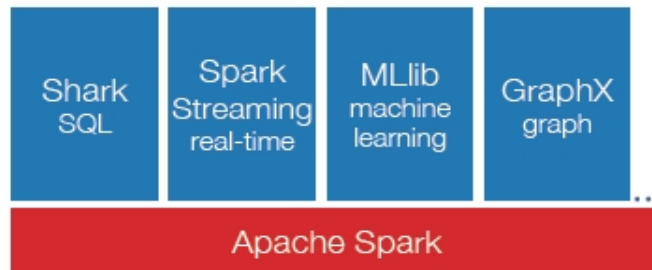


# Apache Spark

- Скорость!
- Работа в памяти
- Кэширование в памяти
- Простота развертывания



**Sophisticated:** can run today's most advanced algorithms



# Парадигма MapReduce

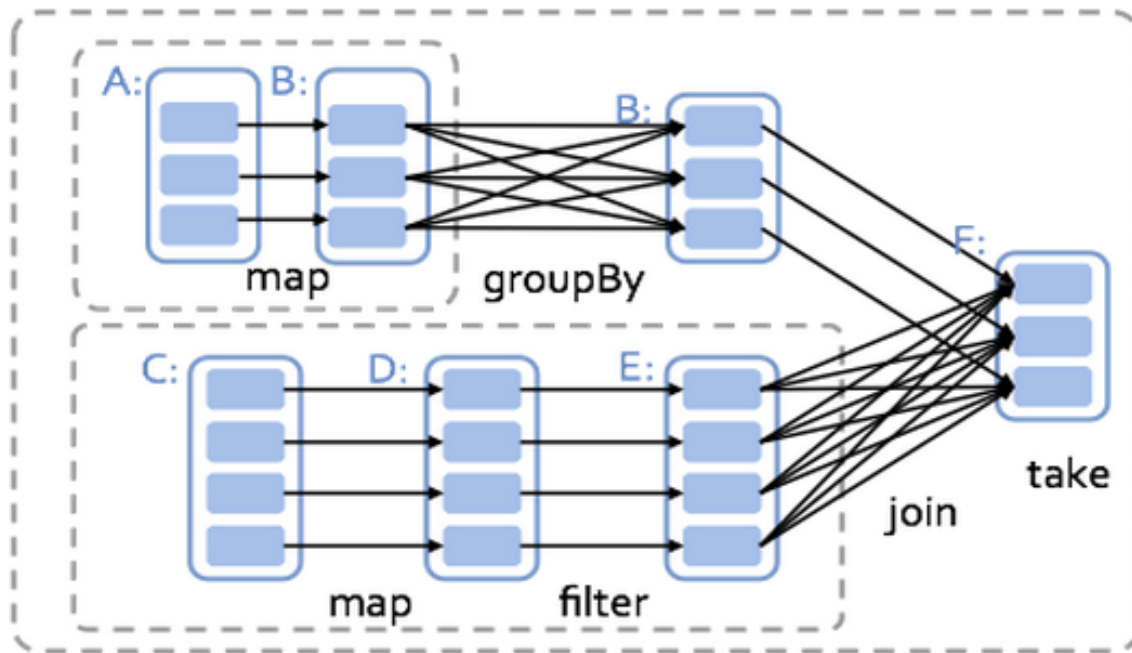
2.3	Algorithms Using MapReduce . . . . .	30
2.3.1	Matrix-Vector Multiplication by MapReduce . . . . .	31
2.3.2	If the Vector $v$ Cannot Fit in Main Memory . . . . .	32
2.3.3	Relational-Algebra Operations . . . . .	33
2.3.4	Computing Selections by MapReduce . . . . .	35
2.3.5	Computing Projections by MapReduce . . . . .	36
2.3.6	Union, Intersection, and Difference by MapReduce . . . . .	36
2.3.7	Computing Natural Join by MapReduce . . . . .	37
2.3.8	Grouping and Aggregation by MapReduce . . . . .	38
2.3.9	Matrix Multiplication . . . . .	38
2.3.10	Matrix Multiplication with One MapReduce Step . . . . .	39

«Mining of Massive Datasets»: Leskovec, Rajaraman, Ullman





# Apache Spark



# «Online» алгоритмы, они другие!

- Кластеризация
- Уникальные элементы
- Агрегация
- Ограничения по памяти
- Это – уже не SQL ;-)

