How we solved **Real-Time User Segmentation** using **HBase** ?

Murtaza Doctor Principal Architect
Giang Nguyen Sr. Software Engineer

Outline

- {rr} Story
- {rr} Personalization Platform
- User Segmentation: Problem Statement
- Design & Architecture
- Performance Metrics
- Q&A

richrelevance®

Dynamic Personalization for Commerce

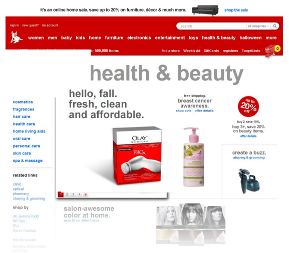
Multiple Personalization Placements

richrecs

People Who Viewed This Laptop Also Viewed
Lenovo ThinkPad X200 12.1"Notebook - Bla...
Toshiba Satellite L555-S7916 17.3 Inch N...
Lenovo ThinkPad W500 15.4" Notebook - Bl...
$1379.99
$699.99
VIEW MORE
VIEW MORE
kmart

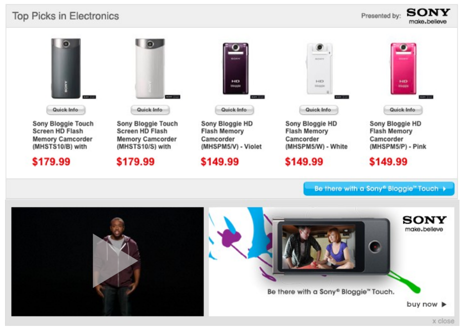
100+ algorithms dynamically targeted by page, context, and user behavior, across all retail channels.

richpromo


health & beauty
hello, fall. fresh, clean and affordable.
salon-awesome color at home.
create a buzz.

Targeted content optimization to customer segments

richads


Top Picks in Electronics
Presented by: SONY make.believe
Sony Bloggie Touch Screen HD Flash Memory Camcorder (MHSTS10/B) with
$179.99
Sony Bloggie Touch Screen HD Flash Memory Camcorder (MHSTS10/S) with
$179.99
Sony Bloggie HD Flash Memory Camcorder (MHSPM5/V) - Violet
$149.99
Sony Bloggie HD Flash Memory Camcorder (MHSPM5/W) - White
$149.99
Sony Bloggie HD Flash Memory Camcorder (MHSPM5/P) - Pink
$149.99
Be there with a Sony® Bloggie™ Touch
SONY make.believe
Be there with a Sony® Bloggie™ Touch.
buy now
x close

Monetization through relevant brand advertising

Re ng

enRICH™
personalization engine

hadoop

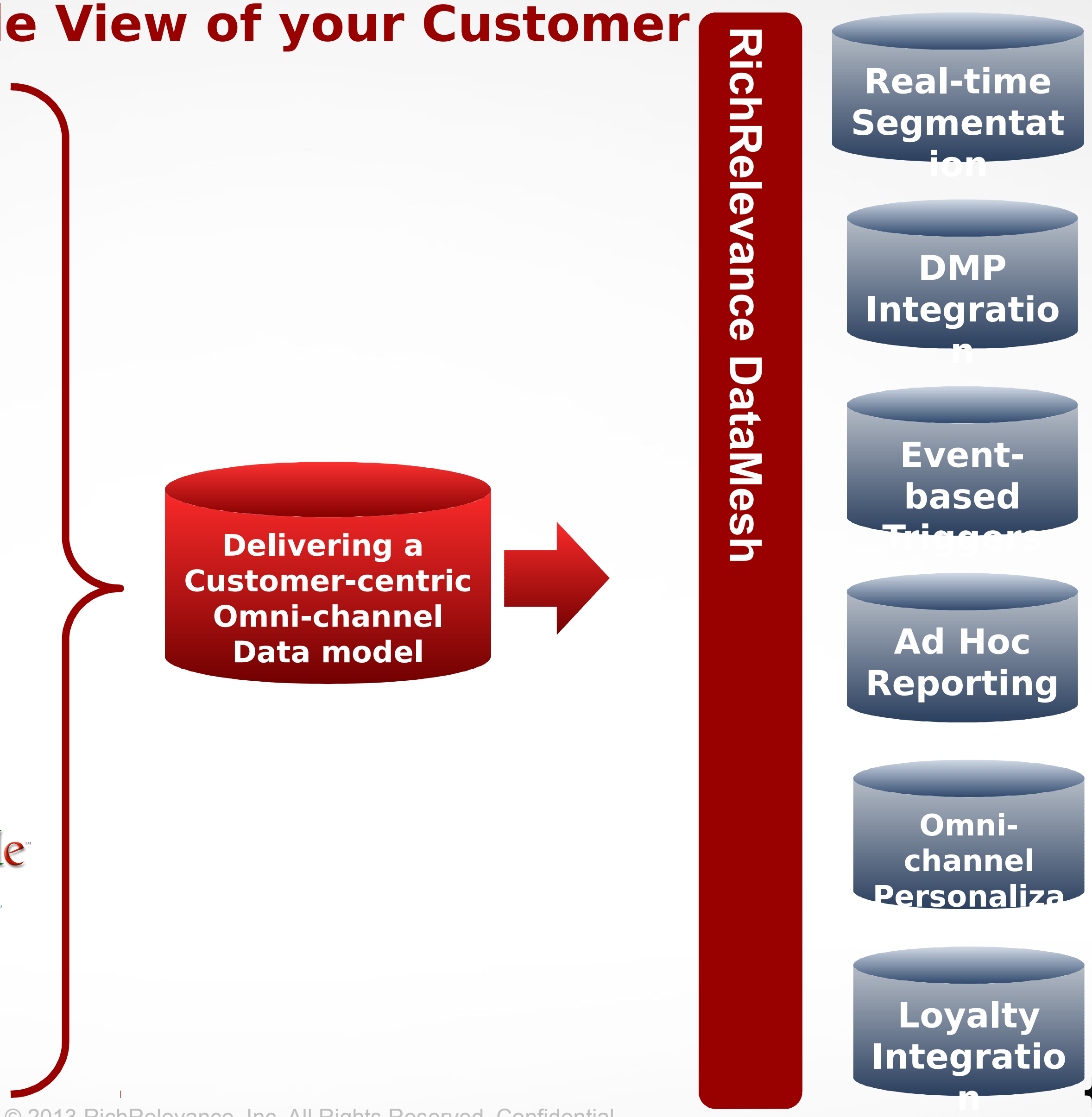
Input Data

- Real time user behavior
- Multi-Channel purchase history
- Inventory and Margin Data
- Catalog Attribute Data
- 3rd Party Data Sources
- Future Data Sources

