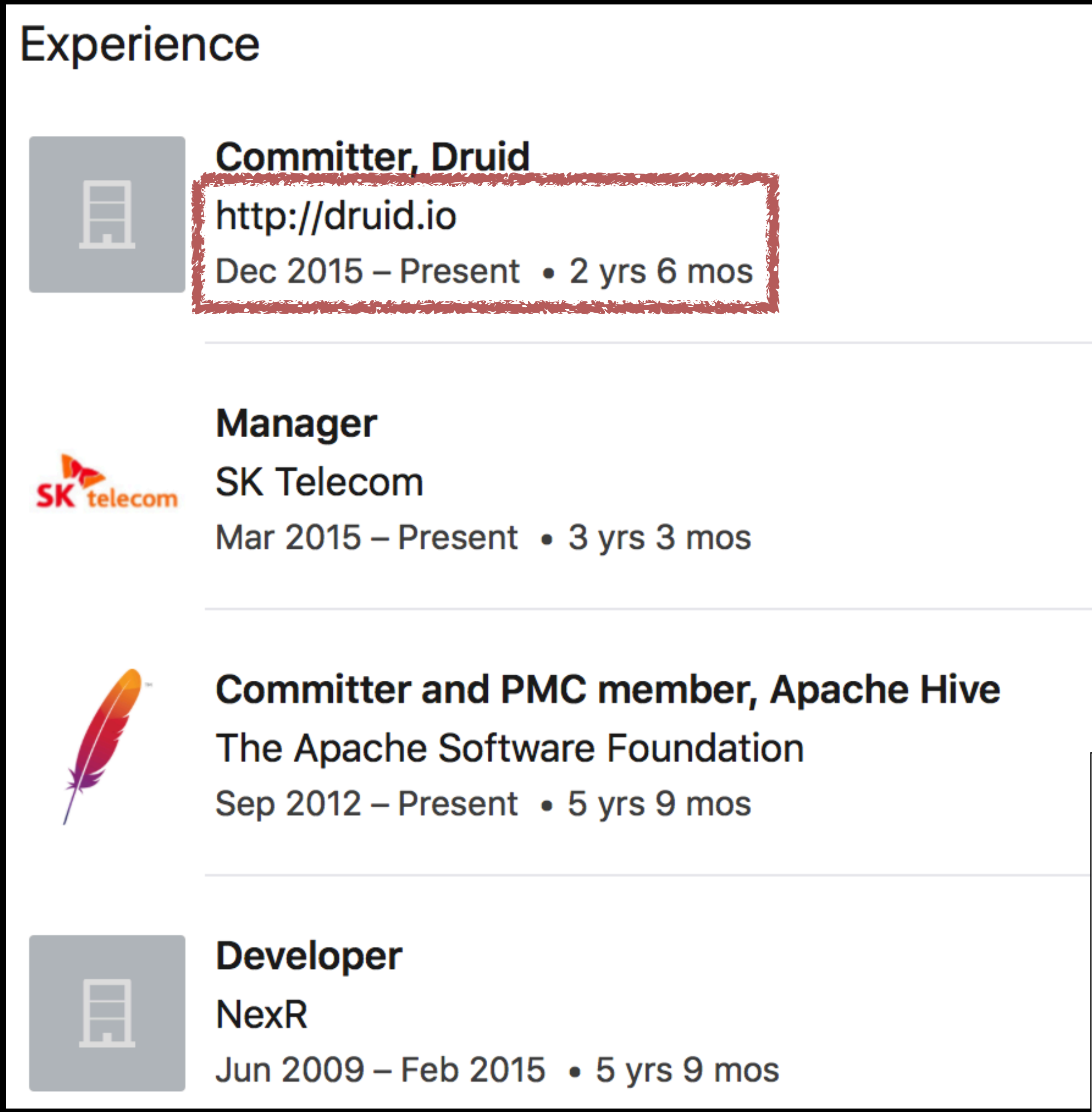
2.5 year of Druid-ing

Druid meetup, 7th

2018.06.11, Navis (SK Telecom)

Who am I

- Navis : 2.5 Druid / 17 Java



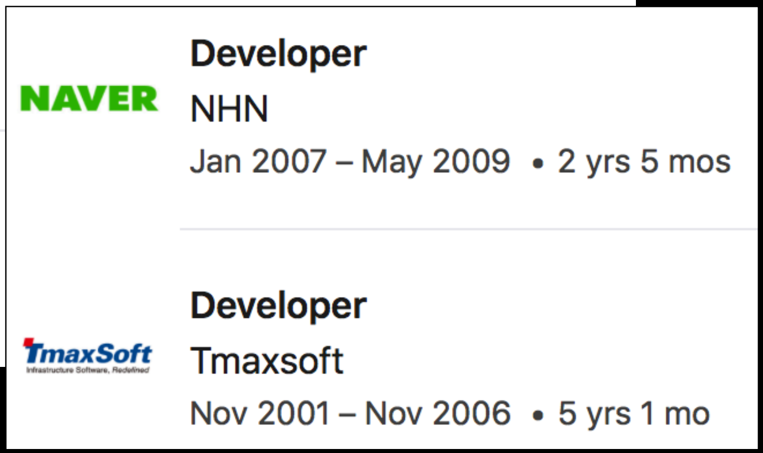
Experience

Committer, Druid
http://druid.io
Dec 2015 – Present • 2 yrs 6 mos

Manager
SK Telecom
Mar 2015 – Present • 3 yrs 3 mos

Committer and PMC member, Apache Hive
The Apache Software Foundation
Sep 2012 – Present • 5 yrs 9 mos

Developer
NexR
Jun 2009 – Feb 2015 • 5 yrs 9 mos



Developer
NHN
Jan 2007 – May 2009 • 2 yrs 5 mos

Developer
Tmaxsoft
Nov 2001 – Nov 2006 • 5 yrs 1 mo

Agenda

- Brief introduction to Druid
- Follow-up works in SKT (of previous meet-ups)

Introduction to Druid

* Nishant Bangarwa: An Introduction to Druid (DataWorks Summit, 2018.5)

History

- Initial Use case

  Power ad-tech analytics product (Metamarkets, 2011)

- Apache License v2 (2015. 2)

  Initially open sourced in late 2012 as LGPL v2

  Imply launched (2015.10)

  Apache incubator (2018. 3)

- Requirements

  Query any combination of metrics and dimensions

  Scalability : trillions of events/day

  Real-time : data freshness

  Streaming Ingestion

  Interactive : low latency queries

* Nishant Bangarwa: An Introduction to Druid (DataWorks Summit, 2018.5)

Motivation

- Business Intelligence Queries

  Arbitrary slicing and dicing of data

  Interactive real time visualizations on Complex data streams

- Answer BI questions

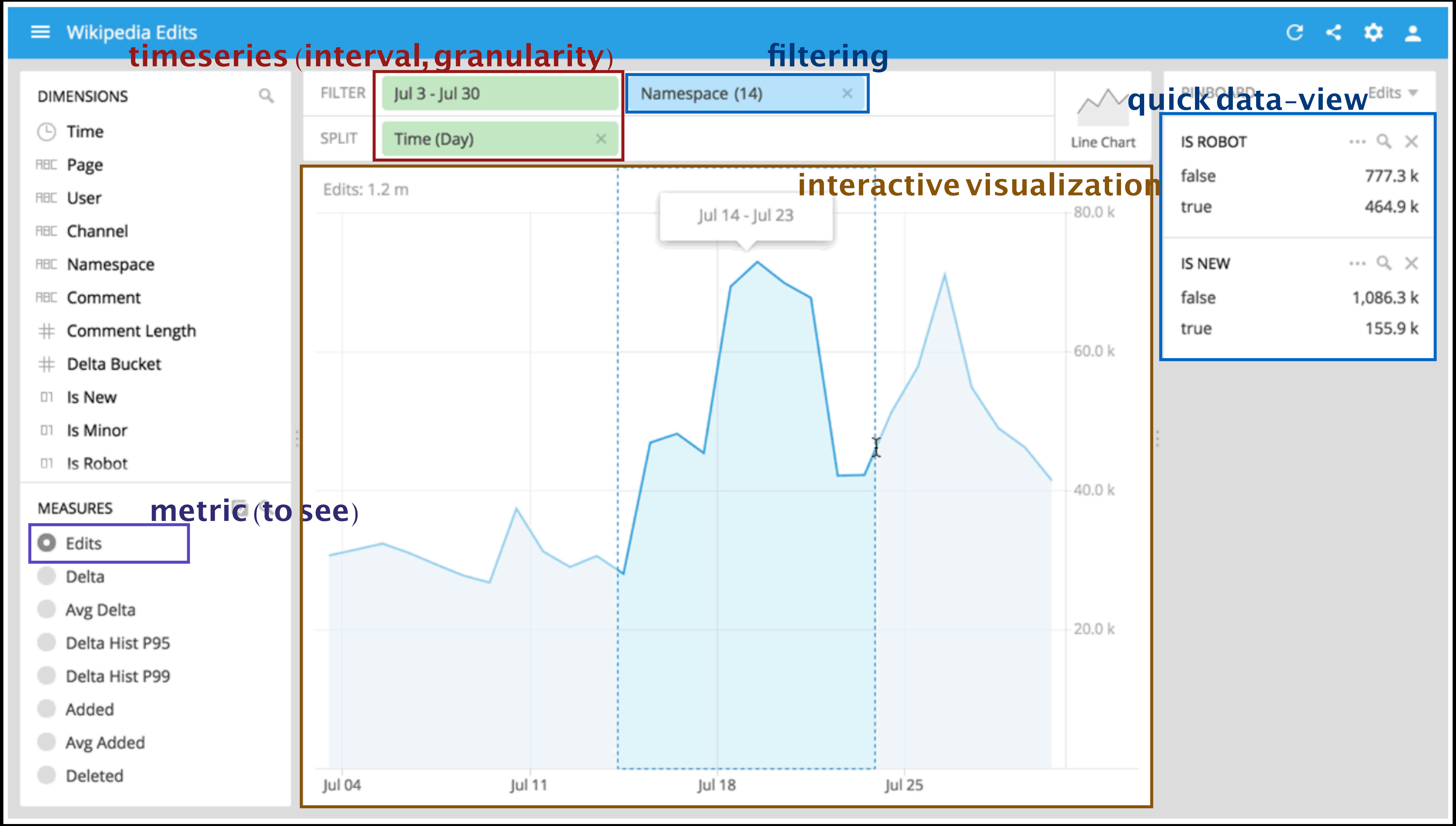
  How many unique male visitors visited my website last month ?