18 / 32

4	Mining Data Streams	131
4.1	The Stream Data Model	131
4.1.1	A Data-Stream-Management System	132
4.1.2	Examples of Stream Sources	133
4.1.3	Stream Queries	134
4.1.4	Issues in Stream Processing	135
4.2	Sampling Data in a Stream	136
4.2.1	A Motivating Example	136
4.2.2	Obtaining a Representative Sample	137
4.2.3	The General Sampling Problem	137
4.2.4	Varying the Sample Size	138
4.2.5	Exercises for Section 4.2	138
4.3	Filtering Streams	139
4.3.1	A Motivating Example	139
4.3.2	The Bloom Filter	140
4.3.3	Analysis of Bloom Filtering	140
4.3.4	Exercises for Section 4.3	141
4.4	Counting Distinct Elements in a Stream	142
4.4.1	The Count-Distinct Problem	142
4.4.2	The Flajolet-Martin Algorithm	143
4.4.3	Combining Estimates	144
4.4.4	Space Requirements	144
4.4.5	Exercises for Section 4.4	145
4.5	Estimating Moments	145
4.5.1	Definition of Moments	145
4.5.2	The Alon-Matias-Szegedy Algorithm for Second Moments	146
4.5.3	Why the Alon-Matias-Szegedy Algorithm Works	147
4.5.4	Higher-Order Moments	148
4.5.5	Dealing With Infinite Streams	150
4.5.6	Exercises for Section 4.5	150
4.6	Counting Ones in a Window	150
4.6.1	The Cost of Exact Counts	151
4.6.2	The Datar-Gionis-Indyk-Motwani Algorithm	151
4.6.3	Storage Requirements for the DGIM Algorithm	151



2016  
CEE-SEC(R)

# Война систем хранения

- SQL на MapReduce: Hive, Pig, Spark SQL
- SQL на MPP (massive parallel processing):

Impala, Presto, Amazon RedShift, Vertica

- NoSQL: Cassandra, Hbase, Amazon DynamoDB
- Классика: MySQL, MS SQL, Oracle, ...