RichRelevance DataMesh Cloud Platform

Delivering a Single View of your Customer

- ✓ **One place** where all data about your customer lives (e.g. loyalty, customer service, offline, catalogue)
- ✓ **Direct access** to data for your entire enterprise via API
- ✓ **Real-time** actionable data and business intelligence
- ✓ Link customer activity across **any channel**
- ✓ Leverage **3rd party data** (e.g. weather, geography, census)



Did You Know?

Our cloud-based platform supports both real-time processes and analytical use cases, utilizing technologies to name a few: **Crunch, Hive, HBase, Avro, Azkaban, Voldemort, Kafka**

In the US, we serve **7000 requests** per second with an average response time of **50 ms**

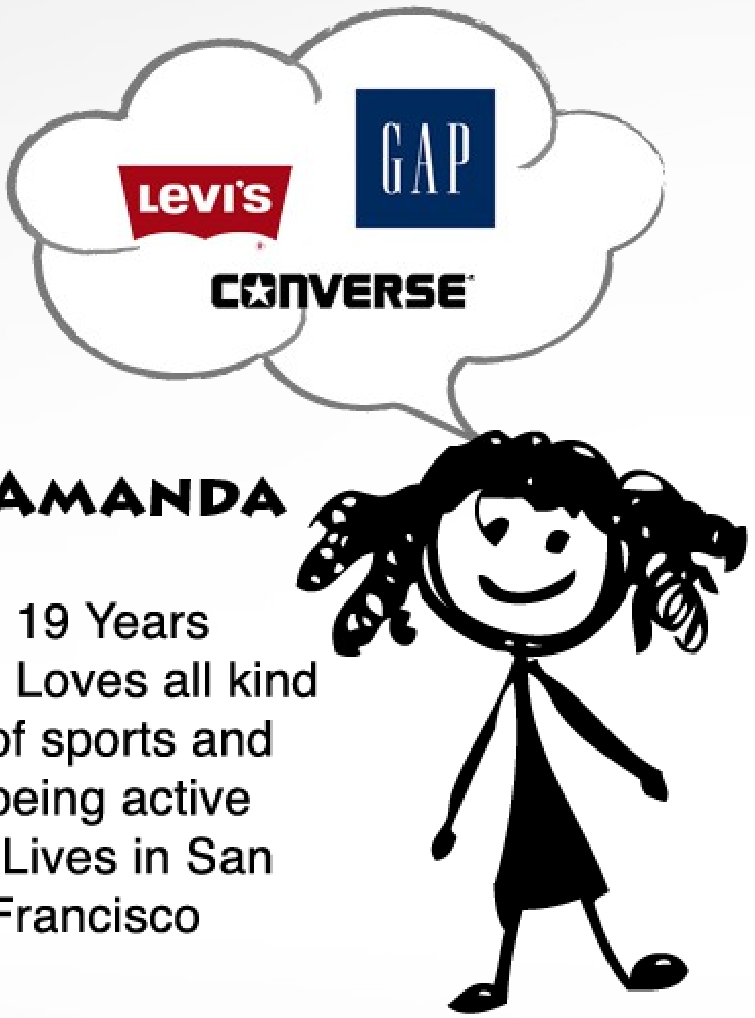
Our data capacity includes a **1.5 PB** Hadoop infrastructure, which enables us to employ **100+ algorithms** in real-time

Someone clicks on a {rr} recommendation every **21 milliseconds**

Real-Time User Segmentation

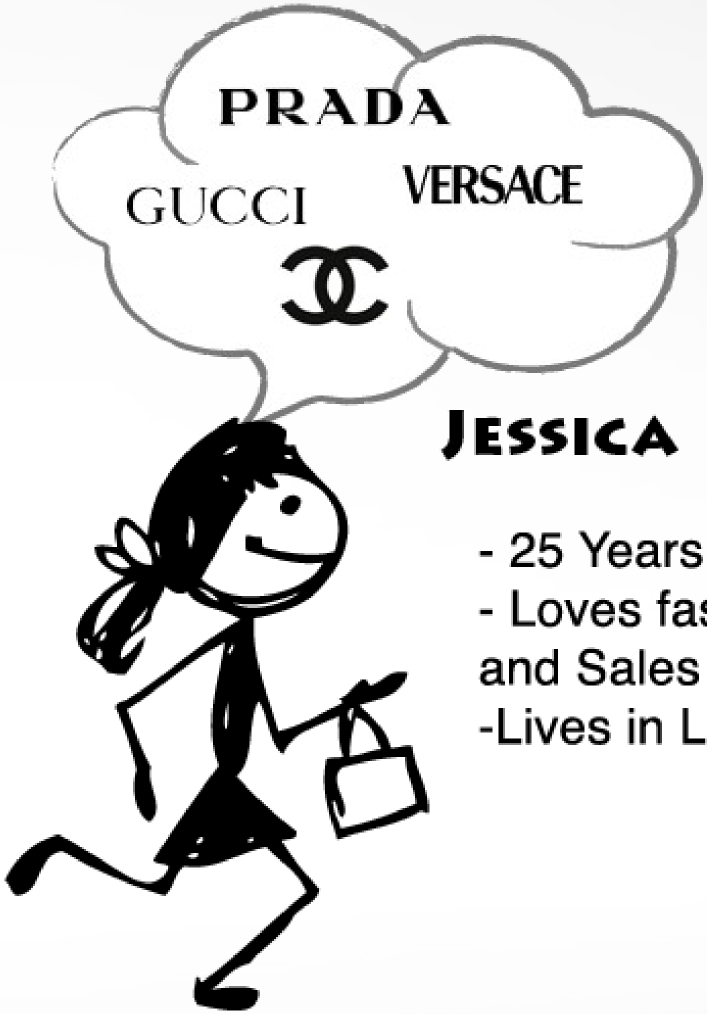
Finding and Targeting the Right Audience

