Real-time analytical query processing and predictive model building on high dimensional document datasets with timestamps

Debasish Das

Data & Artificial Intelligence, Verizon

Contributors

Algorithm: Santanu Das, Zhengming Xing

Platform: Ponrama Jegan

Frontend: Altaff Shaik, Jon Leonhardt

verizon✓

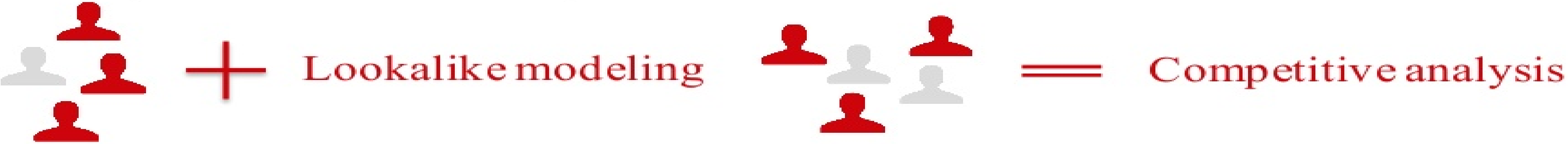
SPARK
SUMMIT
EAST 2017

Data Overview

- Location data
  - Each srcip defined as unique row key
  - Provides approximate location of each srcip
  - Timeseries containing latitude, longitude, error bound, duration, timezone for each srcip
- Clickstream data
  - Contains clickstream data of each row key
  - Contains startTime, duration, httphost, httpuri, upload/download bytes, httpmethod
  - Compatible with IPFIX/Netflow formats

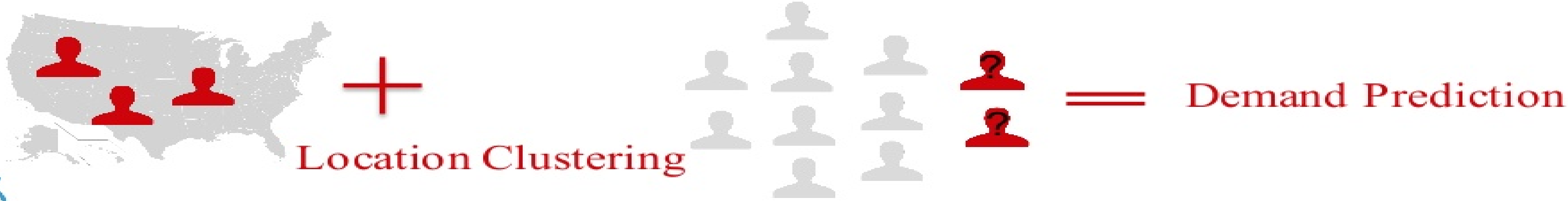
Marketing Analytics

- Aggregate Anonymous analysis for insights
- Spark Summit Europe 2016



+ Lookalike modeling = Competitive analysis

- Spark Summit East 2017



+ Location Clustering = Demand Prediction

Data Model

- Schema: srcip, timestmap, tld, zip, tldvisits, zipvisits
- Dense dimension, dense measure
  - Data: 10.1.13.120, d1H2, company1.com, 94555, 2, 4
- Sparse dimension, dense measure
  - Data: 10.1.13.120, d1, {company1.com, company2.com}, {94555, 94301}, 10, 15
- Sparse dimension, sparse measure
  - Data: 10.1.13.120, d1, {company1.com, company2.com}, {94555, 94301}, {company1.com:4, company2.com:6}, {94555:8, 94301:7}
- Timestamp optional
- Competing technologies: PowerDrill, Druid, LinkedIn Pinot, Essbase

Lucene Document Mapping

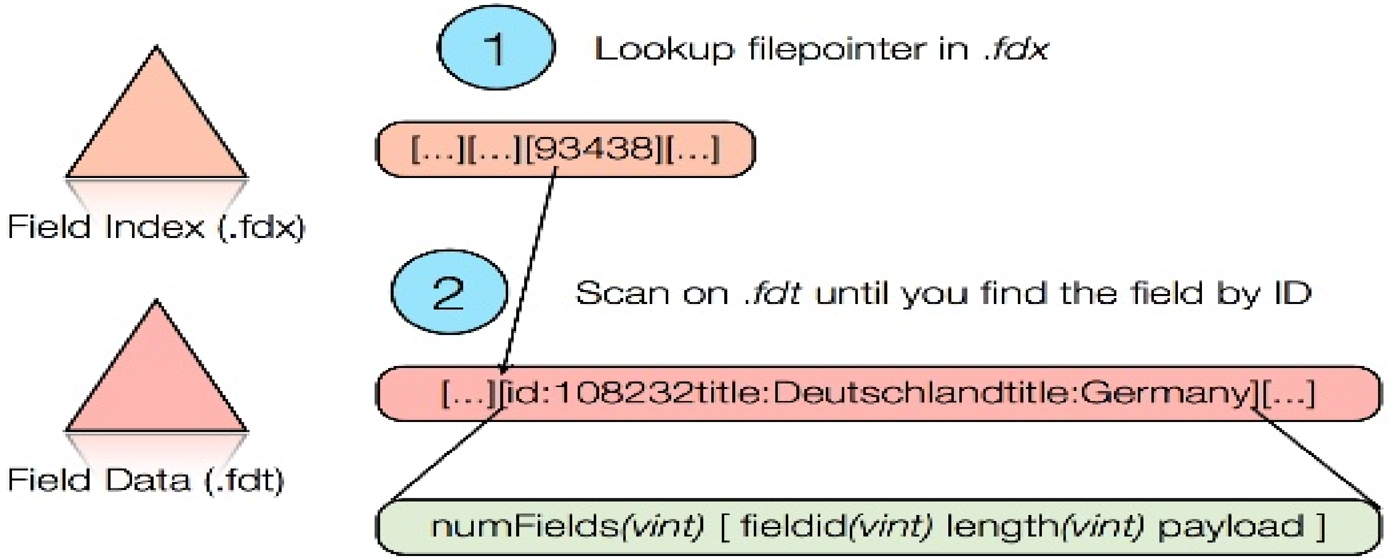
- Example
  Schema: srcip, timestamp, tld, zip, tldvisits, zipvisits
  Data: 10.1.13.120, d1, {company1.com, company2.com}, 94555, 10, 15
  Data: 10.1.13.120, d4, {company1.com, company3.com}, 94301, 12, 8
- DataFrame Row to Lucene Document mapping

