TALKINGDATA GLOBAL MOBILE DATA SUMMIT

Pinot

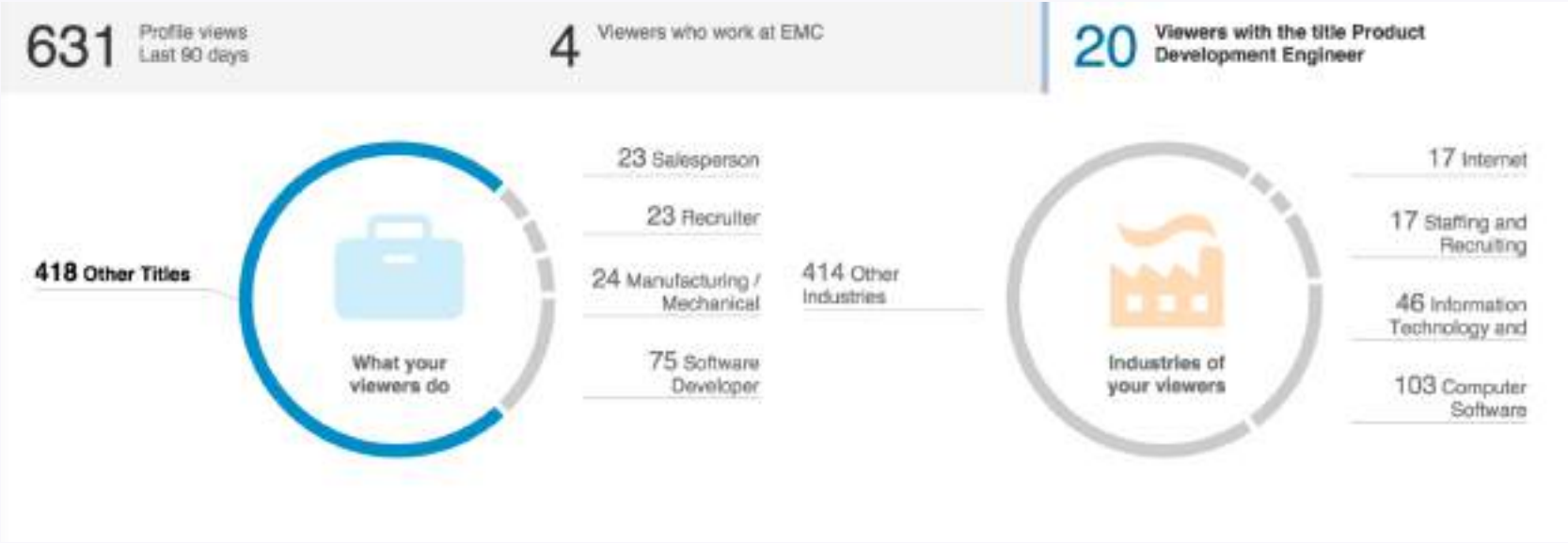
Real-time Analytics at LinkedIn

Greg Brandt (2015-09-11)

Agenda

- Pinot at LinkedIn
- Design and Architecture
- Segment Layout
- StarTree for Fast Aggregation
- ThirdEye for Business Monitoring