Meet Amanda and Jessica



Amanda

- 19 Years
- Loves all kind of sports and being active
-Lives in San Francisco

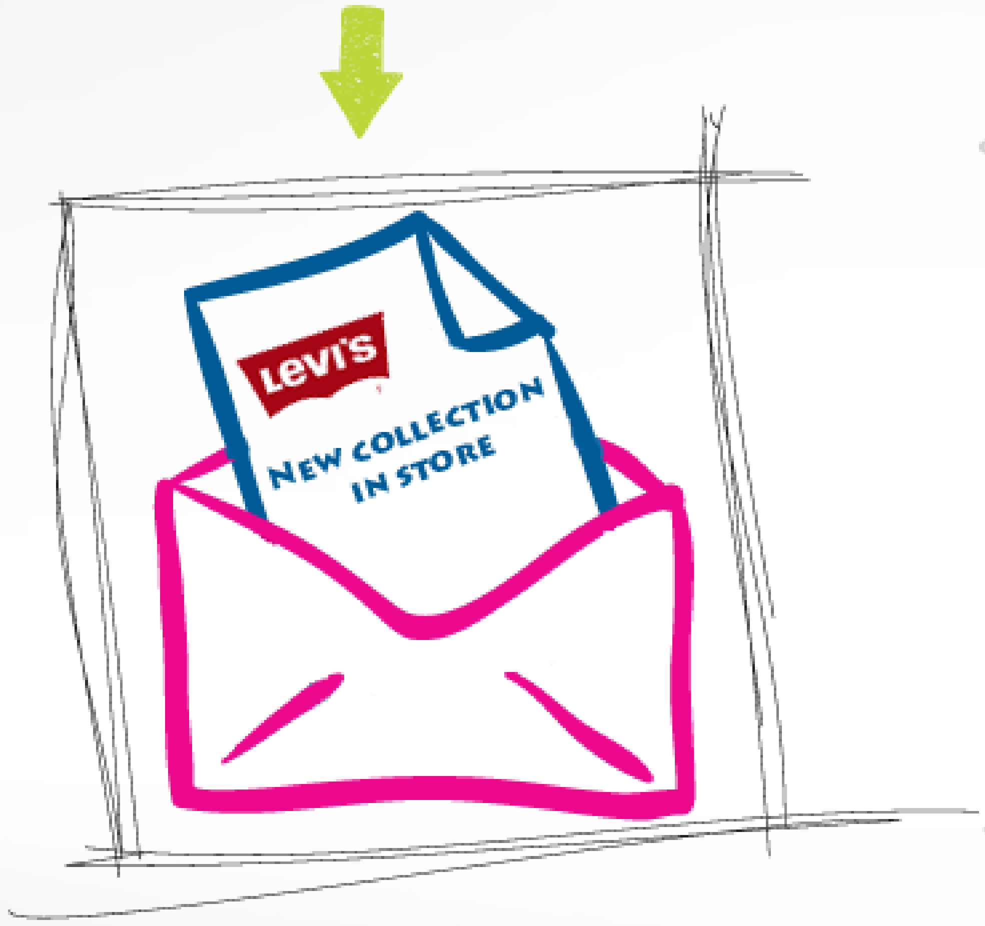


Jessica

- 25 Years
- Loves fashion and Sales
-Lives in L.A.

Because We Know What They Like!

For Amanda

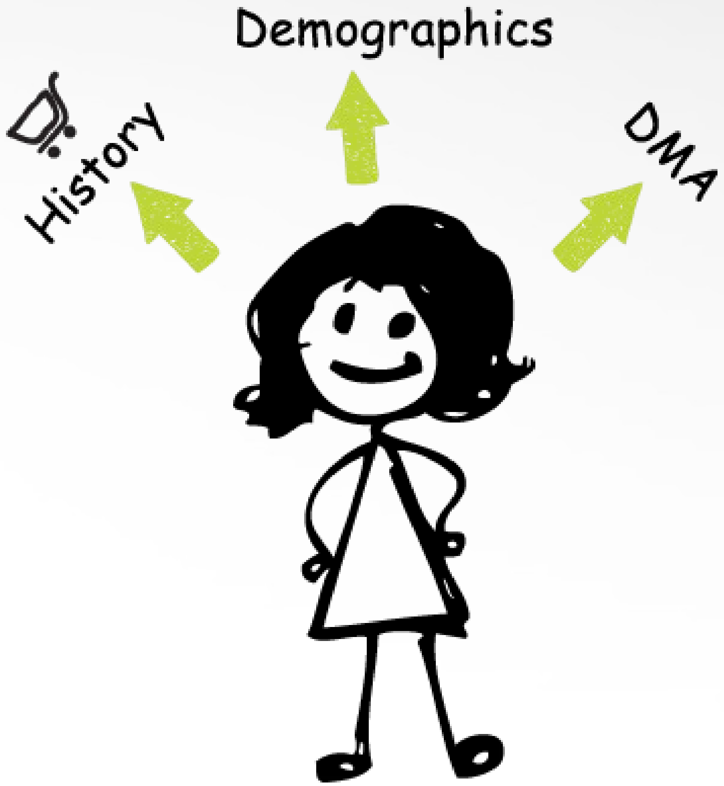


For Jessica

CHANEL
50% Sale

What is the {rr} Segment Builder?