schema	Row	Document	OLAP
srcip	StringType	Stored	Measure
timestamp	TimestampType	Stored	Dimension
tld	ArrayType[StringType]	Indexed + Stored	Dimension
zip	StringType	Indexed + Stored	Dimension
tld/zipvisits	IntegerType	Stored	Measure

SPARK SUMMIT EAST 2017

Lucene Storage

- Row storage: Spark Summit Europe 2016
  - – 2 indirect disk seeks for retrieval



1 Lookup filepointer in *.fdx*

[…][…][93438][…]

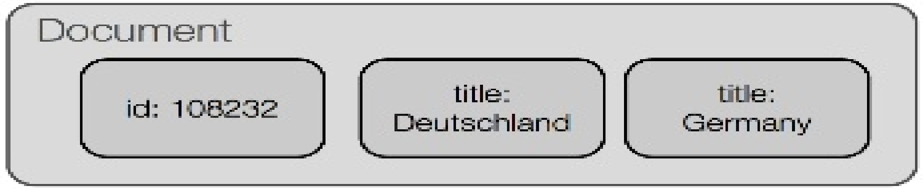
Field Index (.fdx)

2 Scan on *.fdt* until you find the field by ID

[…][id:108232title:Deutschlandtitle:Germany][…]

Field Data (.fdt)

numFields*(vint)* [fieldid*(vint)* length*(vint)* payload]



Document

id: 108232

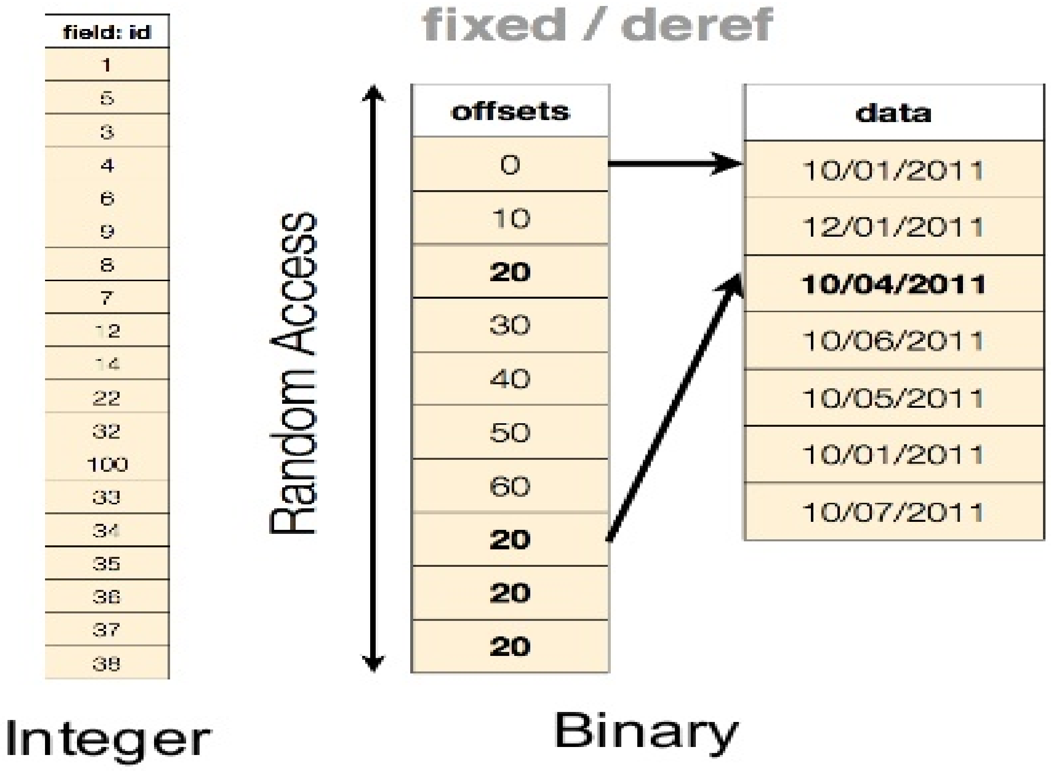
title: Deutschland

title: Germany

Reference: http://www.slideshare.net/lucenerevolution/willnauer-simon-doc-values-column-stride-fields-in-lucene