## Not All SQL on Hadoop is Created Equal

### Batch MapReduce

Make MapReduce faster



Slow, still batch

### Remote Query

Pull data from HDFS over the network to the DW compute layer



Slow, expensive

### Siloed DBMS

Load data into a proprietary database file



Rigid, siloed data, slow ETL

### Impala

Native MPP query engine that's integrated into Hadoop

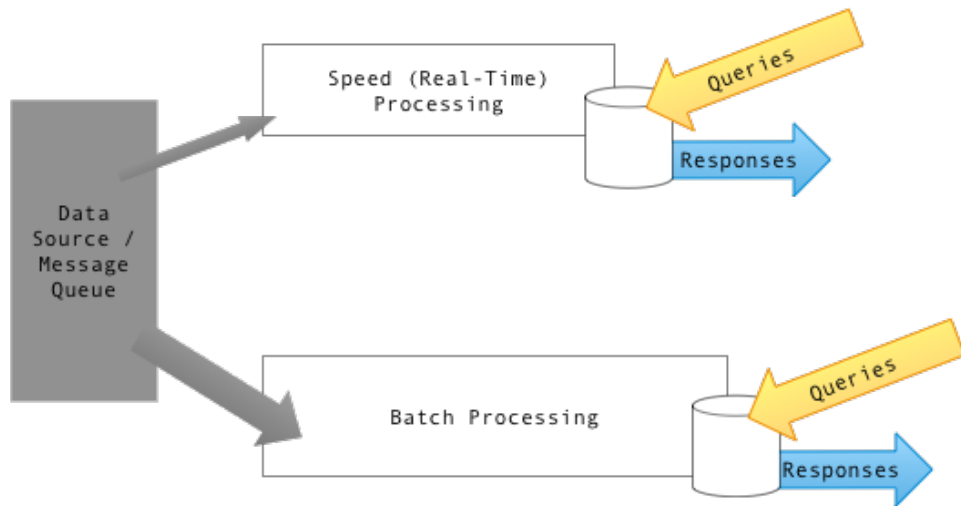