  How many products were sold last quarter broken down by a demographic and product category ?

- Not interested in dumping entire dataset

  Optimized to make highly selective/aggregated data

Motivation



timeseries (interval, granularity)

filtering

quick data-view

interactive visualization

metric (to see)

FILTER: Jul 3 - Jul 30 | Namespace (14)

SPLIT: Time (Day)

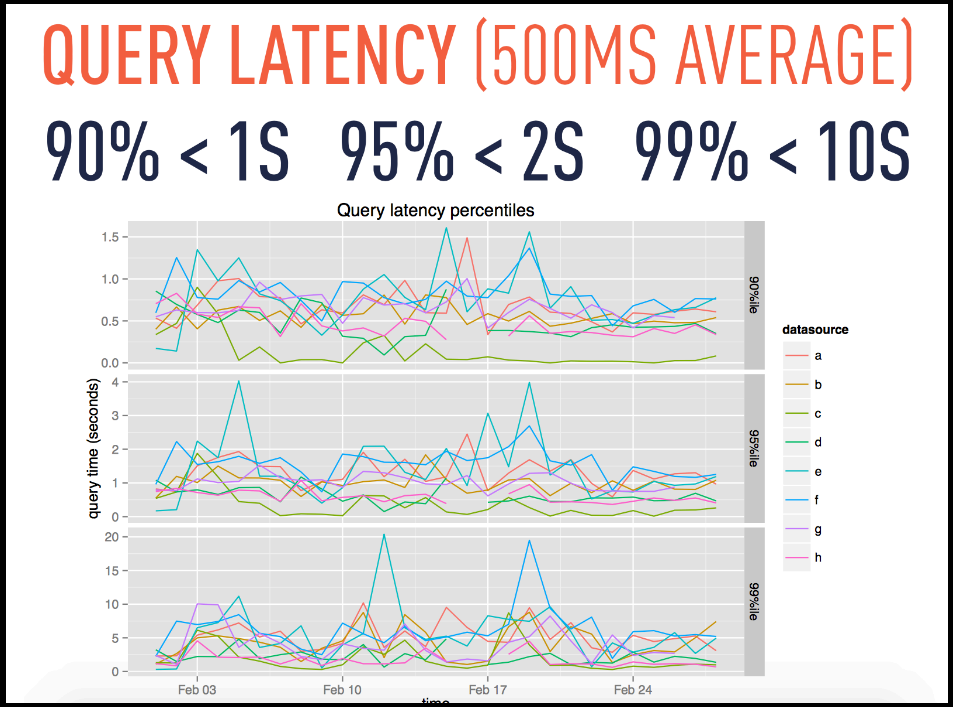
IS ROBOT	
false	777.3 k
true	464.9 k

IS NEW	
false	1,086.3 k
true	155.9 k

* Nishant Bangarwa: An Introduction to Druid (DataWorks Summit, 2018.5)

What is Druid ?

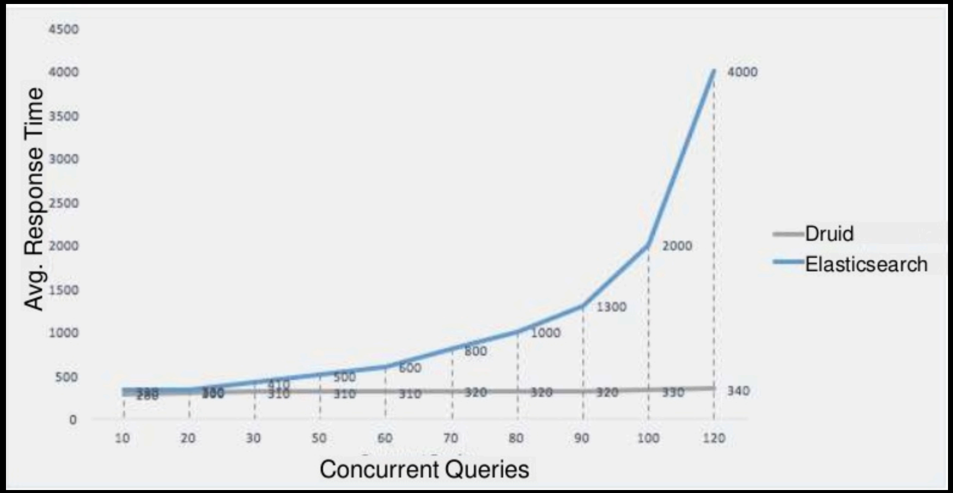
- Column-oriented distributed datastore
- Sub-Second query times
- Realtime streaming ingestion
- Arbitrary slicing and dicing of data
- Automatic Data Summarization
- Approximate algorithms (hyperLogLog, sketch)
- Scalable to petabytes of data
- Highly available
- Mo, Better concurrency



QUERY LATENCY (500MS AVERAGE)