Lucene Column Store

- Column storage: Spark Summit East 2017
  - References: LUCENE-3108, LUCENE-2935, LUCENE-2168, LUCENE-1231
  - Cache friendly column retrieval: 1 direct disk seek
  - Integer column: Min-Max encoding
  - Numeric column: Uncompressed
  - Binary column: Referenced
  - Complex Type: Binary + Kryo



field: id
1
5
3
4
6
9
8
7
12
14
22
32
100
33
34
35
36
37
38

Integer

fixed / deref

offsets
0
10
20
30
40
50
60
20
20
20

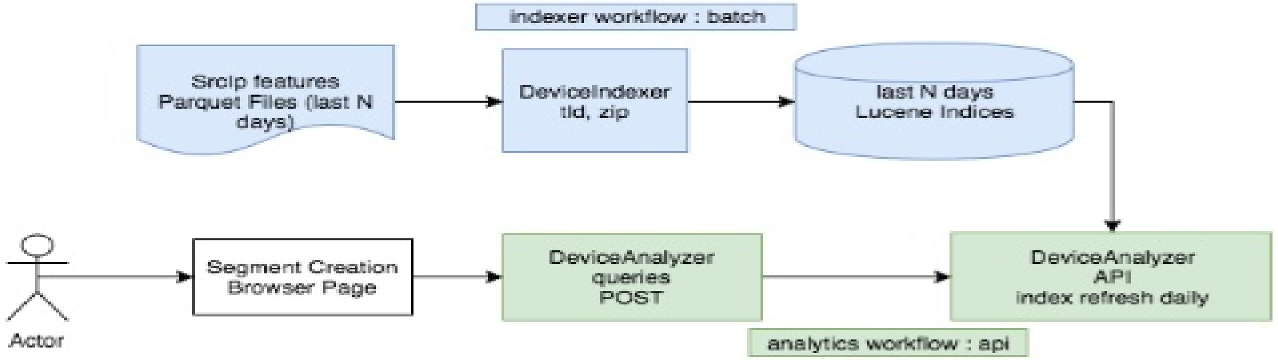
data
10/01/2011
12/01/2011
10/04/2011
10/06/2011
10/05/2011
10/01/2011
10/07/2011

Binary

SPARK SUMMIT EAST 2017

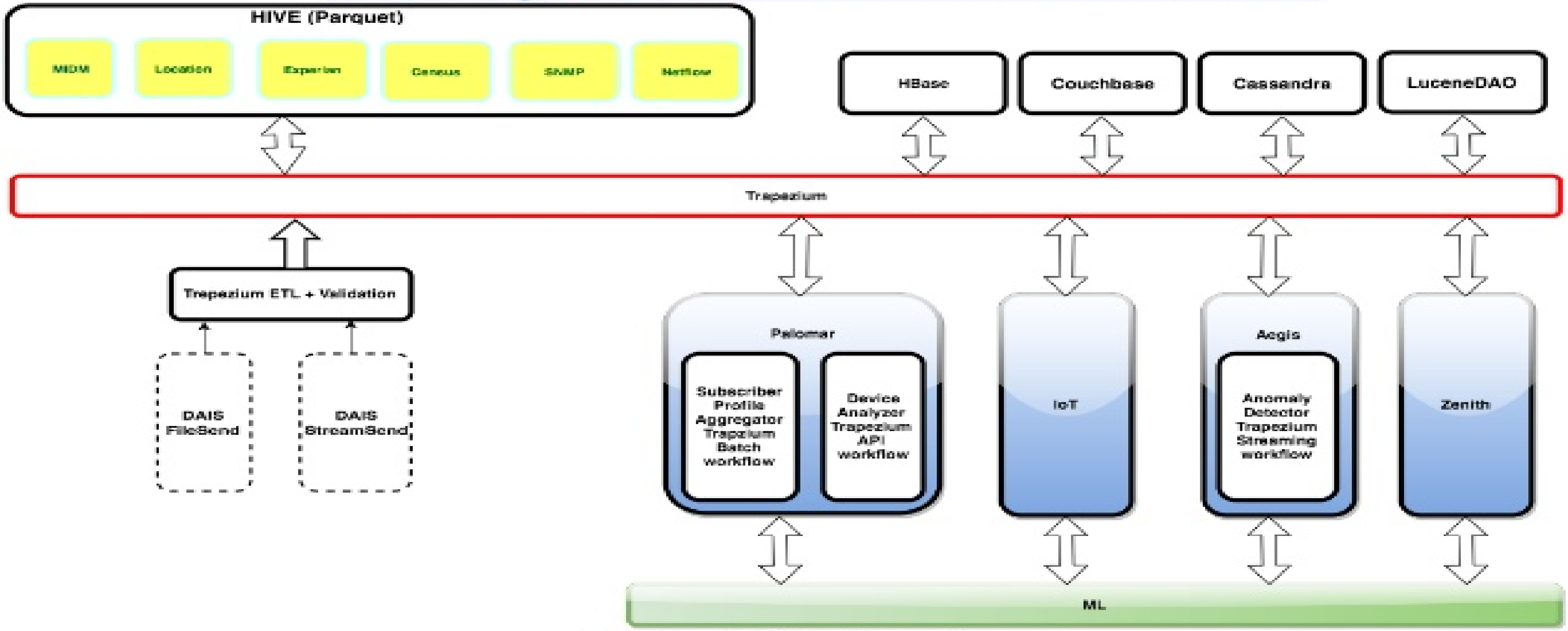
DeviceAnalyzer

- Goals
  - srcip/visits as dense measure
  - Real-Time queries
    - Aggregate
    - Group
    - Timeseries
  - Real-Time Timeseries forecast


indexer workflow : batch
Srcip features Parquet Files (last N days)
DeviceIndexer tld, zip
last N days Lucene Indices
Actor
Segment Creation Browser Page
DeviceAnalyzer queries POST
DeviceAnalyzer API index refresh daily
analytics workflow : api