Fast, flexible, cost-effective



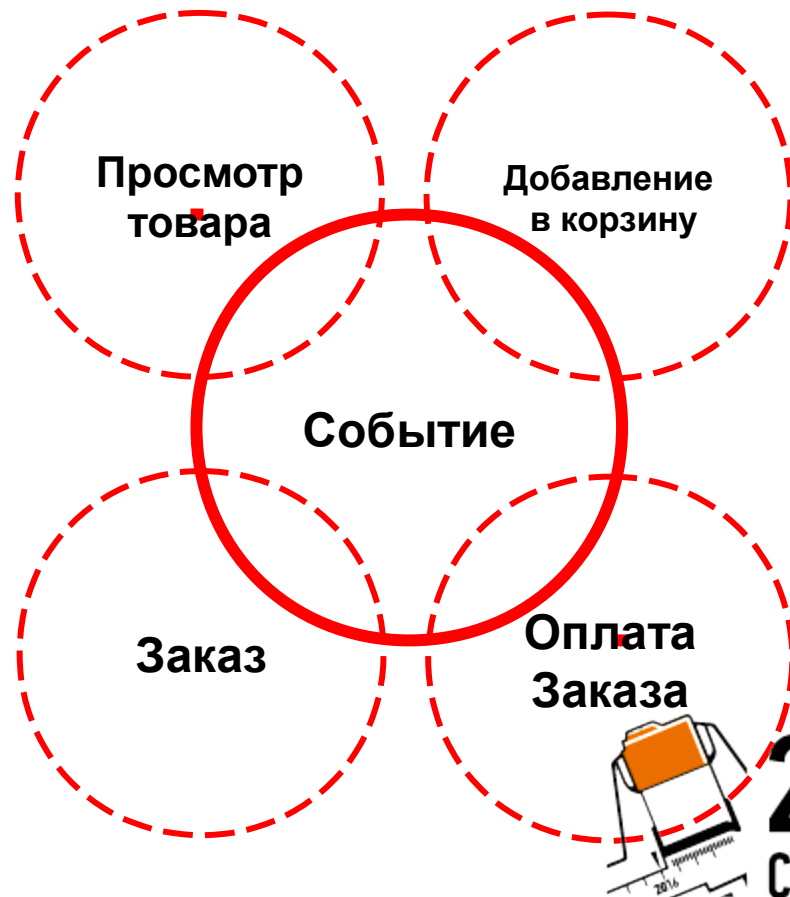
# Lambda-архитектура

- Людей заваливает данными...
- С большой скоростью 😊

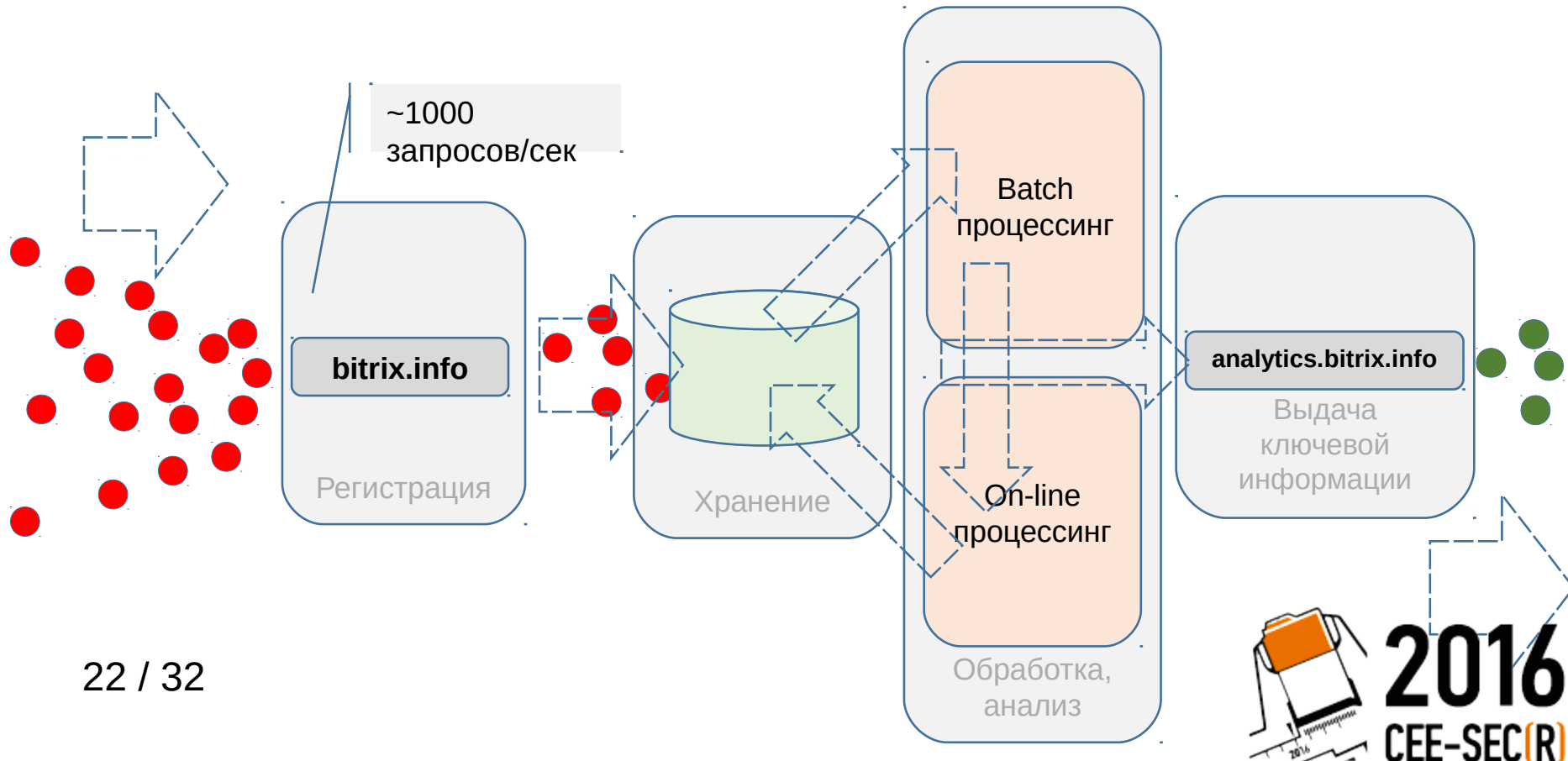


# BigData – «под капотом». Виды событий.

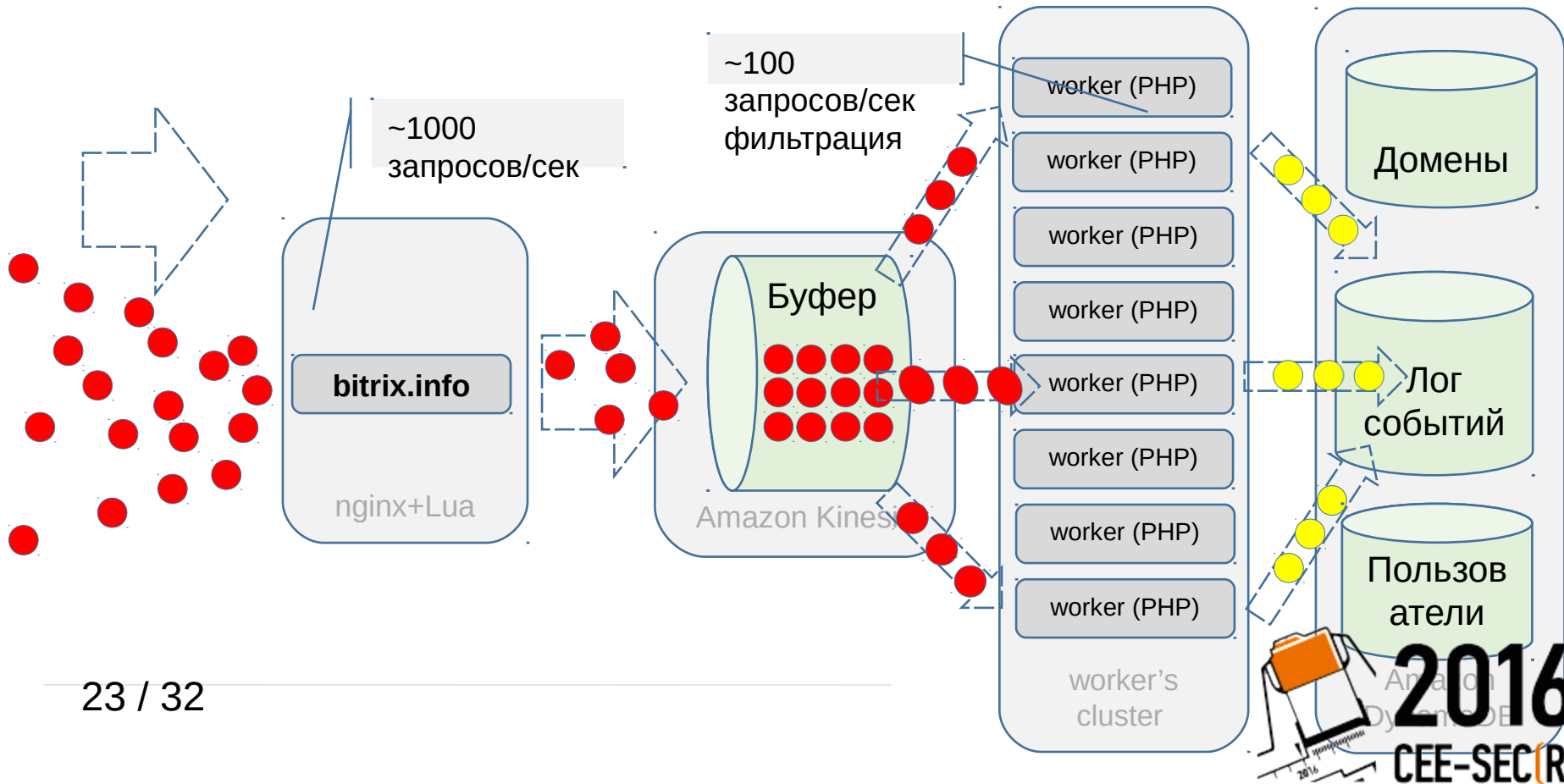
- Кука Пользователя
- Хэш лицензии
- Домен
- ID товара
- Название Товара
- Категории Товара
- ID рекомендации
- ряд других