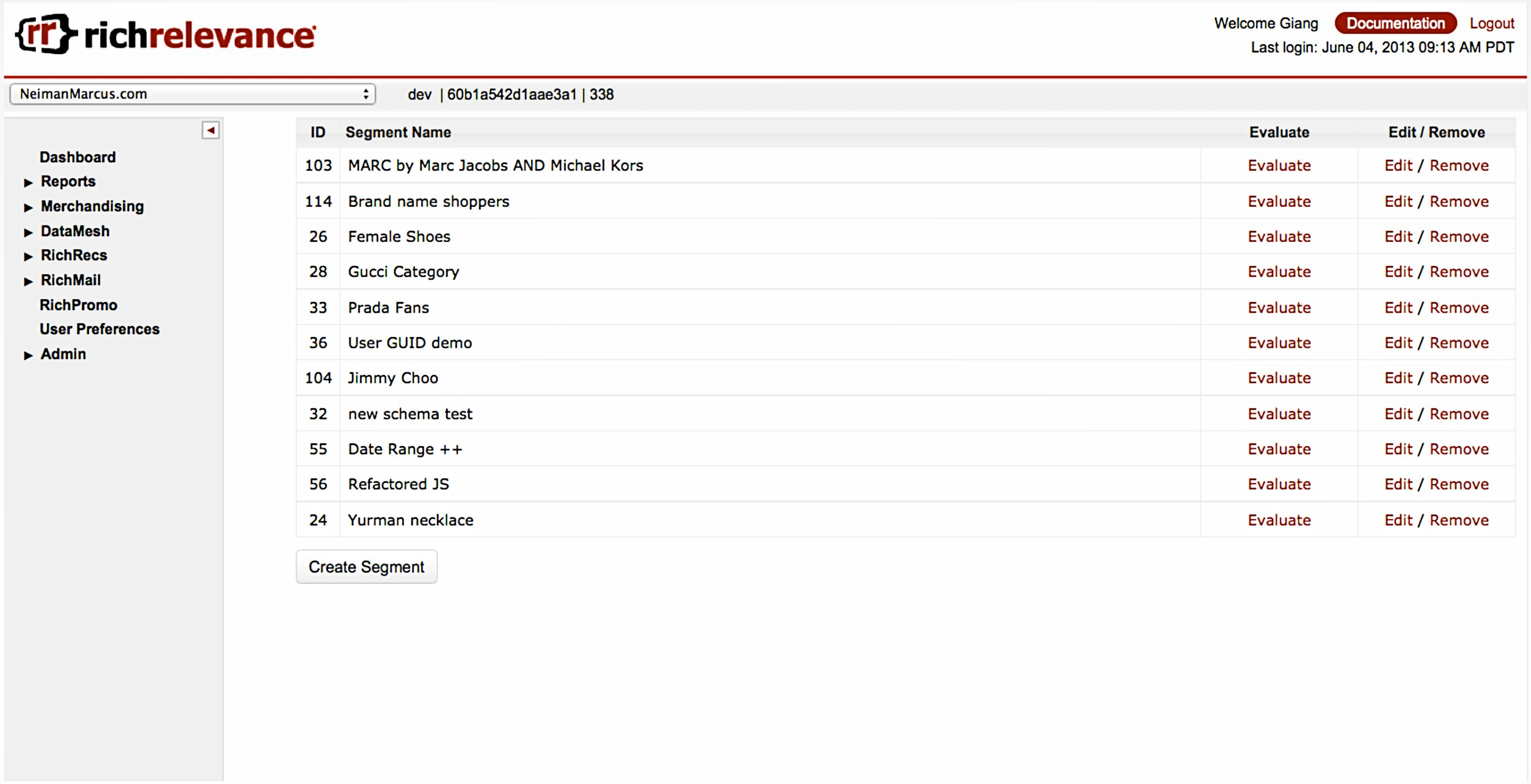
- Utilizes valuable targeting data such as **event logs**, **off-line data**, **DMA**, **demographics**, and etc.
- Finds highly qualified consumers and buckets them into segments



Example: for Retail, Segment Builder supports view, click, purchase on **products**, **categories**, and **brands**