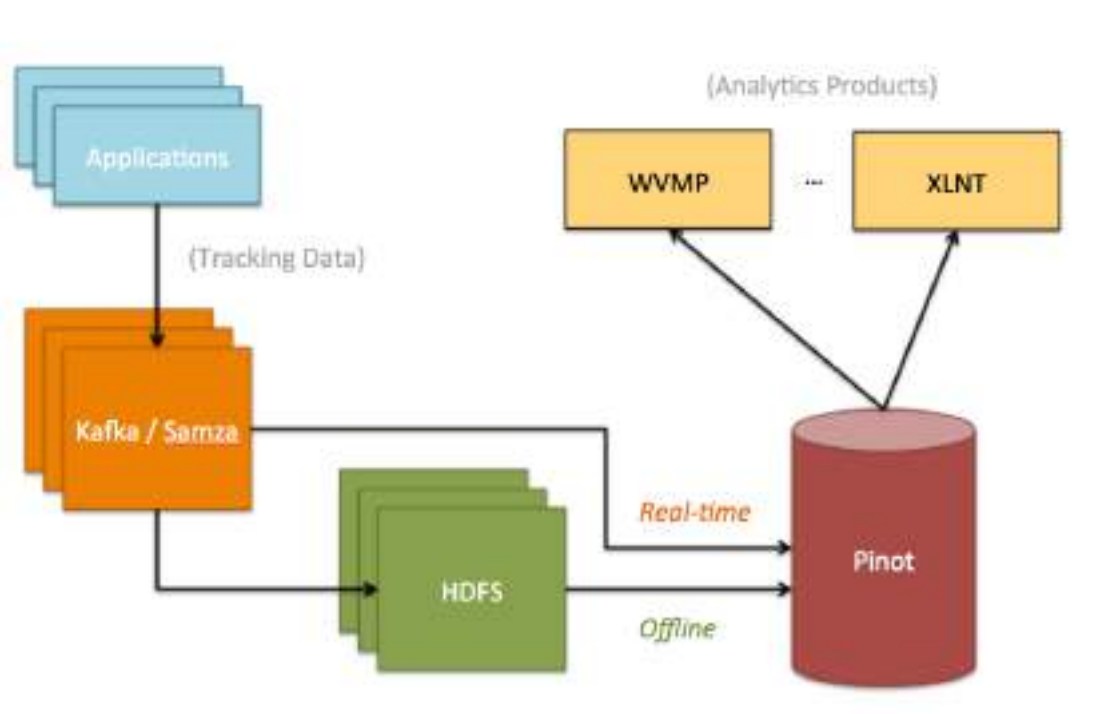
- Who’s Viewed My Profile
  - (member)



- XLNT
  - A/B testing
  - (internal)

- Ad campaigns
  - (customers)