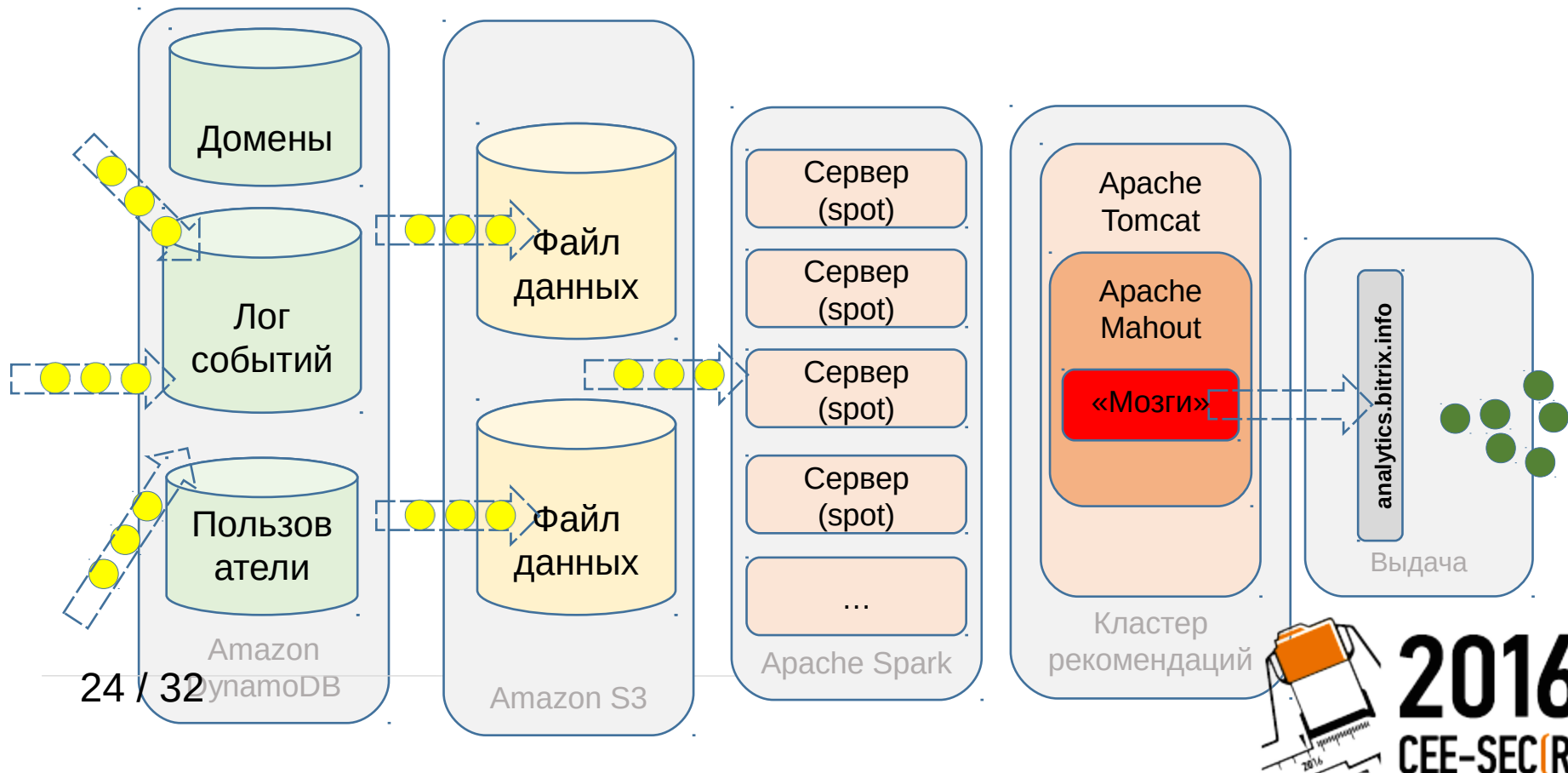
# BigData – «под капотом». С птичьего полета.