Segment Builder

Segment's List page



Welcome Giang | Documentation | Logout

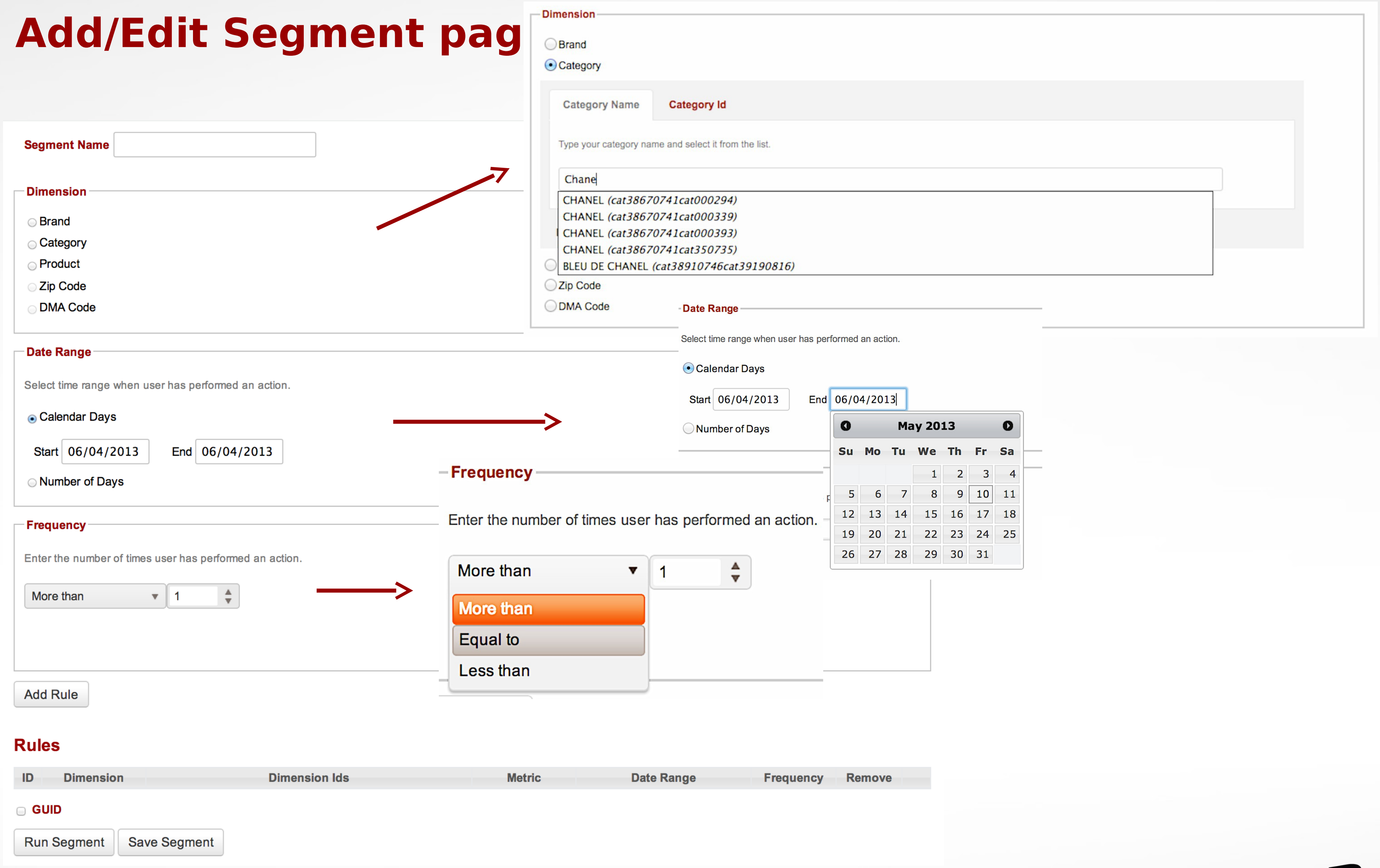
Last login: June 04, 2013 09:13 AM PDT

NeimanMarcus.com | dev | 60b1a542d1aae3a1 | 338

ID	Segment Name	Evaluate	Edit / Remove
103	MARC by Marc Jacobs AND Michael Kors	Evaluate	Edit / Remove
114	Brand name shoppers	Evaluate	Edit / Remove
26	Female Shoes	Evaluate	Edit / Remove
28	Gucci Category	Evaluate	Edit / Remove
33	Prada Fans	Evaluate	Edit / Remove
36	User GUID demo	Evaluate	Edit / Remove
104	Jimmy Choo	Evaluate	Edit / Remove
32	new schema test	Evaluate	Edit / Remove
55	Date Range ++	Evaluate	Edit / Remove
56	Refactored JS	Evaluate	Edit / Remove
24	Yurman necklace	Evaluate	Edit / Remove

Segment Builder

Add/Edit Segment pag



Segment Name

Dimension

- ○ Brand
- ○ Category
- ○ Product
- ○ Zip Code
- ○ DMA Code

Date Range

Select time range when user has performed an action.

- ● Calendar Days

Start 06/04/2013 End 06/04/2013

- ○ Number of Days

Frequency

Enter the number of times user has performed an action.

More than ▾ 1

Add Rule

Rules

ID	Dimension	Dimension Ids	Metric	Date Range	Frequency	Remove

☐ **GUID**

Run Segment | Save Segment

Dimension

- ○ Brand
- ● Category

Category Name | **Category Id**

Type your category name and select it from the list.

Chane

- CHANEL *(cat38670741cat000294)*
- CHANEL *(cat38670741cat000339)*
- CHANEL *(cat38670741cat000393)*
- CHANEL *(cat38670741cat350735)*
- BLEU DE CHANEL *(cat38910746cat39190816)*

- ○ Zip Code
- ○ DMA Code

Date Range

Select time range when user has performed an action.

- ● Calendar Days

Start 06/04/2013 End 06/04/2013

- ○ Number of Days

May 2013

Su	Mo	Tu	We	Th	Fr	Sa
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

Frequency

Enter the number of times user has performed an action.

More than ▾ 1

- More than
- Equal to
- Less than

Design: Segment Evaluation Engine

- Create segments to capture the audience via UI
- Each behavior is captured by a **rule**
- Each rule corresponds to a **row key in HBase**
- Each rule returns the **users**
- Rules are joined using set **union or intersection**
- **Segment** definition consists of one or more rules

Real-Time Data Ingestion

Our choice Avro and Kafka

Real-time data ingestion

- User interacts with a retail site
- Events are triggered in real-time, each event is converted into an **Avro** record
- Events are ingested using our real-time framework built using **Apache Kafka**