Data Analytics Ecosystem at LinkedIn



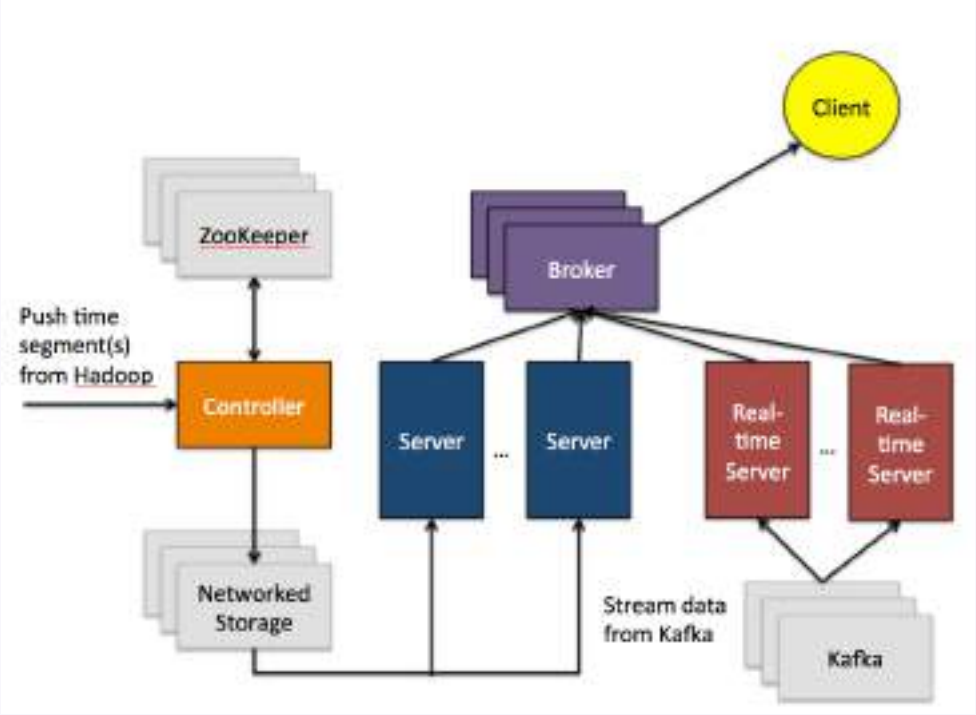
Goals

- SQL-like interface
- Columnar storage and indexing
- Real-time data load

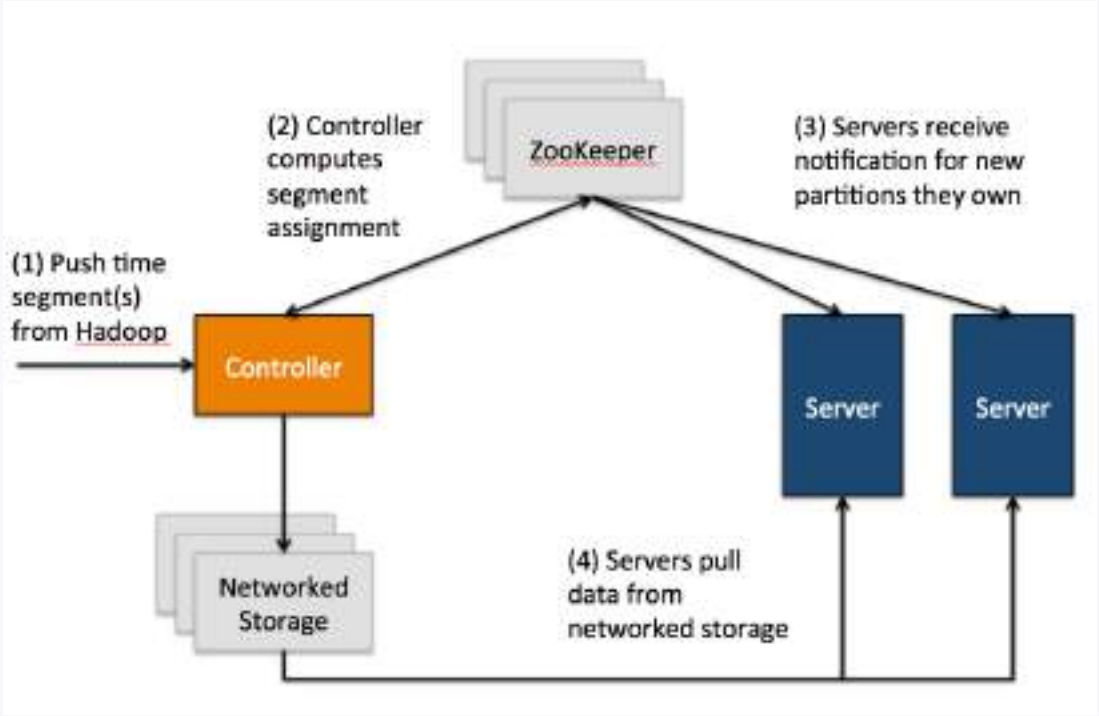
Non-goals

- JOIN, UDF (unpredictable performance)
- Not source of truth
- Mutability
  - Immutability allows trivial data distribution