# BigData – «под капотом». Регистрация событий.



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





# Apache Lucene



- Doug Cutting: Nutch, Hadoop (Yahoo!)... сейчас в Cloudera
- Lucene: Solr, ElasticSearch
- Lucene: грамотная многопоточность из коробки



elastic



# Apache Lucene: +/-



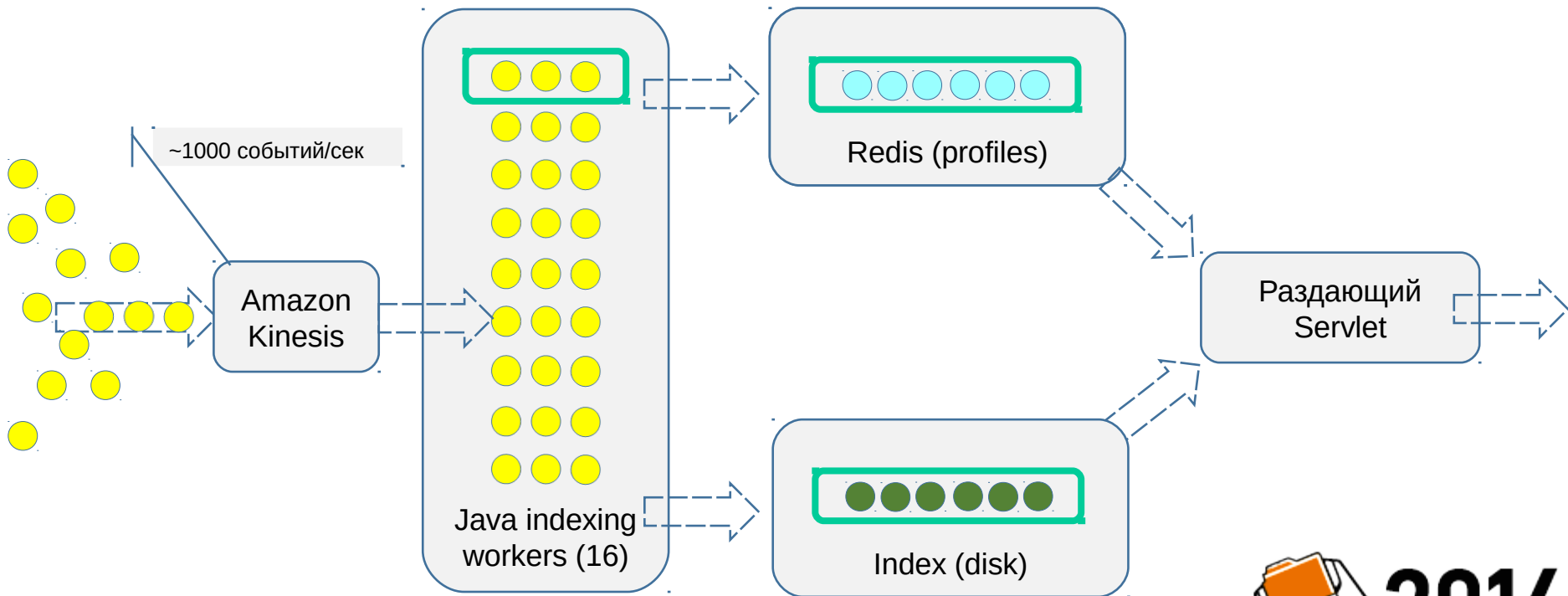
- (-) Нет нормальной поддержки русского языка
- (-) Нет русской морфологии
- (-) Документация иногда оставляет желать лучшего
- (-) Нет решения для 100% онлайн индексации
  