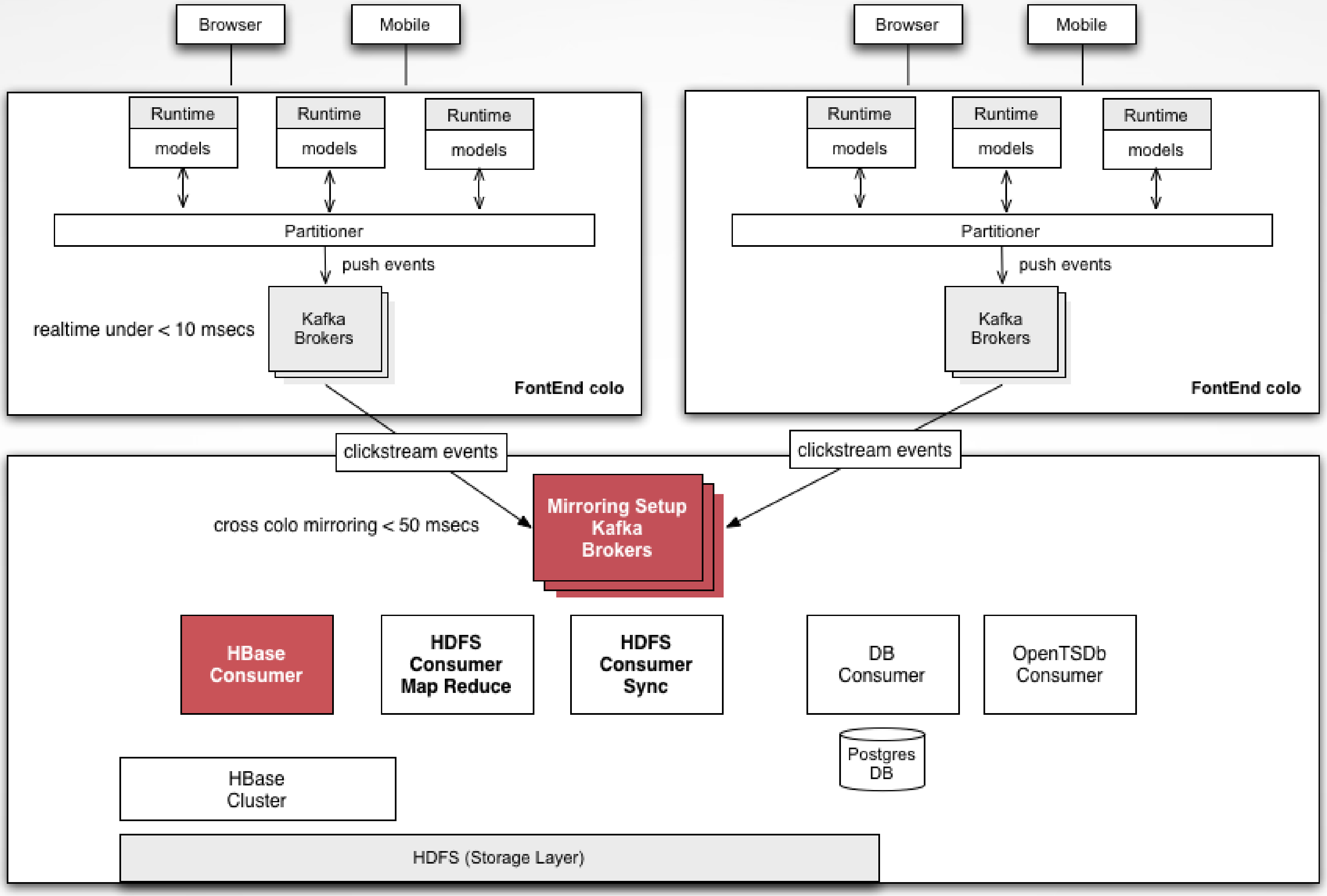
Design Principles: Real-Time Solution

- **Streaming**: Support for streaming versus batch
- **Reliability**: no loss of events and at least once delivery of messages
- **Scalable**: add more brokers, add more consumers, partition data
- **Distributed** system with central coordinator like zookeeper
- **Persistence** of messages
- Push & pull mechanism
- Support for **compression**
- Low operational complexity & easy to monitor

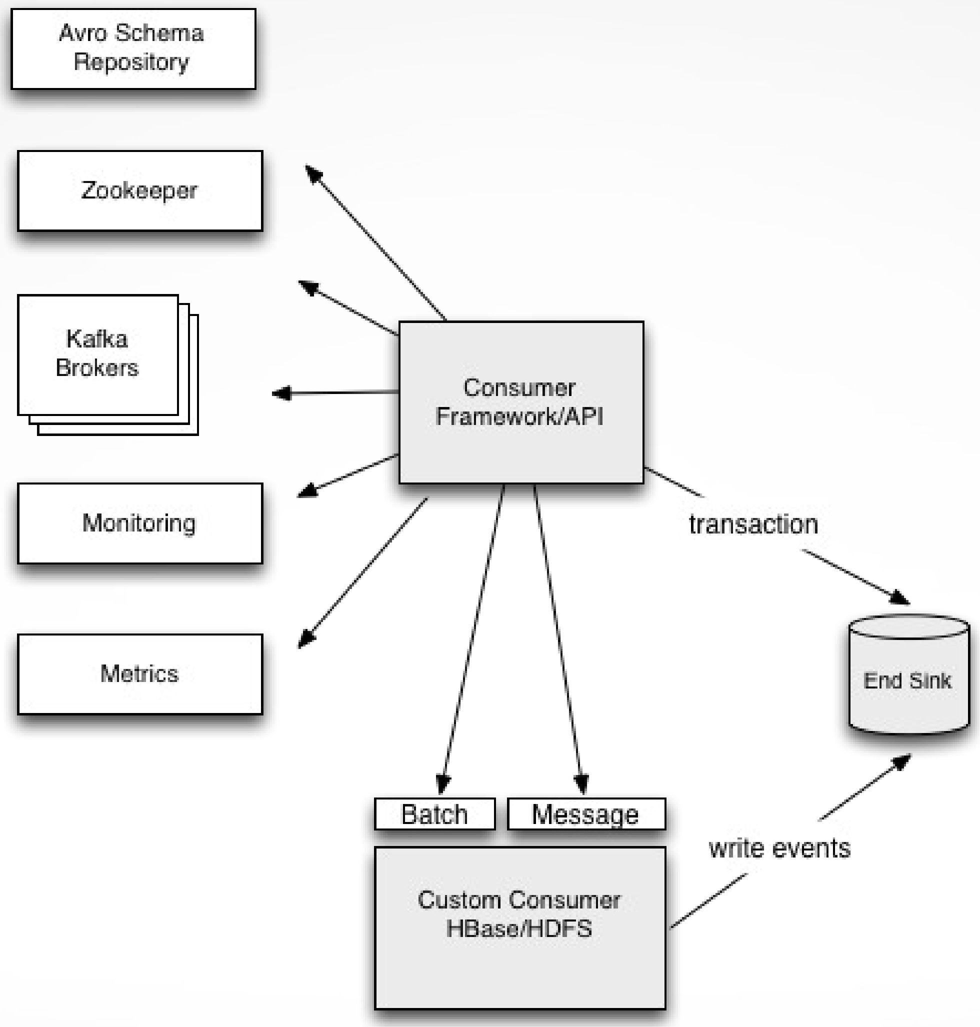
Our Decision – Apache Kafka

- **Distributed** pub-sub messaging system
- More **generic** concept (Ingest & Publish)
- Can support both **offline & online** use-cases
- Designed for **persistent messages** as the common case
- **Guarantee** ordering of events
- Supports **gzip & snappy** (0.8) compression protocols
- Supports **rewind offset & re-consumption** of data
- No concept of master node, **brokers are peers**