Pinot Architecture

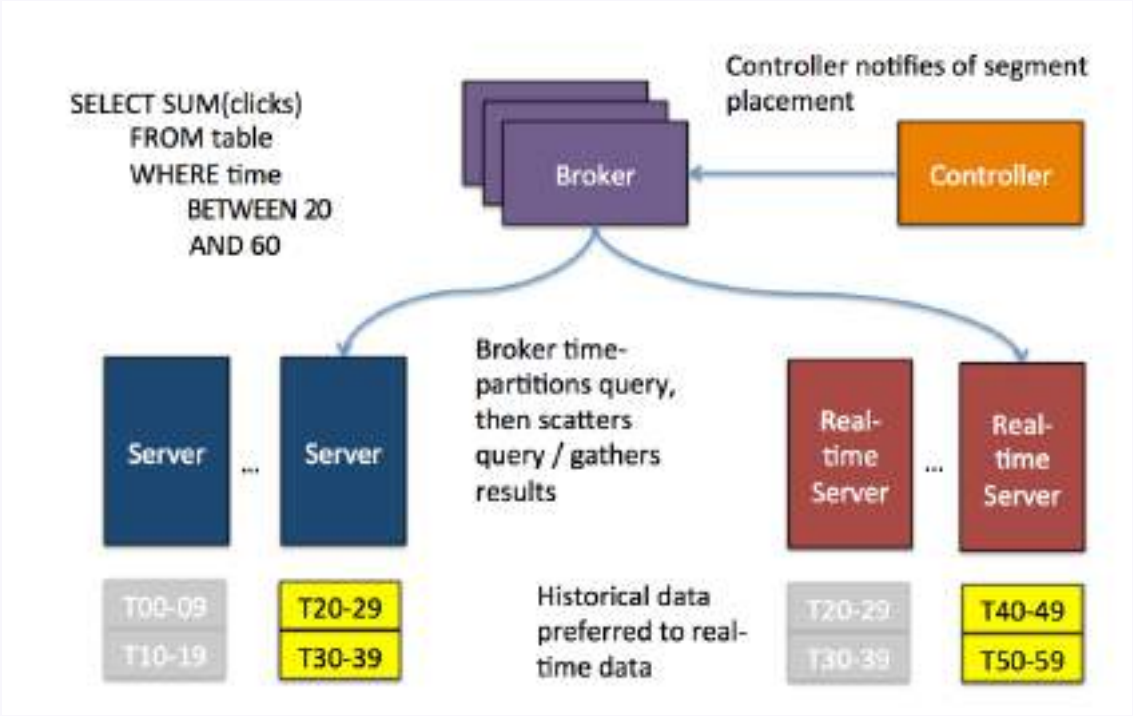


Segment Assignment



- A framework for building distributed, stateful applications
- Developed at LinkedIn for Espresso, (distributed OLTP database)
- Leveraged in Pinot for cluster management (Controller uses Helix libraries)
- http://helix.apache.org

Query Execution: Distributed



1. Parser
   - SELECT COUNT(*) … => Abstract Syntax Tree
2. Logical Planner
   - Operators like COUNT, WHERE, etc.
3. Physical Planner
   - Consider data layout (sort key, dictionary encoding, compression, etc.)

Layout: Columnar Storage

Doc Id	Model	Year	Price	tags
1	Toyota	1997	7500	moon-roof, reliable
2	Toyota	2001	7500	mp3, compact
3	Mazda	2005	1500	Manual, compact
4	Honda	2010	8000	leather, cool
5	Honda	2011	9000	automatic
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
10000	bmw	.	25000	moon-roof, reliable

RAW DATA

INDEX GENERATION

Doc Id	Model	Year	Price	tags
1	Toyota	1997	7500	moon-roof, reliable
2	Toyota	2001	7500	mp3, compact
3	Mazda	2005	1500	Manual, compact
4	Honda	2010	8000	leather, cool
5	Honda	2011	9000	automatic
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
10000	bmw	.	25000	moon-roof, reliable