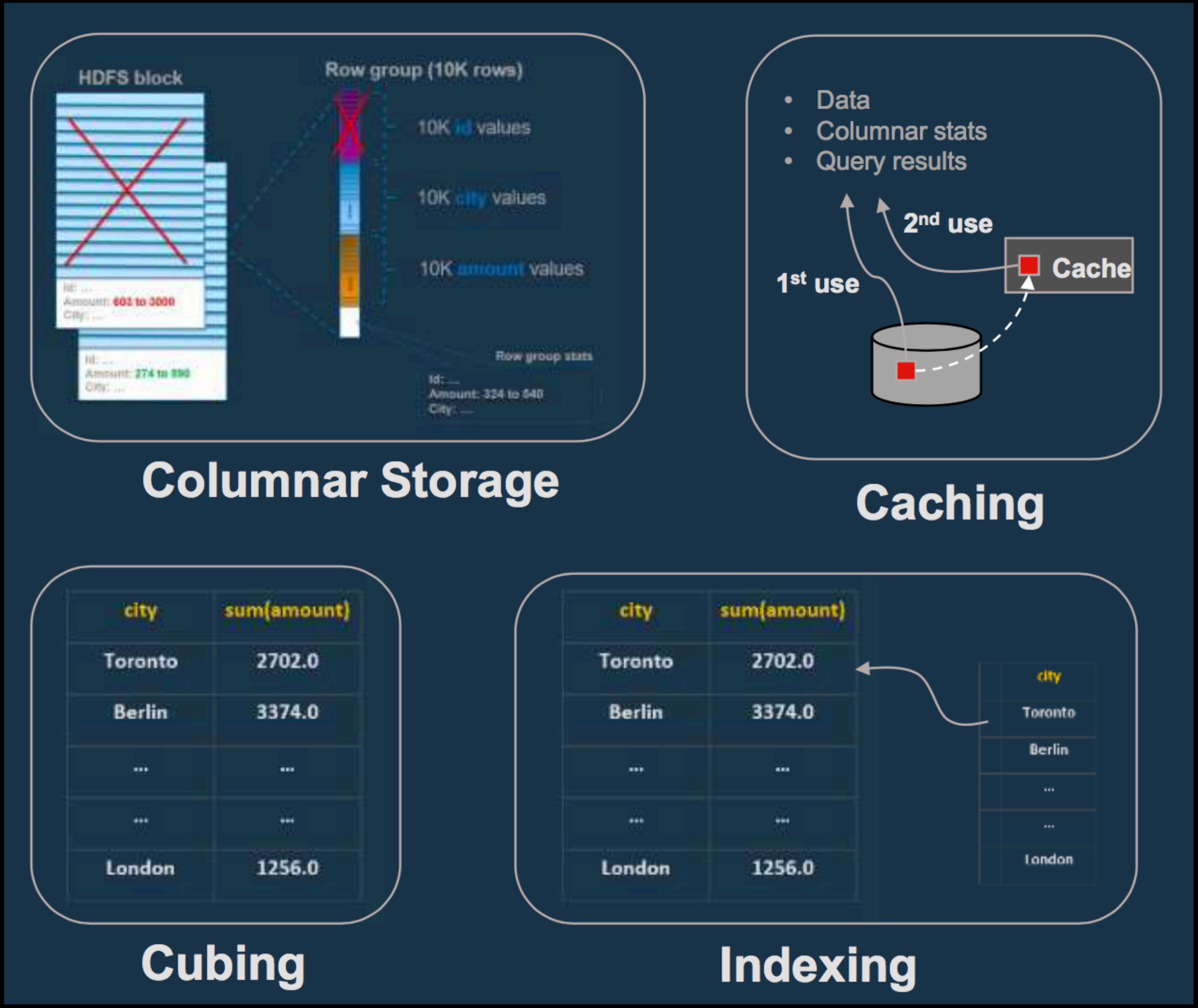
90% < 1S   95% < 2S   99% < 10S

* Nishant Bangarwa: Druid, sub second OLAP queries over petabytes of data



* Itai Yaffe, Our journey with druid - from initial research to full production scale

BI Acceleration Techniques



Columnar Storage

Caching

city	sum(amount)
Toronto	2702.0
Berlin	3374.0
...	...
...	...
London	1256.0

Cubing

city	sum(amount)
Toronto	2702.0
Berlin	3374.0
...	...
...	...
London	1256.0

city
Toronto
Berlin
...
...
london

Indexing

Jethro

Multiple instances of single node SMP engine

Indexing
Cubing
Caching

- **Computes cubes "bottom up" on demand**
- **Creates inverted indexes for all columns**
- **Must re-ingest all the data**

AtScale

Not an engine

Cubing
Caching
Approximate answers (e.g. count distinct)

- **Imposes star schema on all data**
- **Automatic and manual cubes**
- **Uses another engine to execute queries**

Kylin

MOLAP engine, storing cube cells in HBase

Cubing
Cost-based Optimizer

- **Brute force cube building**
- **Routes query to Hive when not in cube**
- **Uses Spark to speed up cube building**

* Gustavo Arocena : The Convergence of Reporting and Interactive BI on Hadoop (DataWorks Summit, London, 2018.5)

Case of Druid

Columnar Storage	• Uses Columnar Format • Processing is not vectorized
Caching	• Cache per segment (broker or historical) • Local / Remote (supports various caches) • Cache per query (in progress) • Not intelligent (key-value pairs, inefficient)
Indexing	• Dictionary + inverted index, R-index • Dictionary is not shared & not compressed • No index for metric (needs full scan)
Cubing	• Via pre-aggregation • Loosing some data • No runtime cube generation

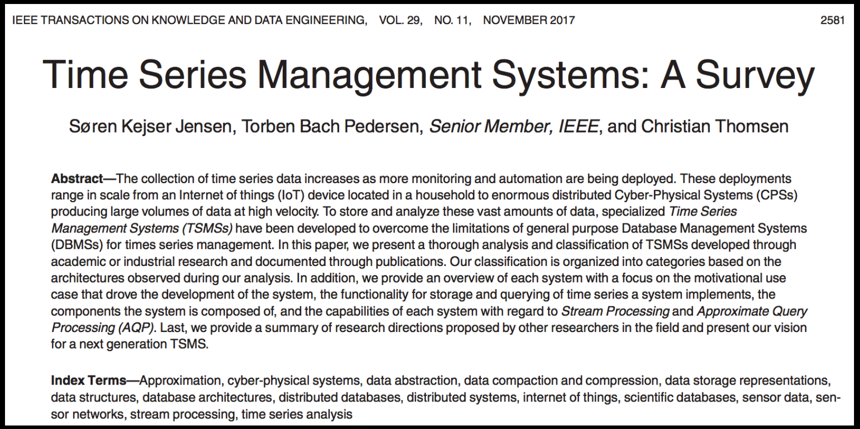
* Nishant Bangarwa: An Introduction to Druid (DataWorks Summit, 2018.5)

Functional Extension

- Plugin Based Architecture

  Leverage Guice in order to load extensions at runtime

  There are many engines faster than Druid, but it’s hard to see extensible one



IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 29, NO. 11, NOVEMBER 2017 2581

Time Series Management Systems: A Survey

Søren Kejser Jensen, Torben Bach Pedersen, *Senior Member, IEEE*, and Christian Thomsen

Abstract—The collection of time series data increases as more monitoring and automation are being deployed. These deployments range in scale from an Internet of things (IoT) device located in a household to enormous distributed Cyber-Physical Systems (CPSs) producing large volumes of data at high velocity. To store and analyze these vast amounts of data, specialized *Time Series Management Systems (TSMSs)* have been developed to overcome the limitations of general purpose Database Management Systems (DBMSs) for times series management. In this paper, we present a thorough analysis and classification of TSMSs developed through academic or industrial research and documented through publications. Our classification is organized into categories based on the architectures observed during our analysis. In addition, we provide an overview of each system with a focus on the motivational use case that drove the development of the system, the functionality for storage and querying of time series a system implements, the components the system is composed of, and the capabilities of each system with regard to *Stream Processing* and *Approximate Query Processing (AQP)*. Last, we provide a summary of research directions proposed by other researchers in the field and present our vision for a next generation TSMS.

Index Terms—Approximation, cyber-physical systems, data abstraction, data compaction and compression, data storage representations, data structures, database architectures, distributed databases, distributed systems, internet of things, scientific databases, sensor data, sensor networks, stream processing, time series analysis

- Possible to add extension to

  Add a new deep storage implementation

  Add a new Firehose for Ingestion

  Add Aggregators

  Add Complex metrics meetup 2nd, (2016.09.06)

  Add new Query types meetup 3rd, (2016.12.26)

  Add new Jersey resources

- Bundle your extension with all the other Druid extensions

  druid-stats, druid-orc

* Nishant Bangarwa: An Introduction to Druid (DataWorks Summit, 2018.5)

UI tools (OSS)