Trapezium

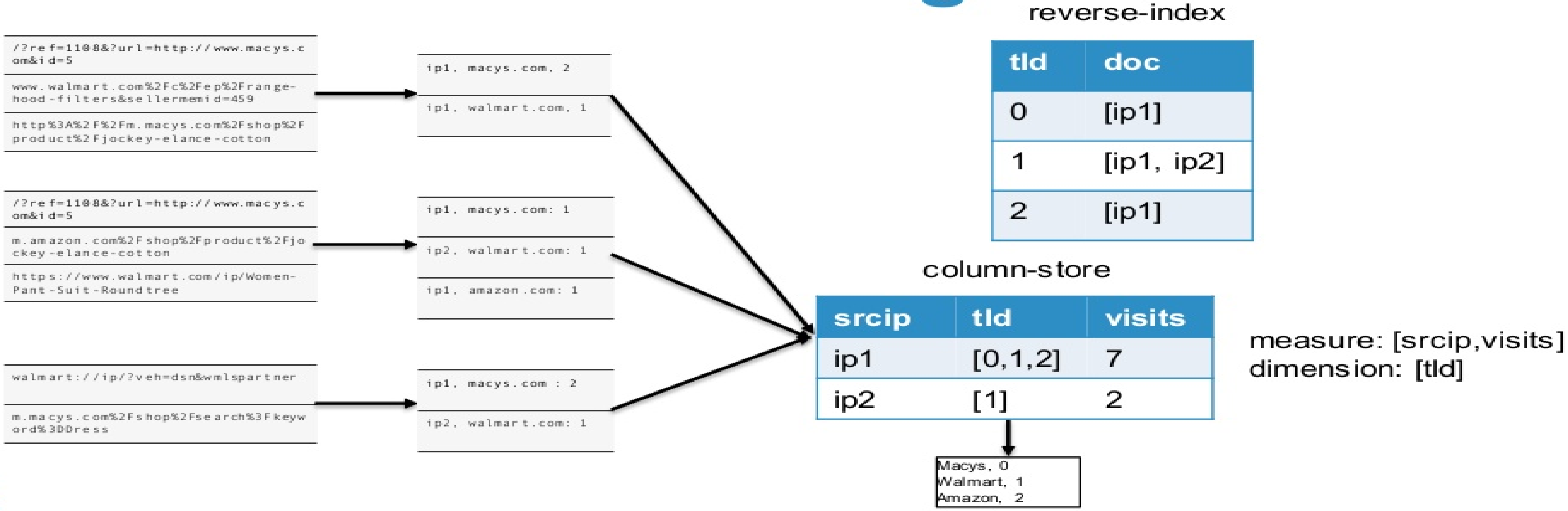
DAIS Open Source framework to build batch, streaming and API services
https://github.com/Verizon/trapezium


HIVE (Parquet)
MDM
Location
Experian
Census
SNMP
Netflow
HBase
Couchbase
Cassandra
LuceneDAO
Trapezium
Trapezium ETL + Validation
DAIS FileSend
DAIS StreamSend
Palomar
Subscriber Profile Aggregator Trapezium Batch workflow
Device Analyzer Trapezium API workflow
IoT
Aegis
Anomaly Detector Trapezium Streaming workflow
Zenith
ML

Trapezium LuceneDAO

- SparkSQL optimized for full scan
  - Column indexing not supported
- Fulfills Real-Time requirements for OLAP queries
- Lucene for indexing + storage per executor
- Spark operators for distributed aggregation
  - treeAggregate
  - mapPartition + treeReduce
- Features
  - Build Distributed Lucene Shards from Dataframe
  - Access saved shards through LuceneDAO for Analytics + ML pipelines
  - Save shards to HDFS for QueryProcessor like SolrCloud

LuceneDAO Indexing



```
/?ref=1108&?url=http://www.macys.com&id=5
www.walmart.com%2Fc%2Fep%2Frange-hood-filters&sellermemid=459
http%3A%2F%2Fm.macys.com%2Fshop%2Fproduct%2Fjockey-elance-cotton
```

```
ip1, macys.com, 2
ip1, walmart.com, 1
```

```
/?ref=1108&?url=http://www.macys.com&id=5
m.amazon.com%2Fshop%2Fproduct%2Fjockey-elance-cotton
https://www.walmart.com/ip/Women-Pant-Suit-Roundtree
```

```
ip1, macys.com: 1
ip2, walmart.com: 1
ip1, amazon.com: 1
```

```
walmart://ip/?veh=dsn&wmlspartner
m.macys.com%2Fshop%2Fsearch%3Fkeyword%3DDress
```

```
ip1, macys.com : 2
ip2, walmart.com: 1
```

reverse-index

tld	doc
0	[ip1]
1	[ip1, ip2]
2	[ip1]

column-store

srcip	tld	visits
ip1	[0,1,2]	7
ip2	[1]	2

measure: [srcip,visits]
dimension: [tld]

Macys, 0
Walmart, 1
Amazon, 2

LuceneDAO API

Index Creation

```
import trapezium.dal.lucene._
import org.apache.spark.sql.types._

object DeviceIndexer extend BatchTransaction {
process(dfs: Map[String, DataFrame], batchTime: Time): {
    df = dfs(“DeviceStore”)
    olapDf = rollup(df)
}

persist(df: DataFrame, batchTime: Time): {
    val dimensions = Set(“tld”, “zip”)
    val types = Map(“tld” -> LuceneType(true, StringType),
                    “srcip” -> LuceneType(false, StringType),
                    “visits” -> LuceneType(false, IntegerType))
    val dao = new LuceneDAO(“path”, dimension, types)
    dao.index(df, new Time(batchTime))
}
```