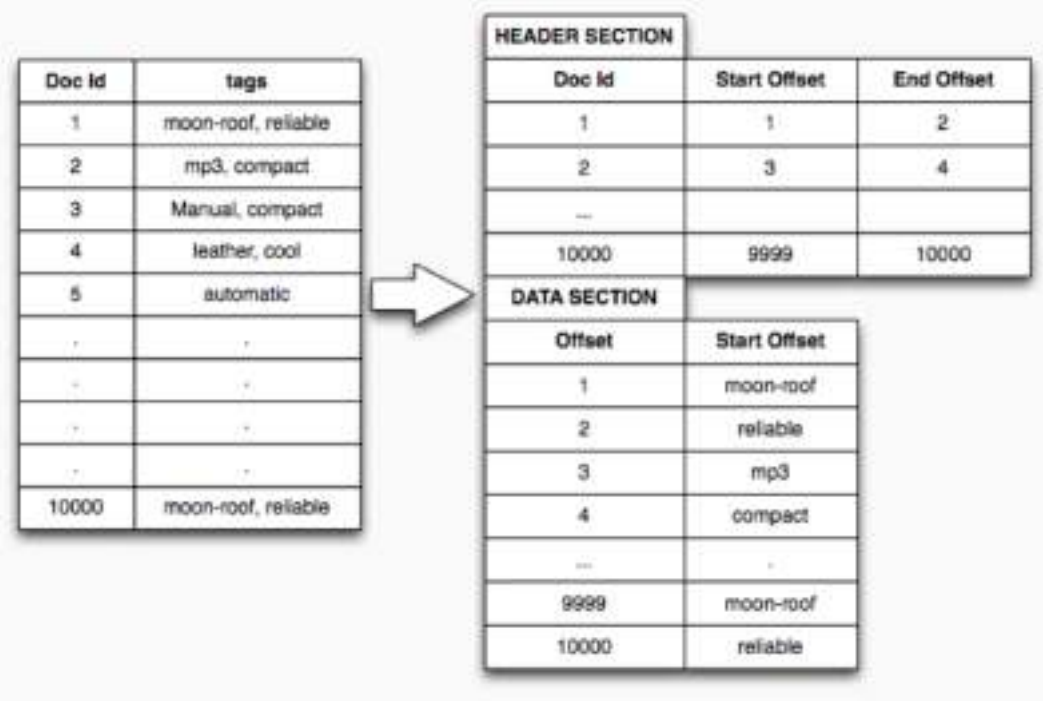
COLUMNAR DATA

Layout: Sorted Forward Index

Doc Id	Model
1	Acura
...	Acura
100	Acura
101	Bmw
..	Bmw
500	Bmw
.	.
.	.
99,999,900	Zagato
100,000,000	Zagato

Model	Start DocId	End DocId
Acura	1	100
Bmw	101	500
...		
...	.	.
...	.	.
...		
Zagato	99,999,900	100,000,000

- Indexes
  - Bitmap,
  - Roaring Bitmap,
  - Inverted
- Compression
  - Dictionary encoding,
  - P4Delta
- Multi-value columns



Doc Id	tags
1	moon-roof, reliable
2	mp3, compact
3	Manual, compact
4	leather, cool
5	automatic
.	.
.	.
.	.
.	.
10000	moon-roof, reliable