- Superset

  Developed at AirBnb

  In Apache Incubation since May 2017

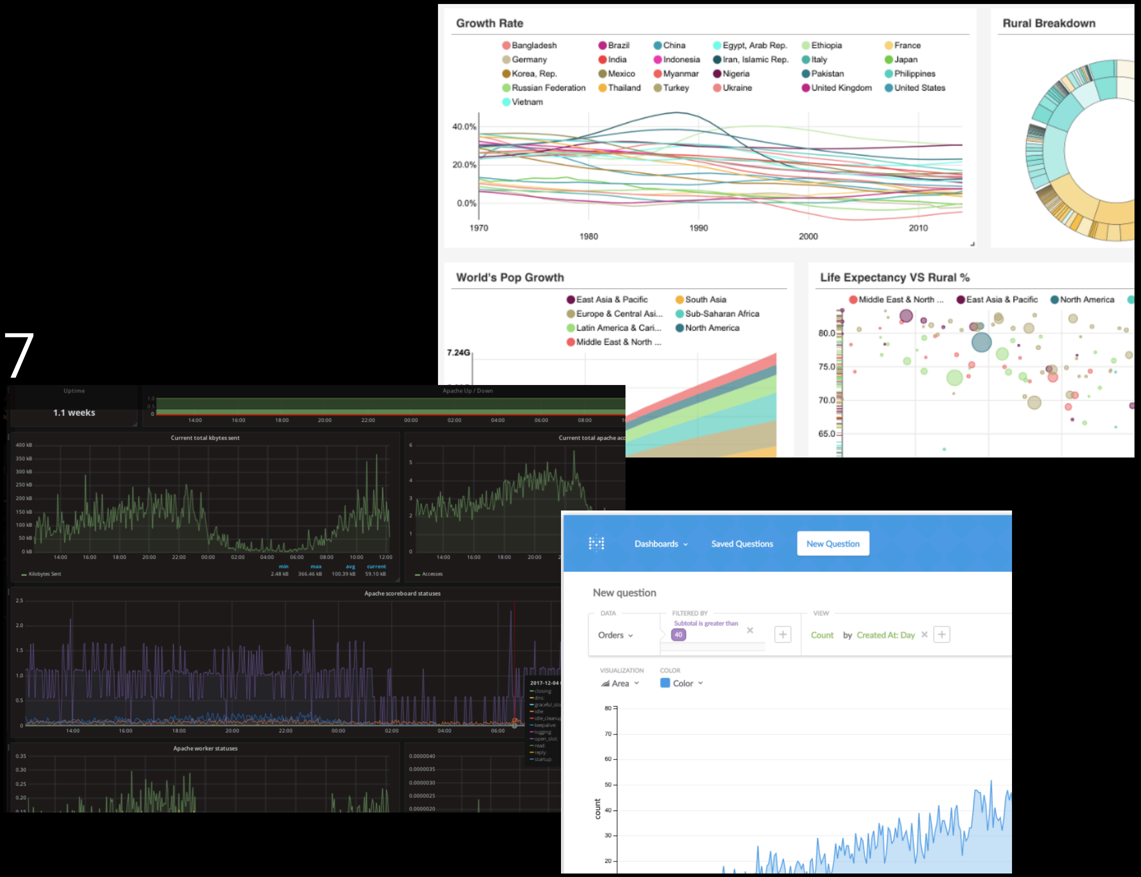
- Grafana - Druid plugin

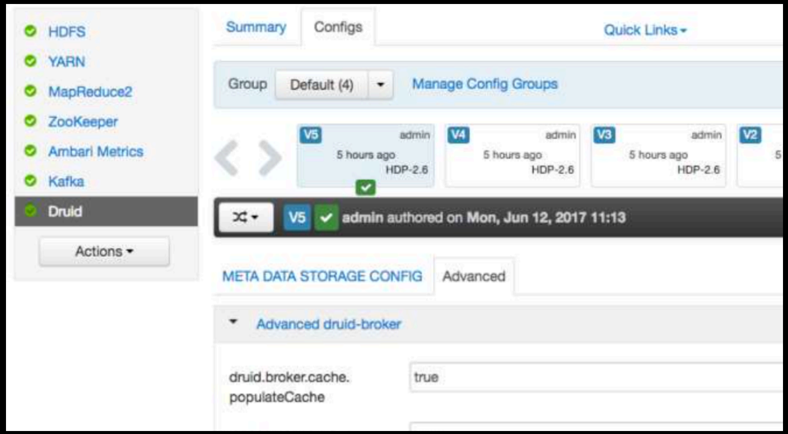
- Metabase

- With in-built SQL, connect with any BI tool supporting JDBC

- ~~Pivot~~

- Ambari (HDP) integrated





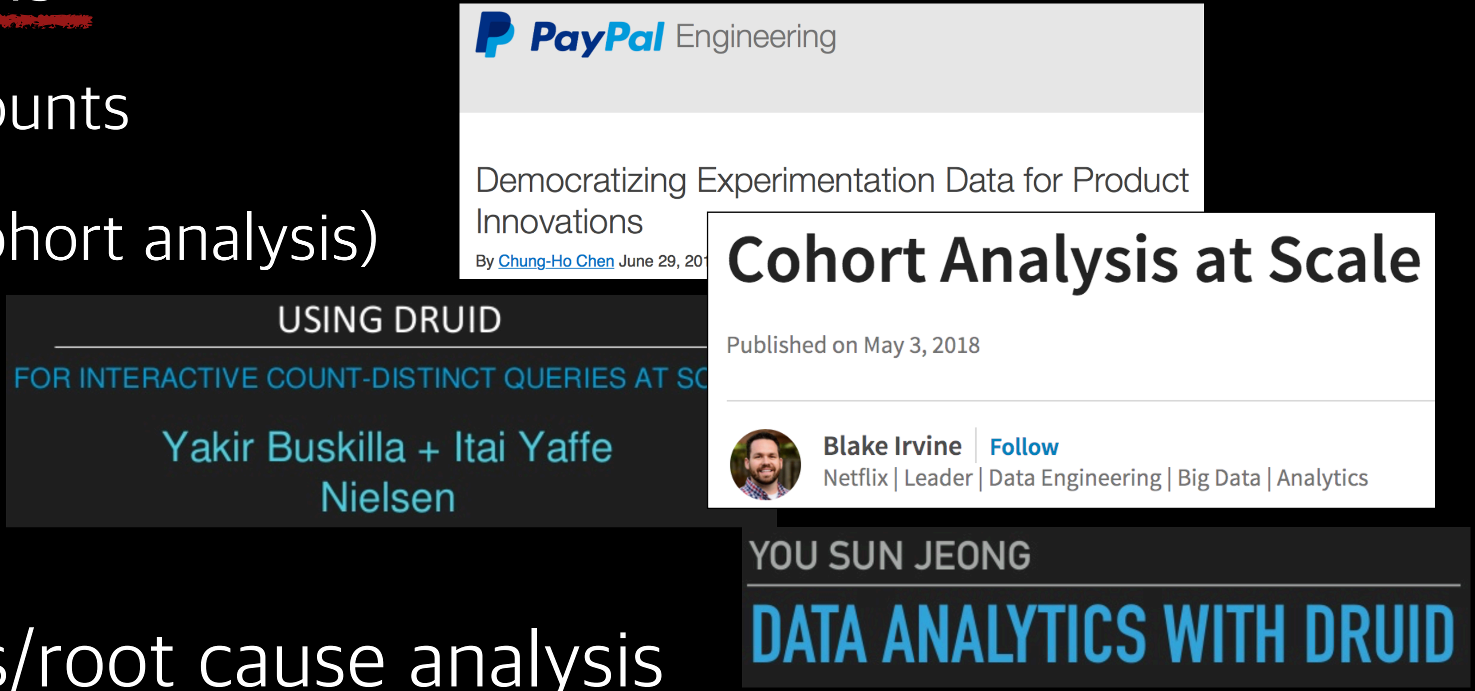
V5 admin authored on Mon, Jun 12, 2017 11:13

druid.broker.cache.populateCache: true

* Nishant Bangarwa: An Introduction to Druid (DataWorks Summit, 2018.5)

Suitable Use Cases

- Powering Interactive user facing applications
- Arbitrary slicing and dicing of large datasets
- User behavior analysis
  - measuring distinct counts
  - retention analysis (cohort analysis)
  - funnel analysis
  - A/B testing
- Exploratory analytics/root cause analysis
- Not interested in dumping entire dataset