Query Processing

```
import trapezium.dal.lucene._
import org.apache.spark.sql.types._
```

Load:

```
val dimensions = Set(“tld”, “zip”)
val types = Map(“tld” -> LuceneType(true, StringType),
                “srcip” -> LuceneType(false, StringType),
                “visits” -> LuceneType(false, IntegerType))
val dao = new LuceneDAO(“path”, dimension, types)
dao.load(sc)
```

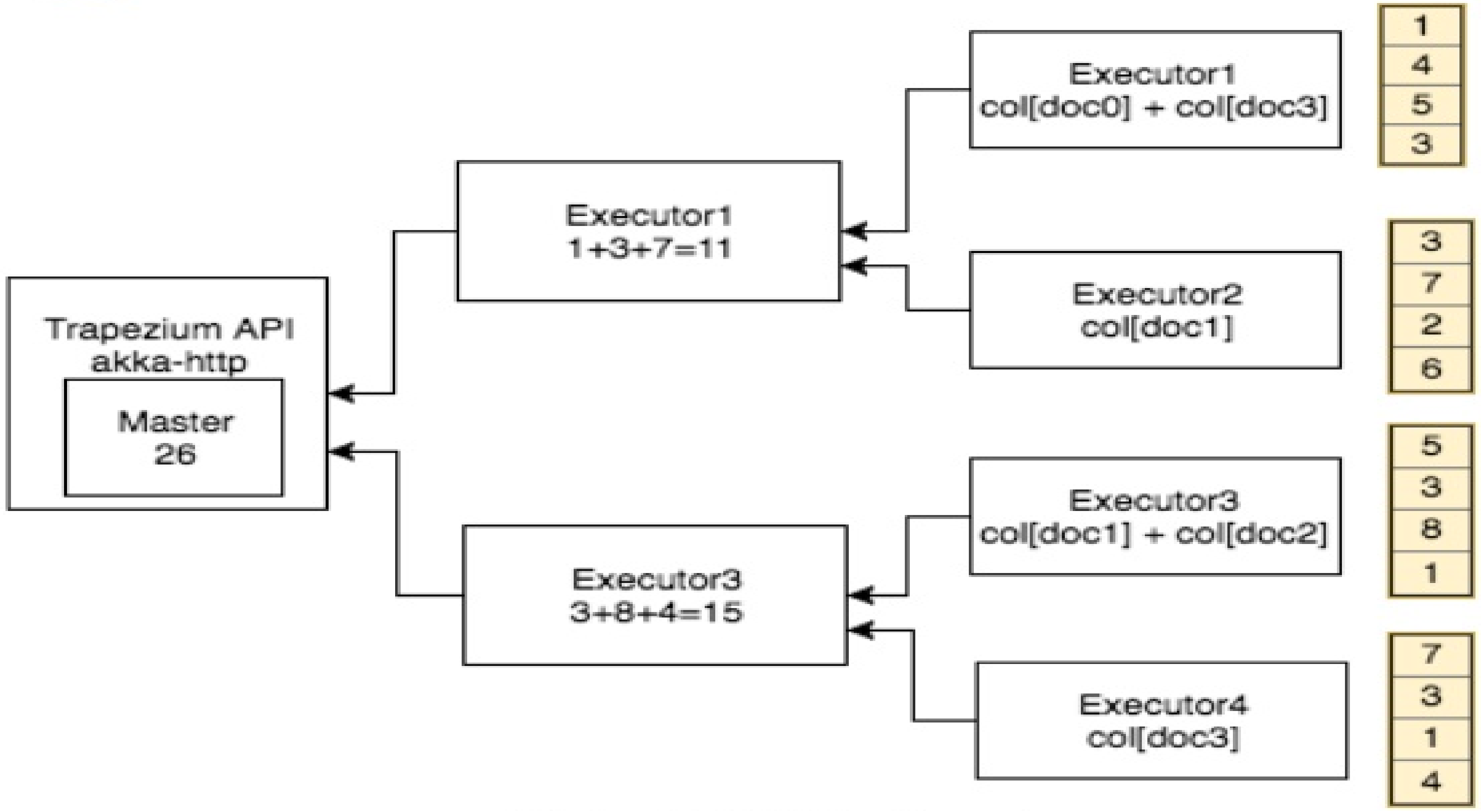
Queries:

```
dao.aggregate(query: String, measure: String, aggregator: String)
dao.group(query:String, dimension: String, measure: String,
          aggregator: String)
dao.timeseries(query: String, minTime: Long, maxTime: Long,
                    rollup: Long, measure: String, aggregator:
String)
dao.search(query: String, columns: Seq[String]): DataFrame
```

LuceneDAO Internals

- Retrieve documents with/without relevance
- Column Accessor over dimension + measures
- Disk / In-Memory Column Accessor
- C-store style while loops over dimension
- Spark ML style aggregators
- treeAggregate for distributed aggregation

Aggregation Architecture



Trapezium API
akka-http

Master
26

Executor1
1+3+7=11

Executor3
3+8+4=15

Executor1
col[doc0] + col[doc3]

Executor2
col[doc1]

Executor3
col[doc1] + col[doc2]

Executor4
col[doc3]

1
4
5
3

3
7
2
6

5
3
8
1

7
3
1
4

Index Generation

- Dataset details:
  57M devices, 4.2B docs
- Parquet: 79 GB
- Lucene Reverse Index: 16 GB
- Lucene DocValues: 59.6 GB
- Global Dictionary Size: 5.5 MB
- Executors: 20 Cores: 8
- RAM Driver: 16g Executor: 16g

- Runtime
  - Parquet:
    - 1831.87 s
  - Dictionary:
    - 213.7 s
  - Index + Stored:
    - 360 s