HEADER SECTION

Doc Id	Start Offset	End Offset
1	1	2
2	3	4
...		
10000	9999	10000

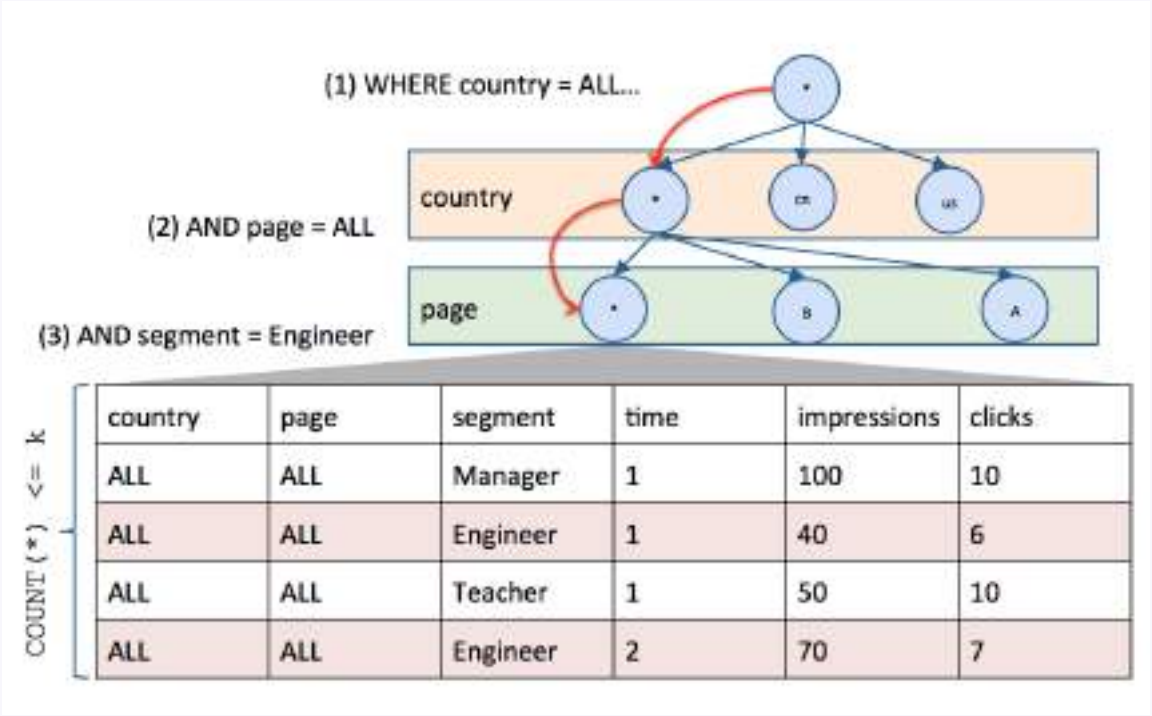
DATA SECTION

Offset	Start Offset
1	moon-roof
2	reliable
3	mp3
4	compact
...	.
9999	moon-roof
10000	reliable

Based on “Iceberg-Cubes” (Bottom-Up)

Invariant: scan <= k records to answer any aggregation query

StarTree Index Structure



country	page	segment	time	impressions	clicks
ALL	ALL	Manager	1	100	10
ALL	ALL	Engineer	1	40	6
ALL	ALL	Teacher	1	50	10
ALL	ALL	Engineer	2	70	7

- Consider a data set with two metrics (clicks, impressions), and four dimensions (country, page, user segment, time)
- ~3B raw events per day
- 196 countries, 100 pages, 25 user segments
- Store documents at 1 minute granularity

Let’s calculate aggregate time series for one week…

```
SELECT SUM(clicks),SUM(impressions)
    FROM table
    WHERE time BETWEEN
        2015-08-01 AND
        2015-08-08
    GROUP BY (country, time)
```

*StarTree space complexity largely impacted by data skew; **1.5 to 2 times raw data size** in practice*

Assuming scan speed of ~40M documents per second

Method	# Docs Scanned	Time Complexity	Space Complexity
Ad-hoc Pig / MR	21B	**8.75 minutes**	Constant
Time roll-up	705M	**2.04 minutes**	Constant
Full cubing	Constant	Constant	**128.66 GB / week**
StarTree (k = 100)	28M	**0.7 seconds**	O(leaves * k)

1. Take highest cardinality dimension, Dh
2. Compute aggregates after removing Dh
3. Append aggregates to segment, sort on Dh
4. For each value of Dh, *if there are greater than k documents in sub-table*, repeat on sub-table
   - Including the new aggregate value, “*”

Building StarTree in Pinot

```
>> tree testTable_0_127_1/
testTable_0_127_1/
├── D0.dict
├── D0.sv.unsorted.fwd
├── M0.dict
├── M0.sv.unsorted.fwd
...
├── creation.meta
├── daysSinceEpoch.dict
├── daysSinceEpoch.sv.unsorted.fwd
├── metadata.properties
└── startree
    ├── D0.dict
    ├── D0.sv.unsorted.fwd
    ├── M0.dict
    ├── M0.sv.unsorted.fwd
    ...
    ├── daysSinceEpoch.dict
    ├── daysSinceEpoch.sv.unsorted.fwd
    ├── metadata.properties
    └── startree.ser
```

The original Pinot segment, now partially sorted on dimension combination

The aggregate Pinot segment, containing partial aggregates (i.e. any StarTree prefix involving a “*” node)

StarTree speeds up Pinot queries two ways…

1. Aggregation queries can use partial aggregates, bound scan time by *k*

2. Raw segment is partially sorted by StarTree path dimension values, improves filter queries

- Leverage StarTree to to monitor highly-dimensional, aggregate time-series business metrics in real-time
- Interactive visualization tools
  - Drill-down, guided root-cause analysis
- Automated anomaly detection algorithms
  - Data-driven OLAP cube exploration
  - Detect spikes, step-level shifts