PayPal Engineering

Democratizing Experimentation Data for Product Innovations

By Chung-Ho Chen June 29, 201

Cohort Analysis at Scale

Published on May 3, 2018

Blake Irvine Follow

Netflix | Leader | Data Engineering | Big Data | Analytics

USING DRUID

FOR INTERACTIVE COUNT-DISTINCT QUERIES AT SC

Yakir Buskilla + Itai Yaffe

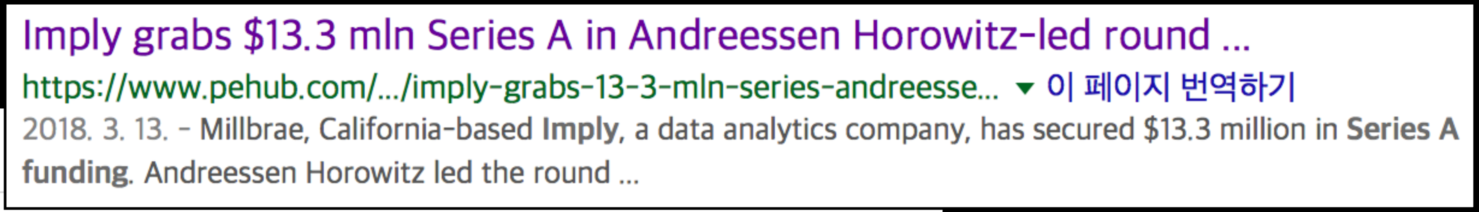
Nielsen

YOU SUN JEONG

DATA ANALYTICS WITH DRUID

Summary

- It’s good
- It’s promising



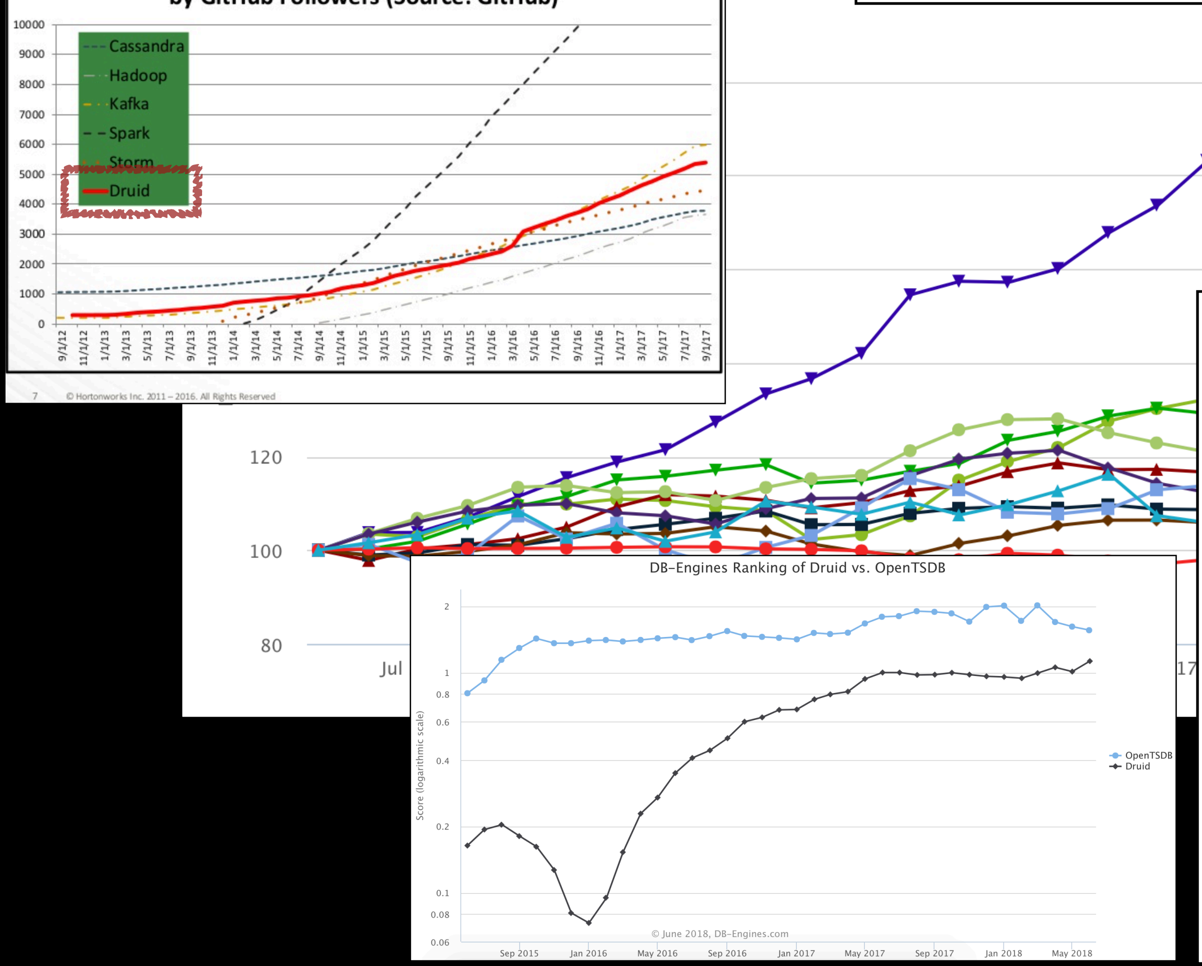
Imply grabs $13.3 mln Series A in Andreessen Horowitz-led round ...

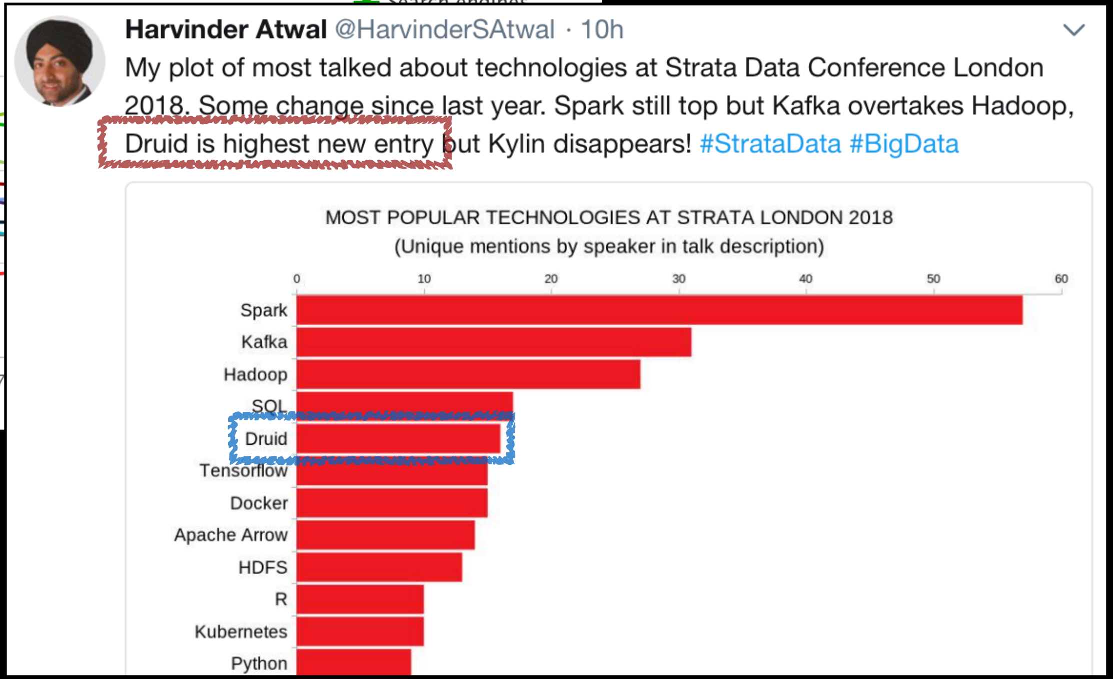
https://www.pehub.com/.../imply-grabs-13-3-mln-series-andreesse... ▾ 이 페이지 번역하기

2018. 3. 13. - Millbrae, California-based **Imply**, a data analytics company, has secured $13.3 million in **Series A funding**. Andreessen Horowitz led the round ...

Trend of the last 24 months

Popularity of Major Data Management Technologies by GitHub Followers (Source: GitHub)





Harvinder Atwal @HarvinderSAtwal · 10h

My plot of most talked about technologies at Strata Data Conference London 2018. Some change since last year. Spark still top but Kafka overtakes Hadoop, Druid is highest new entry but Kylin disappears! #StrataData #BigData

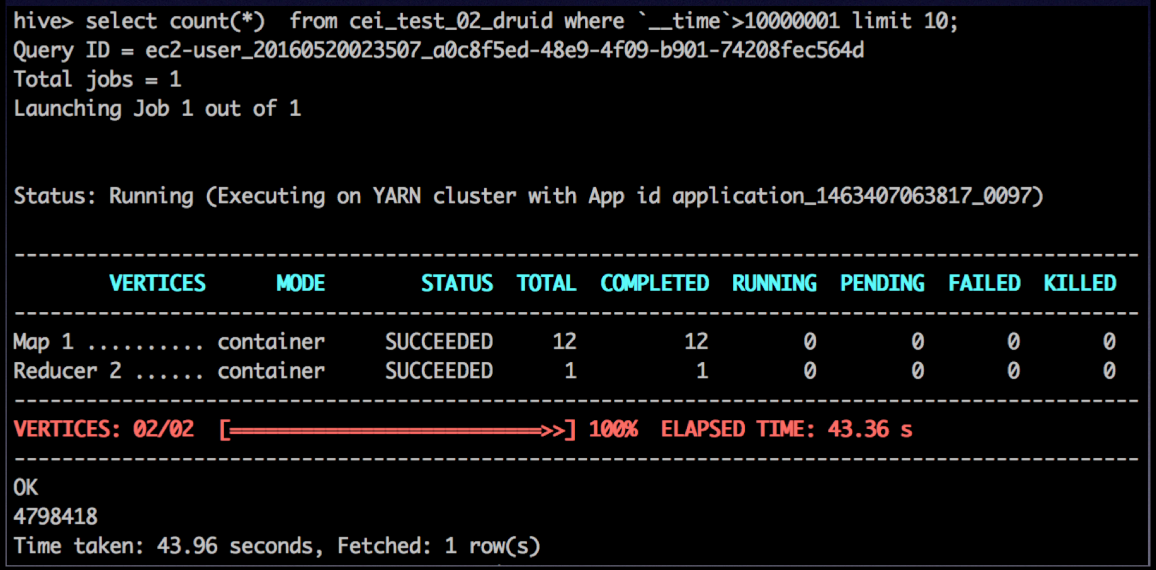
MOST POPULAR TECHNOLOGIES AT STRATA LONDON 2018
(Unique mentions by speaker in talk description)
0
10
20
30
40
50
60
Spark
Kafka
Hadoop
SQL
Druid
Tensorflow
Docker
Apache Arrow
HDFS
R
Kubernetes
Python