SPARK SUMMIT EAST 2017

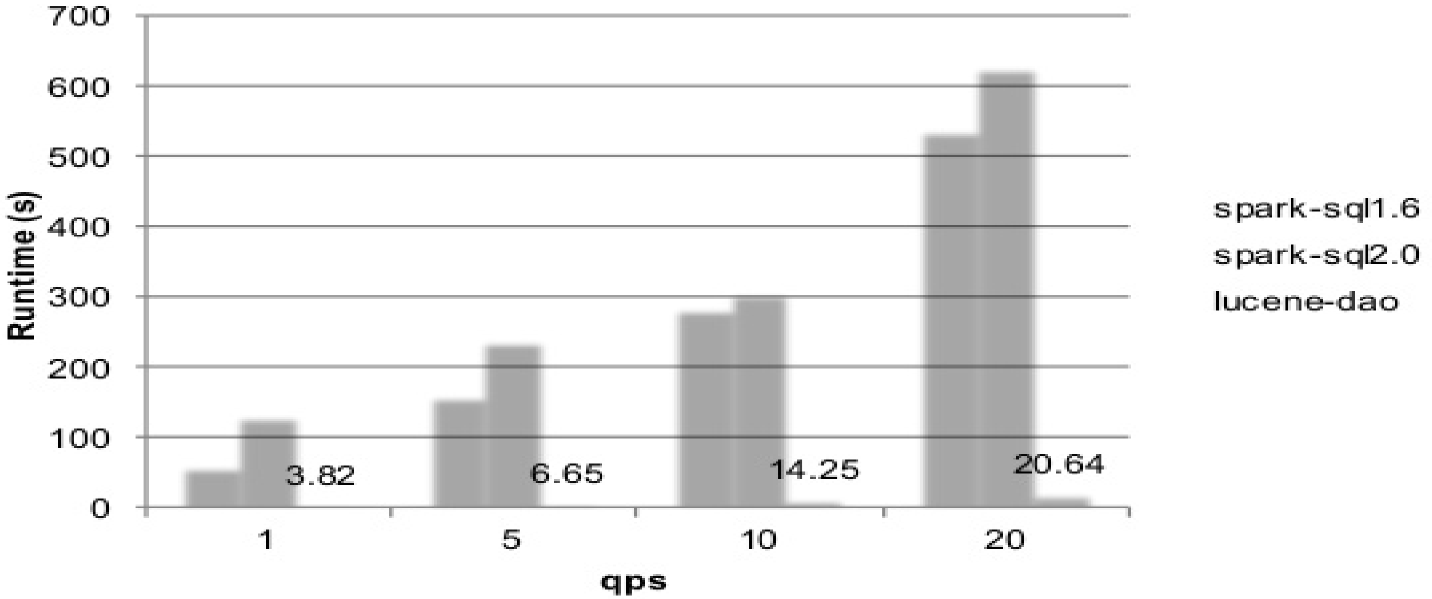
Aggregate Queries

- HashSet aggregation
- SparkSQL

```
df.select(”srcip”,"tld")
.where(array_contains(df("tld"),
”company1.com"))
.agg(countDistinct(”srcip") as "visits")
.collect()
```