Kafka Architecture



Common Consumer Framework



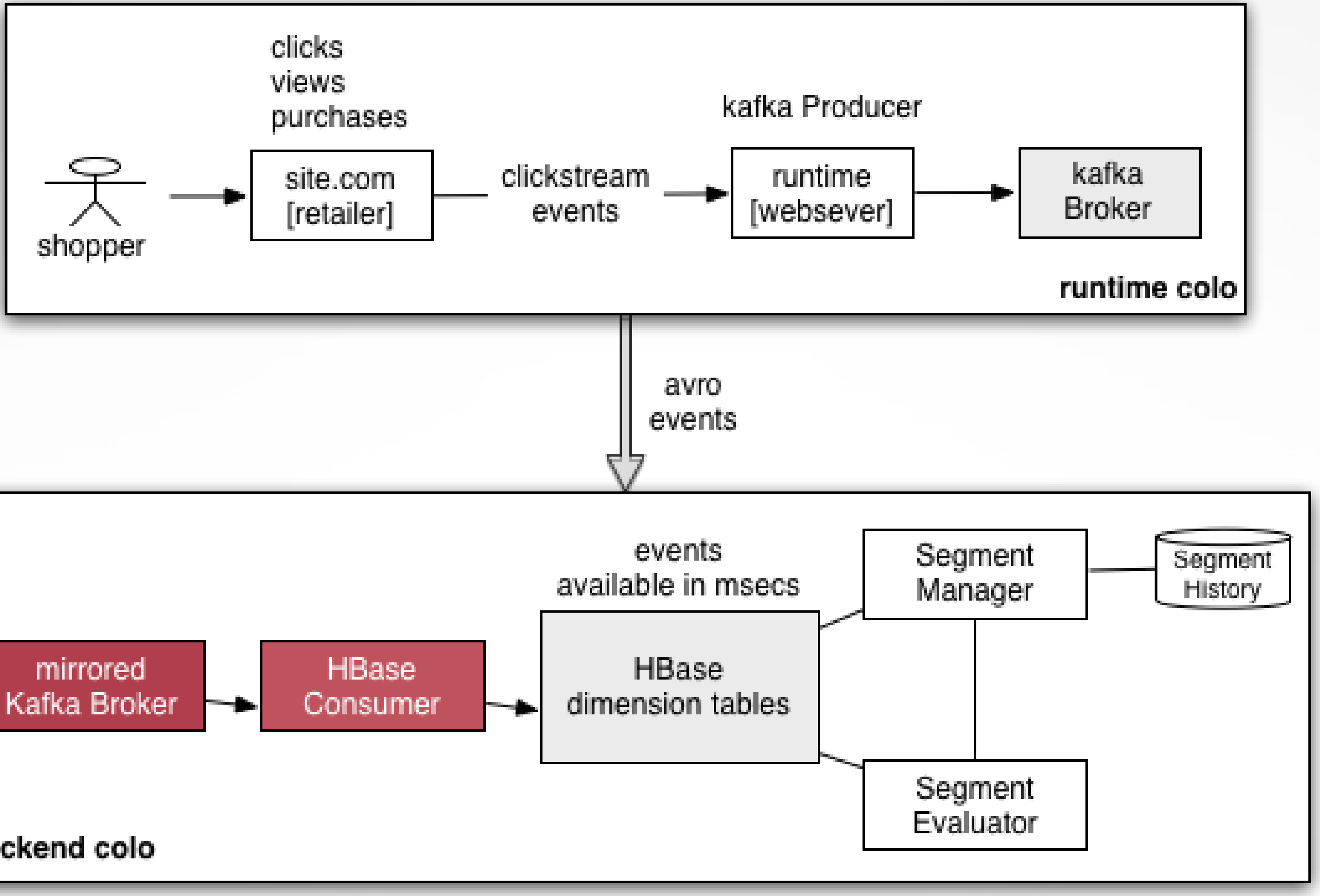
Volume Facts?

Daily clickstream event data	150 GB
Average size of message	2 KB
Batch Size	5000 messages
Producer throughput	5000 messages/sec
Real time HBase Consumer	7000 messages/sec

End to End real-time story…

- User exhibits a behavior
- Events are generated at front-end data-center
- Events are streamed to backend data center via **Kafka**
- Events are consumed and indexed in **HBase tables**
- Takes seconds from event origination to indexing in **HBase**

End to End real-time story...



User Segmentation Engines

Features	{rr} Engine	Other Engines
User's behavior ingestion	Real-time	Not Real-time
Batch style processing is done	Immediately	At end of a day
When segment membership is changed notifications will be	Event driven	N/A
Technologies used	Scalable and open source	Unscalable and proprietary

Use Cases

Use Case #1	Use Case #2
Users exhibit behaviors	Users exhibit behaviors
Behaviors ingested and indexed in real time	Segment membership calculated in real time
Users are now in corresponding segments	Notifications are sent on segment membership change
Retrieving users takes seconds	

Our choice HBase?