Follow-up works in SKT

Hive on Druid

- DruidStorageHandler

```
CREATE TABLE …
STORED BY "io.druid.hive.DruidHiveStorageHandler"
TBLPROPERTIES (
     "druid.broker.address"="http://polaris03:8082",
     "druid.datasource"="cei_test_02")
```



```
hive> select count(*)  from cei_test_02_druid where `__time`>10000001 limit 10;
Query ID = ec2-user_20160520023507_a0c8f5ed-48e9-4f09-b901-74208fec564d
Total jobs = 1
Launching Job 1 out of 1


Status: Running (Executing on YARN cluster with App id application_1463407063817_0097)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     12         12        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 43.36 s
----------------------------------------------------------------------------------------------
OK
4798418
Time taken: 43.96 seconds, Fetched: 1 row(s)
```

We do Druid

Druid meetup, 1st

2016.5.31, Navis (SK Telecom)

- Some improvements : BSON, StreamRawQuery, etc.
- Not using though : HortonWorks is elaborating it

Result Forwarding

- CSV, TSV

We do Druid

Druid meetup, 1st

2016.5.31, Navis (SK Telecom)

- Json, Excel, ORC, etc.
- Druid index
  - register as permanent or temporary data-source
- Parallel forwarding
  - select / stream query

Aggregation Functions

- variance, stddev

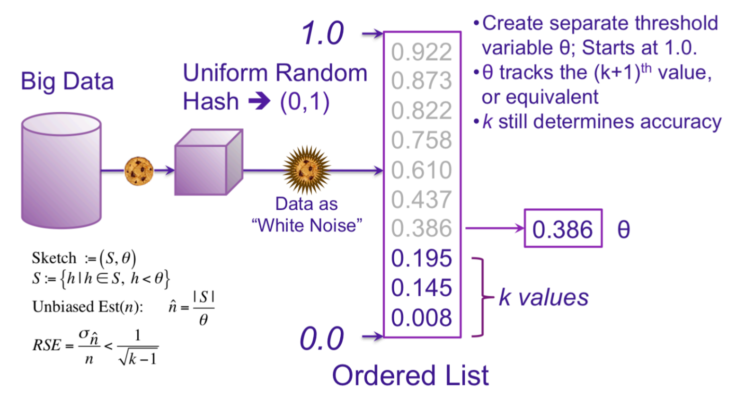
Aggregation Function

in Druid

Druid meetup, 2nd

2016.09.06, Navis (SKTelecom)

- range
- covariance, pearson, kurtosis
- timeMin, timeMax
- sketch (theta, quantile, sample, frequency)



- Create separate threshold variable θ; Starts at 1.0.
- θ tracks the (k+1)th value, or equivalent
- *k* still determines accuracy

$$\text{Sketch} := (S, \theta)$$

$$S := \{h \mid h \in S,\ h < \theta\}$$

$$\text{Unbiased Est}(n): \quad \hat{n} = \frac{|S|}{\theta}$$

$$RSE = \frac{\sigma_{\hat{n}}}{n} < \frac{1}{\sqrt{k-1}}$$

Queries

- SketchQuery

Druid 쿼리 확장

Druid meetup, 3rd (2016.12.26)

- Extended query function
  - GroupByQuery
    - GroupingSet (group#, cube, rollup)
    - Windowing (window functions, pivot, flatten, etc.)
    - LateralView
    - OutputColumns

Queries

- More query types
    - UnionAllQuery
        - Join, Summary, Covariance
    - IteratingQuery
        - FindNearest (k-means)
    - ManagementQuery
        - JMX, Config

Queries

- Query rewriting (Broker)
  - GroupByQuery : Timeseries (+ limit ordering pushdown)
  - CovarianceQuery : SelectMeta + Timeseries + CovariancePostProcessor
  - JoinQuery : UnionAll + JoinPostProcessor
  - KMeansQuery : SegmentMetadata (generate centroid) +
    FindNearest (IteratingQuery)
  - SummaryQuery : SelectMeta +
    UnionAll (Sketch.theta, Sketch.quantile) +
    Timeseries (metric) or Search (dimension) +
    SegmentMetadata (timestamp)

Queries

- Local optimization (historical)
  - Query splitting
    - Applicable to steaming queries : GroupByQuery, StreamRawQuery
    - Make histogram on a column, split and process one by one
    - Reduced (first) response time from historical nodes
    - Avoids OOM in historical nodes
  - Segment filtering
    - Remove unnecessary scan
    - SelectQuery

Druid Index

Extending Druid Index File

Druid meetup, 5th

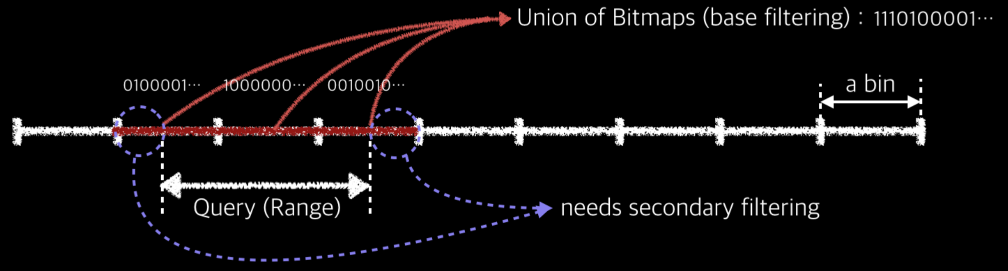
2017.10.19, Navis (SK Telecom)

- Ranged Histogram
- Lucene (text)
  - QueryFilter

- Bit sliced bitmap
- Lucene (spatial)
  - types : latlon, spatial (recursive prefix tree)
  - filters : Point, Spatial, GeoJsonPolygon, Nearest

Index (BSB)

- Why BSB?
  - Ranged histogram is hard to make (well) in single-phase
  - Easy to implement, low cost for building
  - Exact (Not like ranged histogram)


Union of Bitmaps (base filtering) : 1110100001···
0100001···
1000000···
0010010···
a bin
Query (Range)
needs secondary filtering

  - Only applicable for fixed-length types
    - cannot apply to string or BigDecimal
    - all primitive types (with some bit permutation)

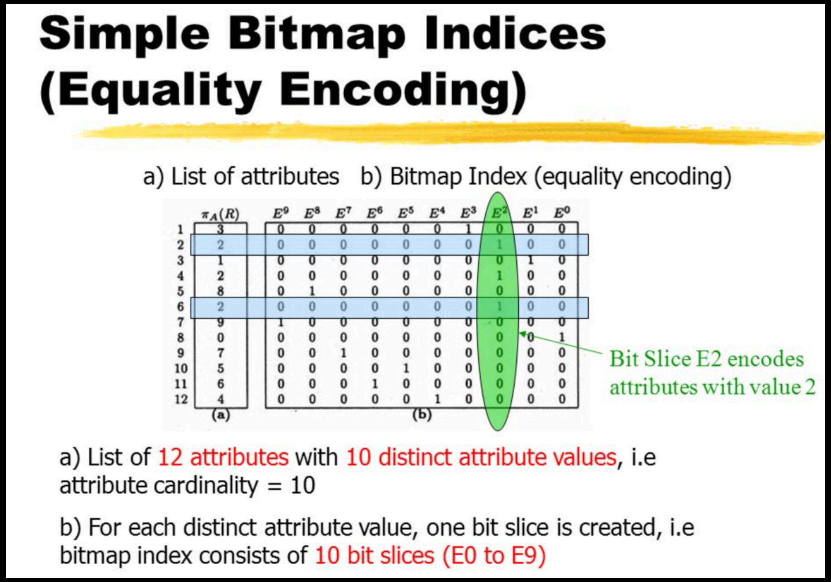
Index (BSB)

- What is BSB?
  - Example : {100, 135, 150, 200}, find x > 134

```
100  = 01100100
135  = 10000111
150  = 10010110
200  = 11000100

134  : 10000110

b8  : 0111  <— 1  (1 < 0 : fail = x)
b7  : x001  <— 0  (0 > 1 : ok = y)
b6  : x00y  <— 0  …
b5  : x01y  <— 0  …
b4  : x0yy  <— 0  …
b3  : x1yy  <— 1  …
b2  : x1yy  <— 1  …
b1  : x1yy  <— 0  …
      xyyy      result : 135, 150, 200
```