- LuceneDAO

```
dao.aggregate("tld:company1.com”,
"srcip", "count")
```



SPARK SUMMIT EAST 2017

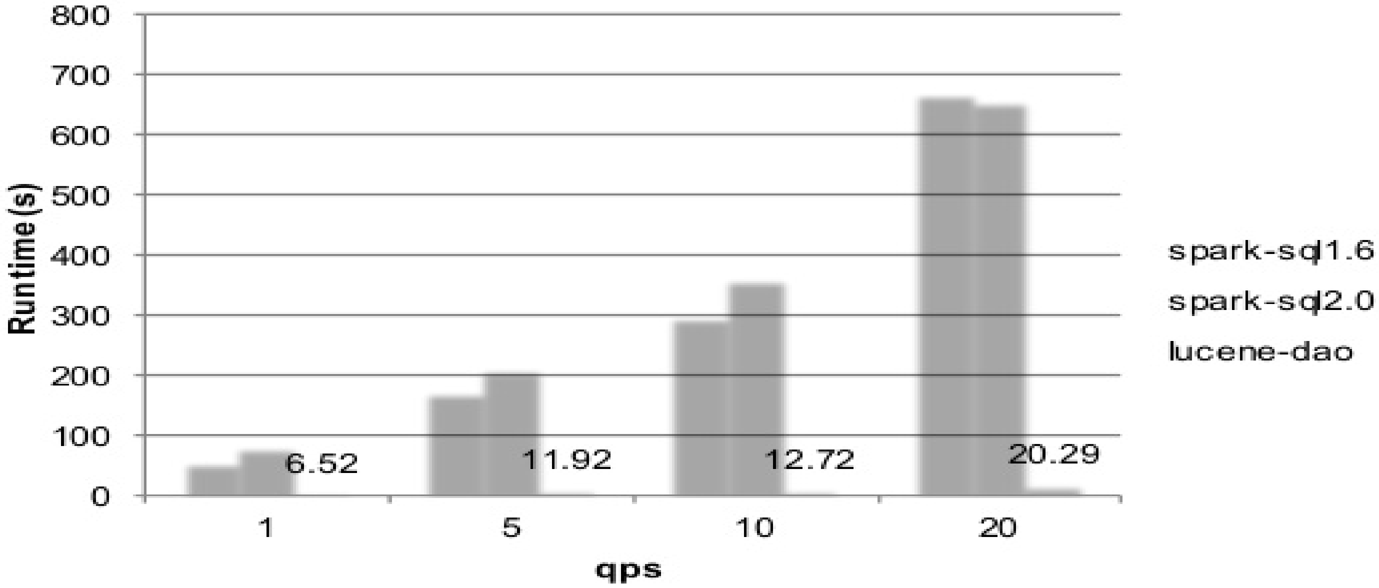
Group Queries

- HLL aggregation
- SparkSQL

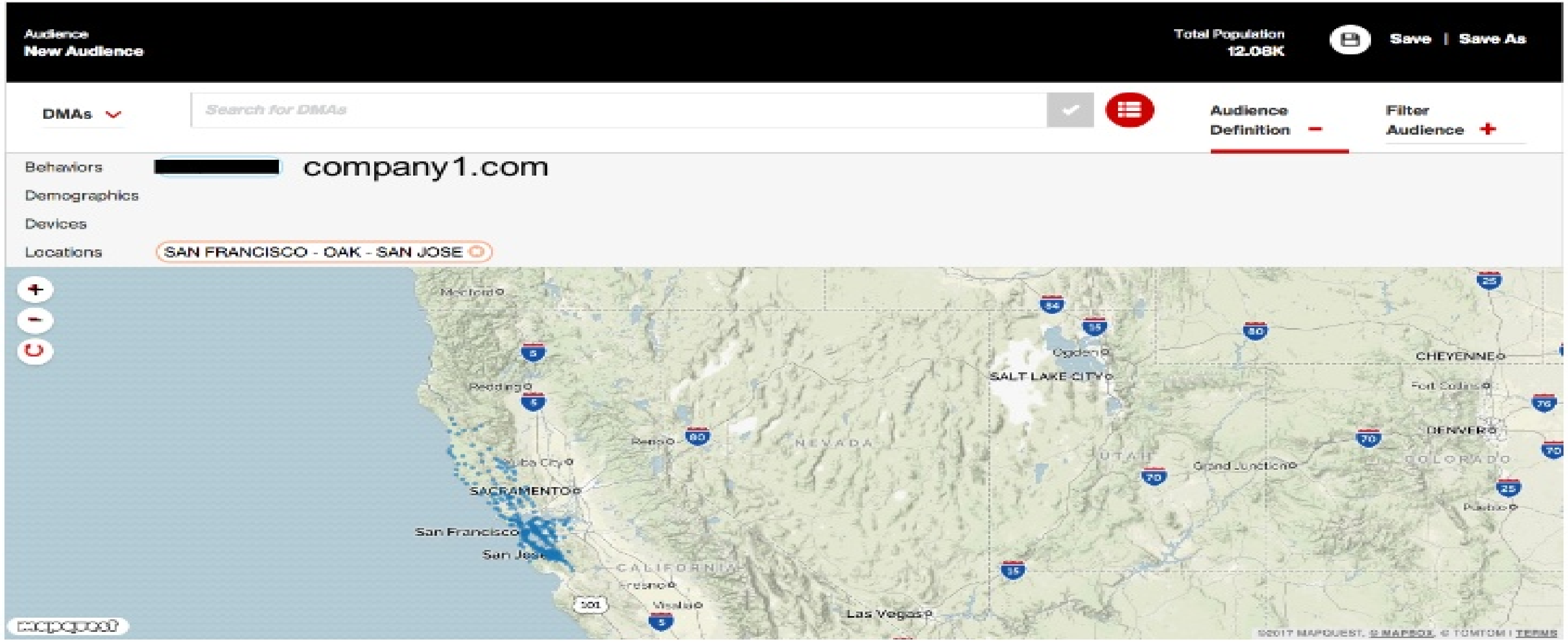
```
df.select(”srcip”,"tld", “zip”)
.where(array_contains(df("tld"),
"company1.com"))
.select(“zip”, “srcip”).groupBy(“zip”)
.agg(approxCountDistinct(”srcip") as
"visits")
.collect()
```

- LuceneDAO

```
dao.aggregate("tld:company1.com", "srcip",
"count")
```



Device Heat-Map



Audience
New Audience

Total Population
12.08K

Save | Save As

DMAs

Search for DMAs

Audience Definition −

Filter Audience +

Behaviors [redacted] company1.com

Demographics

Devices

Locations SAN FRANCISCO - OAK - SAN JOSE

Information contained herein is provided AS IS and subject to change without notice. All trademarks used herein are property of their respective owners.