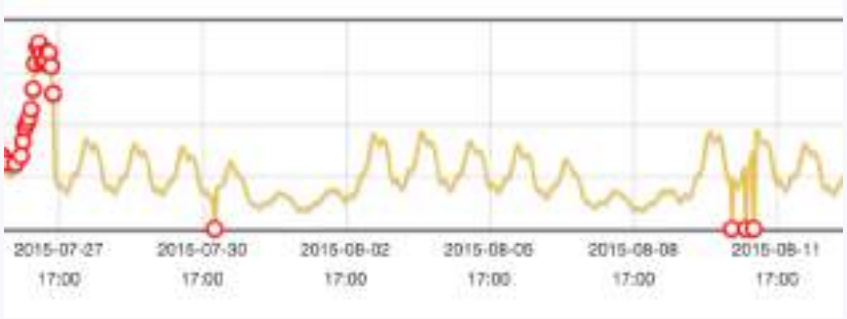
- Used in financial sector
- E.g. Showing y/3y on 2008-08-08, for S&P 500
- Click on cell to fix `ticker = 'aig'` in WHERE clause, drill down

Financial services like AIG and CitiGroup

Anomaly Detection: Exploration

- Explore OLAP cube top-down (start at overall aggregate metric)
- Use heuristic (like total volume) to prune search, determine dimension exploration order
- Compute GROUP BY (next_dimension, time) at each level, and run anomaly detection algorithm on each time series

- Model time series structural components: seasonality + trend + noise
- Best for detecting spikes/dips
- Applied in Ads metric monitoring



- Hypothesis testing to determine whether the pattern within time series interval differs from the rest
- Best for detecting step-level changes

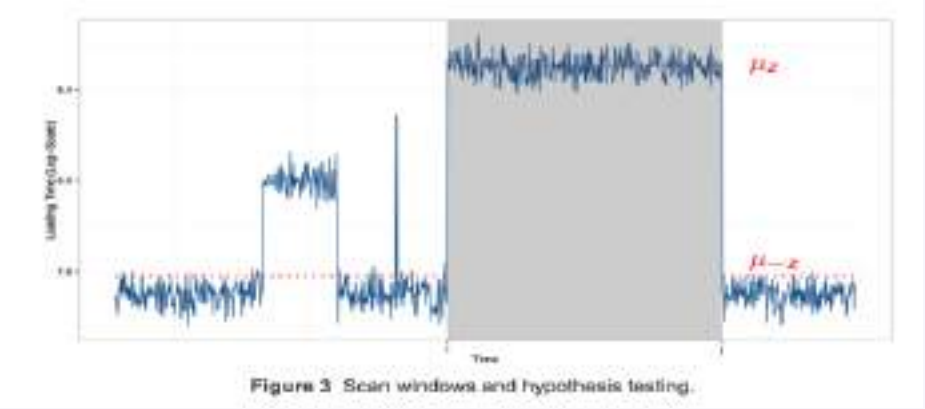
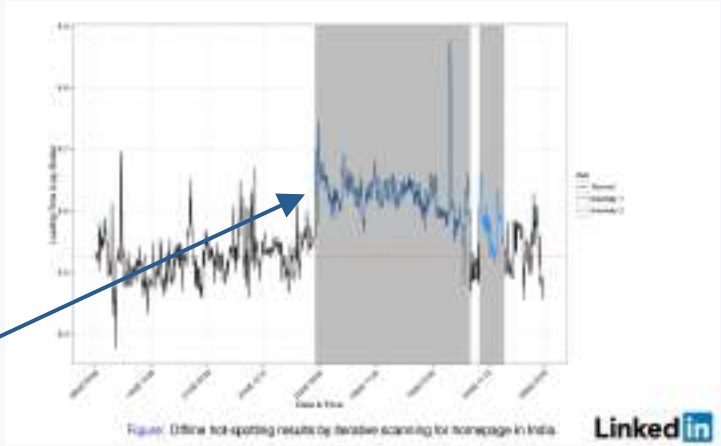


Figure 3 Scan windows and hypothesis testing.

Anomalies in Home Page download time, in India



Questions?

gbrandt@linkedin.com


T11
TALKINGDATA GLOBAL MOBILE
DATA SUMMIT