- (+) Компактный индекс (гигабайты)
- (+) Лаконичное API
- (+) Транзакционность
- (+) Thread-safety

# Redis



- Профиль Пользователя: десятки тэгов
- Стемминг Портера
- Высокочастотные слова
- Алгоритмы вытеснения тэгов
- Куда можно развивать... (word2vec, glove, синонимы ...)

# Архитектура content-based рекомендаций



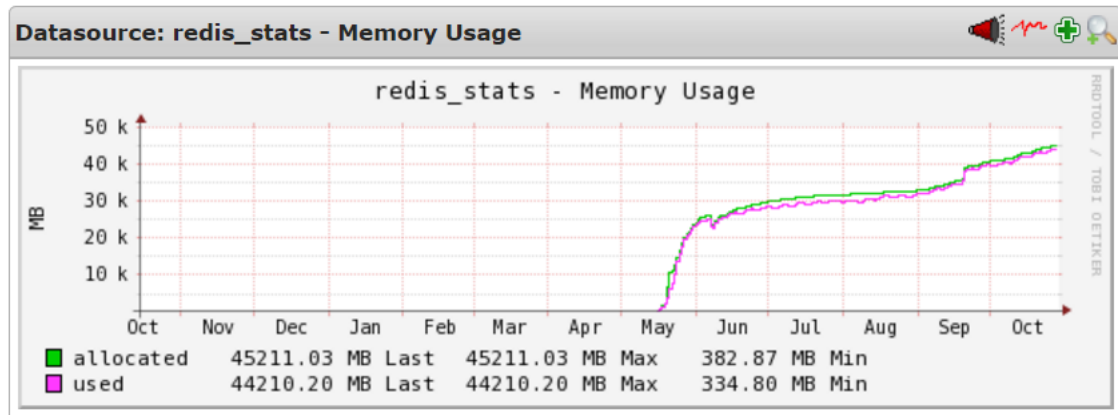
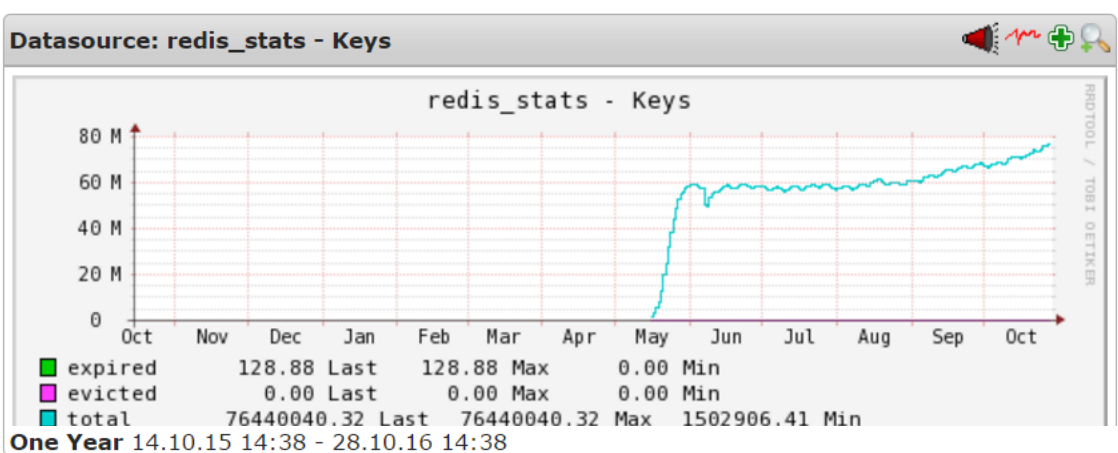
# Процессы

- Многопоточный индексатор, java/lucene
- Amazon Kinesis – как буфер
- Индекс в папке на диске, вытеснение
- Как реализован “онлайн”
- Раздающий Servlet

# Цифры и данные

- “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

# Цифры и данные



Спасибо за внимание!  
Вопросы?

**Александр Сербул**

 @AlexSerbul

serbul@1c-bitrix.ru



**2016**  
CEE-SEC(R)