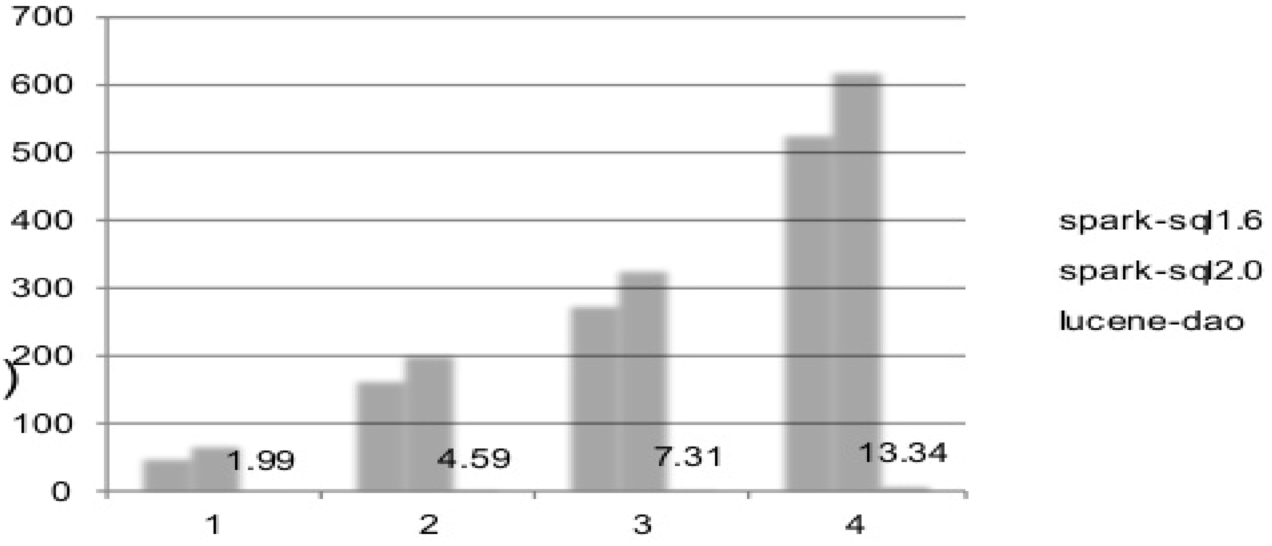
Timeseries Queries

- HLL aggregation
- SparkSQL

```
df.select(“time”,”srcip”,"tld”)
.where(array_contains(df("tld"),
”company1.com"))
.select(“time”, “srcip”).groupBy(“time”)
.agg(approxCountDistinct(”srcip") as "visits")
.collect()
```

- LuceneDAO

```
dao.aggregate("tld:company1.com", "srcip",
"count")
```



TimeSeries Forecast

- Given a query:

select

timestamp, (srcip) as deviceCount

where

tld=‘company1.com’ AND state=‘CA’

- Predict deviceCount for next timestamp
- Forecast deviceCount for next N timestamps

Trapezium ML

TimeSeriesKNNRegression.predict

Input:

timeseries: Array[Double]

topk: Int

featureDim: Int

normalize: Boolean

multiStep: Int

metric: KernelType=Euclidean

Output:

predicted values: Array[Double]

Forecast Service

Powered by Trapezium API

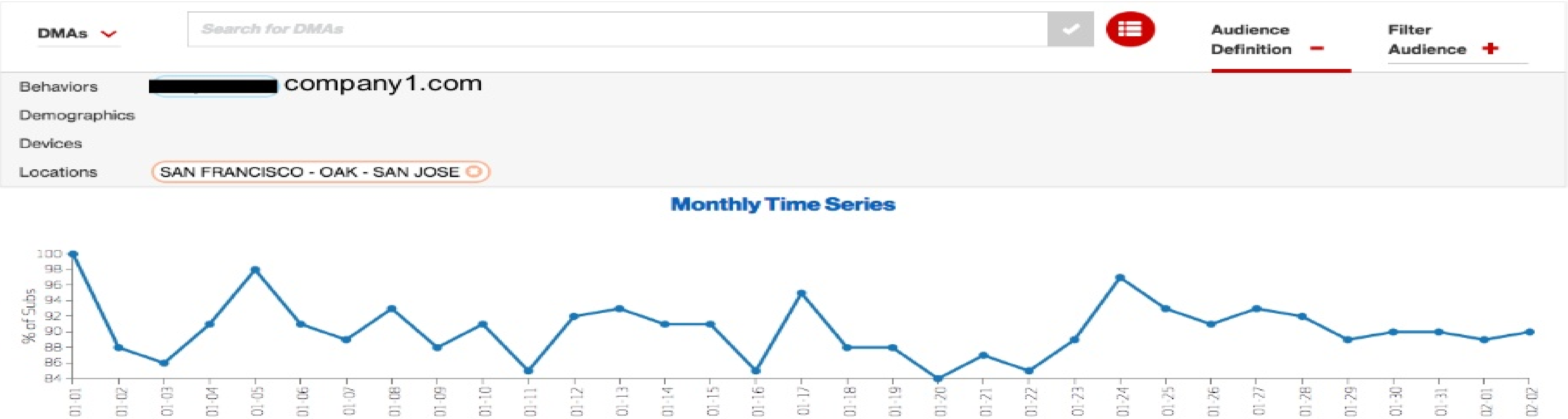
```
httpServer = {
  provider = "akka"
  hostname = "localhost"
  port = 19999
  contextPath = "/"
  endPoints = [{
     path = “analyzer-api"
     className =
"TimeseriesEndPoint"
   }]
}
```

```
class TimeseriesEndPoint(sc: SparkContext)
extends SparkServiceEndPoint(sc) {
override def route : timeseriesRoute

val types = Map(“tld” -> LuceneType(true, StringType),
                          “srcip” -> LuceneType(false, StringType),
                          “visits” -> LuceneType(false, IntegerType))
val dao = new LuceneDAO(“path”, dimension, types)
dao.load(sc)

def timeseriesRoute : {
  post { request => {
    ts = dao.timeseries(request, minTime, maxTime, rollup,
                              “srcip”, “count_approx”)
    predicted = TimeseriesKNNRegression.predict(ts, topk=5,
                         featureDim=3, normalize=false, multiStep=5,
                         metric=Euclidean)
    generateResponse(ts, predicted)
  }
}
```

Device-Count Forecast



Behaviors [redacted] company1.com

Demographics

Devices

Locations SAN FRANCISCO - OAK - SAN JOSE

Monthly Time Series

5 step prediction

Thank You.
Q&A

Join us and make machines intelligent
Data & Artificial Intelligence Systems
499 Hamilton Ave, Palo Alto
California

verizon✓

SPARK SUMMIT
EAST 2017