- Real time segment evaluation
- Optimized for read and scan
- Column cell supports frequency use case
- Eventual consistency does not work
- Seamless integration with Hadoop
- Possible with good row key design

HBase Row Key Design

- Took a few attempts
- Design considerations
  - Timestamp in row or columns
  - Partition behavior by date
  - Optimized for read or write
  - Hot spotting issues
  - Uniform key distribution

Design: First Attempt

- Row key represents behavior
- Columns store the user id
- Cell stores behavior time and capture frequency
- One column family U

RowKey	Columns
338VBChanel	23b93laddf82:1370377143973 Hd92jslahd0a:1313323414192
338CCElectronic	z3be3la2dfa2:1370477142970 kd9zjsla3d01:1313323414192

Design: First Attempt Issues

- Row too wide
- May exceed HFile size
- Terrible write/read performance

Design: Second Attempt

- Partition behavior by date
- Reduce row size
- Gained ability to scan across dates

Rowkey	Columns
338VBChanel1370377143	23b93laddf82:1370377143973 Hd92jslahd0a:1313323414192
338CCElectronic1370377143	z3be3la2dfa2:1370477142970 kd9zjsla3d01:1313323414192

Design: Second Attempt Issues

- Hot spotting
- Popular products or high level categories can have millions of users, each day
- One region serving same dimension type
- Terrible write/read performance

OK…I need a BREAK!!!

Design: Final

- Shard row to prevent hot-spotting
- Shard into N number of regions
- Significant improvement in read/write
- Prepend a salt to each key

Design: Final

Row key contains					
attribute value	**siteId**	**metric**	**attribute**	**timestamp**	**userGUID**

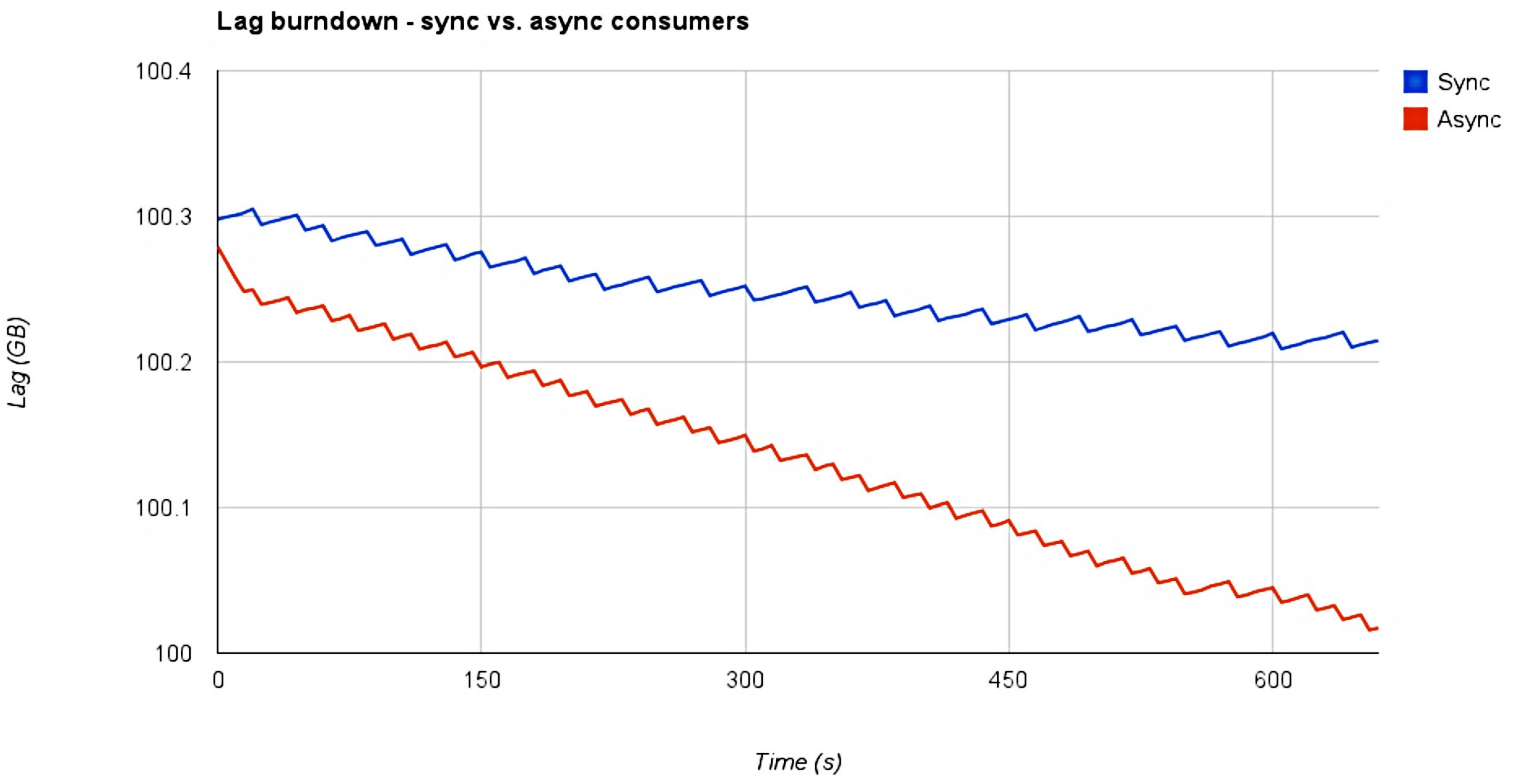
[salt]_len_[siteId]_len_[metric]_len_[dimension]_len_[value][timestamp]

len	is the integer length of the field following it
timestamp	is stored in day granularity
[salt]	is computed by first creating a hash from the siteId, metric, and dimension, then combining this with a random number between 0 and N (number of shards) for sharding

Design: Behavior Joins

- Complex segments contain many rules
- Each rule = one behavior = one row key
- Each row key returns a set of users
- OR = Full outer join
- AND = Inner join
- Done in memory for small rules
- Merged on disk for large rules

HBase Consumer Sync versus Async API



Segmentation Performance

- Seconds latency
- 40K puts/sec over 2 tables, 8 regions per table
- Scaling achieved through addition of regions
- Small segments calculated in msecs
- Mid-size segments in seconds
- Large segments calculated in 10s of seconds

richrelevance
is hiring!
richrelevance.com/careers
AMERICAN LASER SKINCARE
FLICK'S CANDY
KNBR 680

Thank You