Simple Bitmap Indices (Equality Encoding)

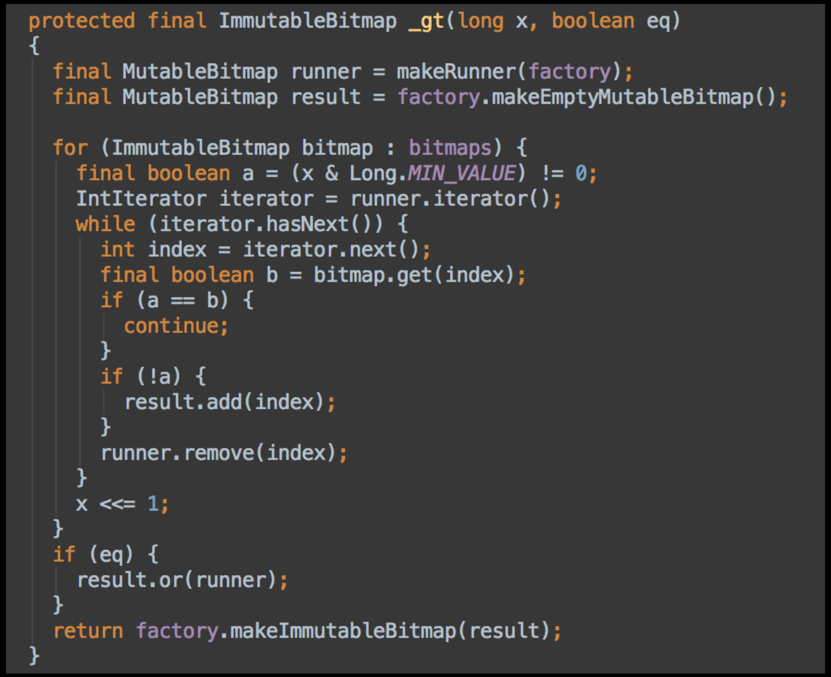
a) List of attributes   b) Bitmap Index (equality encoding)

	$\pi_A(R)$	E^9	E^8	E^7	E^6	E^5	E^4	E^3	E^2	E^1	E^0
1	3	0	0	0	0	0	0	1	0	0	0
2	2	0	0	0	0	0	0	0	1	0	0
3	1	0	0	0	0	0	0	0	0	1	0
4	2	0	0	0	0	0	0	0	1	0	0
5	8	0	1	0	0	0	0	0	0	0	0
6	2	0	0	0	0	0	0	0	1	0	0
7	9	1	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	1
9	7	0	0	1	0	0	0	0	0	0	0
10	5	0	0	0	0	1	0	0	0	0	0
11	6	0	0	0	1	0	0	0	0	0	0
12	4	0	0	0	0	0	1	0	0	0	0
	(a)	(b)									

Bit Slice E2 encodes attributes with value 2

a) List of 12 attributes with 10 distinct attribute values, i.e attribute cardinality = 10

b) For each distinct attribute value, one bit slice is created, i.e bitmap index consists of 10 bit slices (E0 to E9)



```java
protected final ImmutableBitmap _gt(long x, boolean eq)
{
  final MutableBitmap runner = makeRunner(factory);
  final MutableBitmap result = factory.makeEmptyMutableBitmap();

  for (ImmutableBitmap bitmap : bitmaps) {
    final boolean a = (x & Long.MIN_VALUE) != 0;
    IntIterator iterator = runner.iterator();
    while (iterator.hasNext()) {
      int index = iterator.next();
      final boolean b = bitmap.get(index);
      if (a == b) {
        continue;
      }
      if (!a) {
        result.add(index);
      }
      runner.remove(index);
    }
    x <<= 1;
  }
  if (eq) {
    result.or(runner);
  }
  return factory.makeImmutableBitmap(result);
}
```

Index (Lucene spatial)

- New requirements : GPS coordinates
    - T-map, Tango, etc.
- Druid supports r-index
    - Only supports euclidian coordinates
    - Inefficient in footprint (stored twice, in dimension & r-index)
- Improve r-index ?
    - Knows nothing on GIS : Am I doing it right?
    - Heard that ES supports it well

elastic{on}

Geospatial Data Structures
in Elasticsearch and Lucene

Nick Knize
@nknize

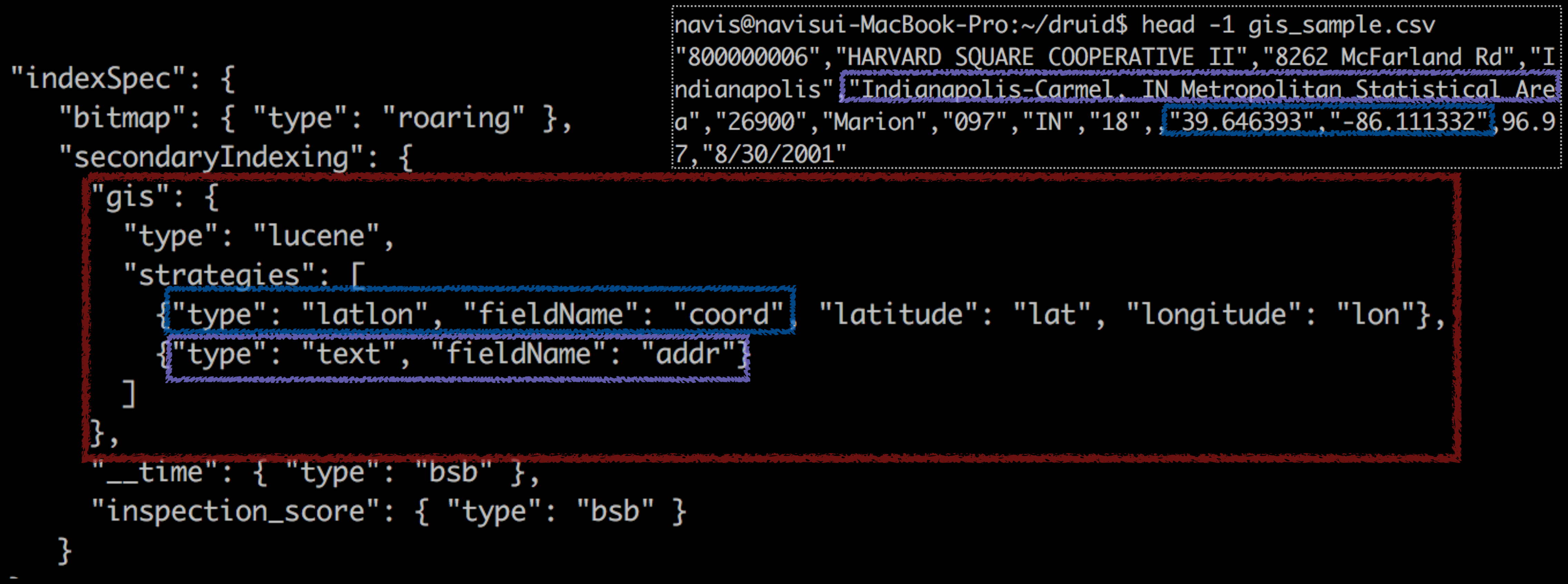
- Then, let’s store the coordinates as a column, index it with lucene

Index (Lucene spatial)

- Store coordinates to
    - dimension : string or string[] with index
    - metric : float, long (+ double, string, decimal), array
    - internal types : map, list, dateTime
- Use dimension? string? or array.double?
    - Inefficient or not intuitive
    - Cannot include other fields (to be indexed by lucene altogether)
    - Introduced “struct” type
        - example : struct (latitude:double, longitude:double, address:string)

Index (Lucene spatial)

- Index with Lucene
  - Extend indexSpec to accept lucene strategies
  - type : text + latlon, spatial

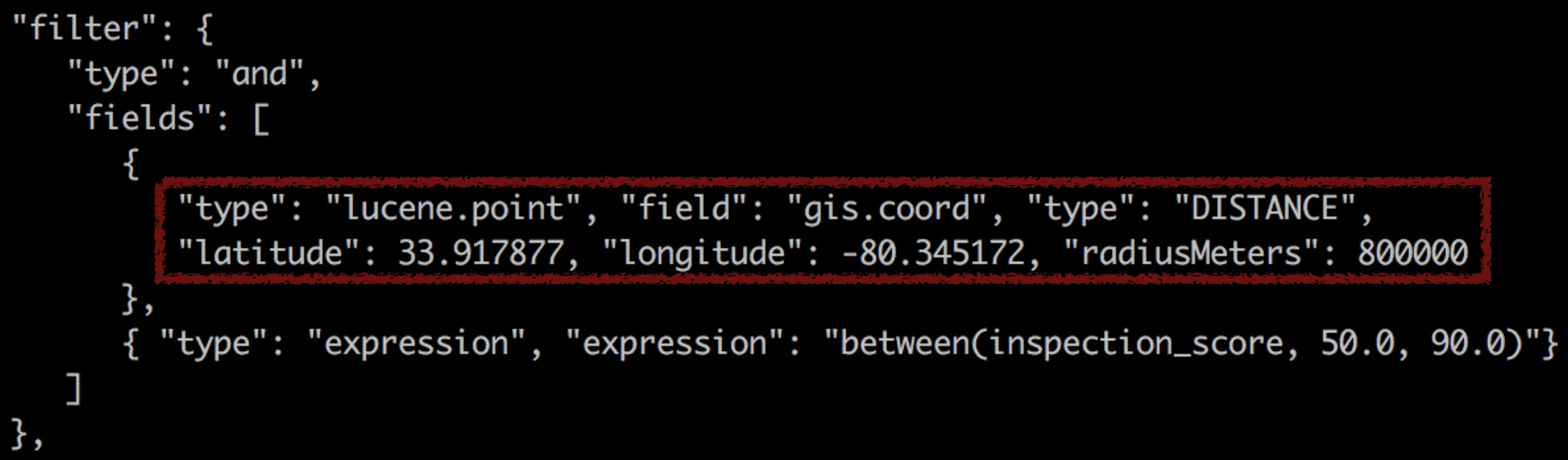


```
navis@navisui-MacBook-Pro:~/druid$ head -1 gis_sample.csv
"800000006","HARVARD SQUARE COOPERATIVE II","8262 McFarland Rd","I
ndianapolis","Indianapolis-Carmel, IN Metropolitan Statistical Are
a","26900","Marion","097","IN","18","39.646393","-86.111332",96.9
7,"8/30/2001"
```

```
"indexSpec": {
   "bitmap": { "type": "roaring" },
   "secondaryIndexing": {
     "gis": {
       "type": "lucene",
       "strategies": [
         {"type": "latlon", "fieldName": "coord", "latitude": "lat", "longitude": "lon"},
         {"type": "text", "fieldName": "addr"}
       ]
     },
     "__time": { "type": "bsb" },
     "inspection_score": { "type": "bsb" }
   }
```

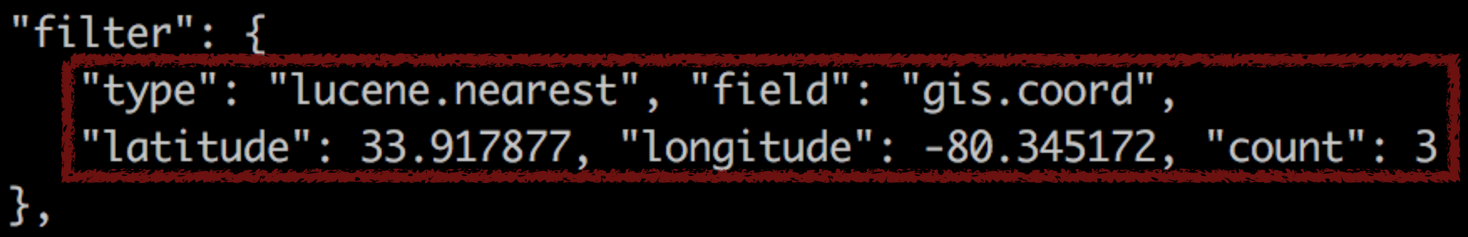
Index (Lucene spatial)

- Point filter
  - type : distance, box, polygon



```
"filter": {
   "type": "and",
   "fields": [
      {
         "type": "lucene.point", "field": "gis.coord", "type": "DISTANCE",
         "latitude": 33.917877, "longitude": -80.345172, "radiusMeters": 800000
      },
      { "type": "expression", "expression": "between(inspection_score, 50.0, 90.0)"}
   ]
},
```

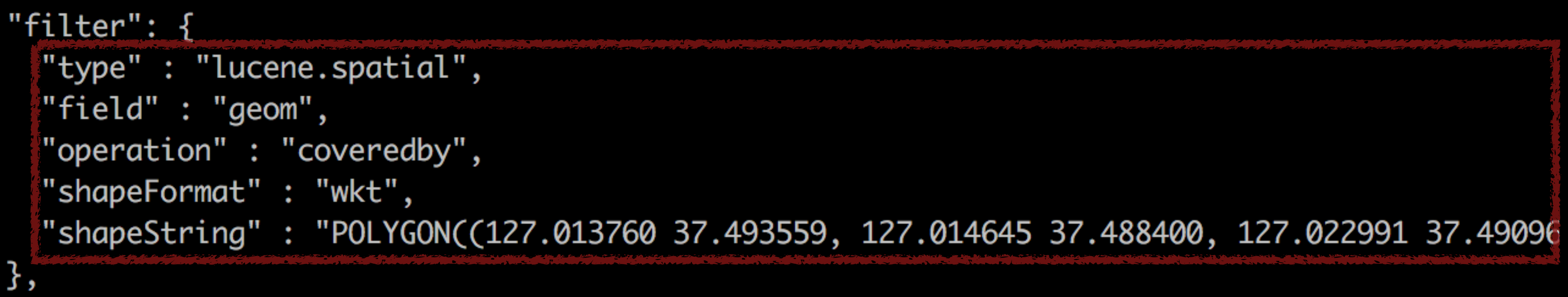
- Point nearest



```
"filter": {
   "type": "lucene.nearest", "field": "gis.coord",
   "latitude": 33.917877, "longitude": -80.345172, "count": 3
},
```

Index (Lucene spatial)

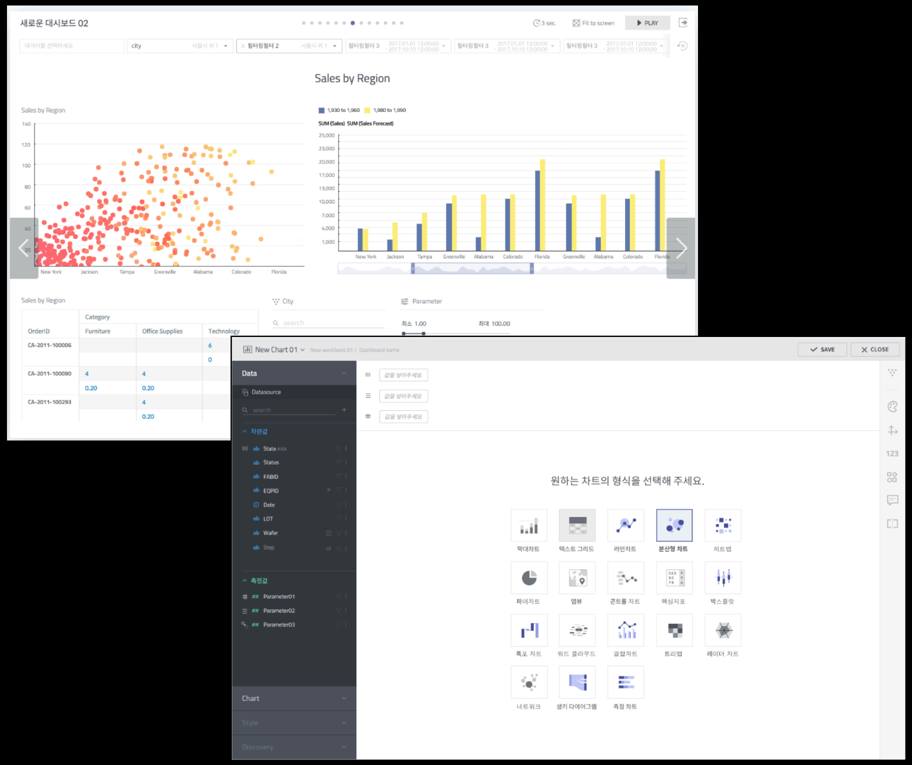
- Spatial filter
  - operation : covered, coveredBy, intersects
  - shapeFormat : WKT, GeoJson



```
"filter": {
  "type" : "lucene.spatial",
  "field" : "geom",
  "operation" : "coveredby",
  "shapeFormat" : "wkt",
  "shapeString" : "POLYGON((127.013760 37.493559, 127.014645 37.488400, 127.022991 37.49096
},
```

Summary

- We are taking Druid seriously
- Built Metatron on it
  - http://metatron.app
  - SKT, Hynix, IBK, Bharti Airtel, etc.
  - And will continue investigating on it
- So,



원하는 차트의 형식을 선택해 주세요.

we're
HIRING

Questions?