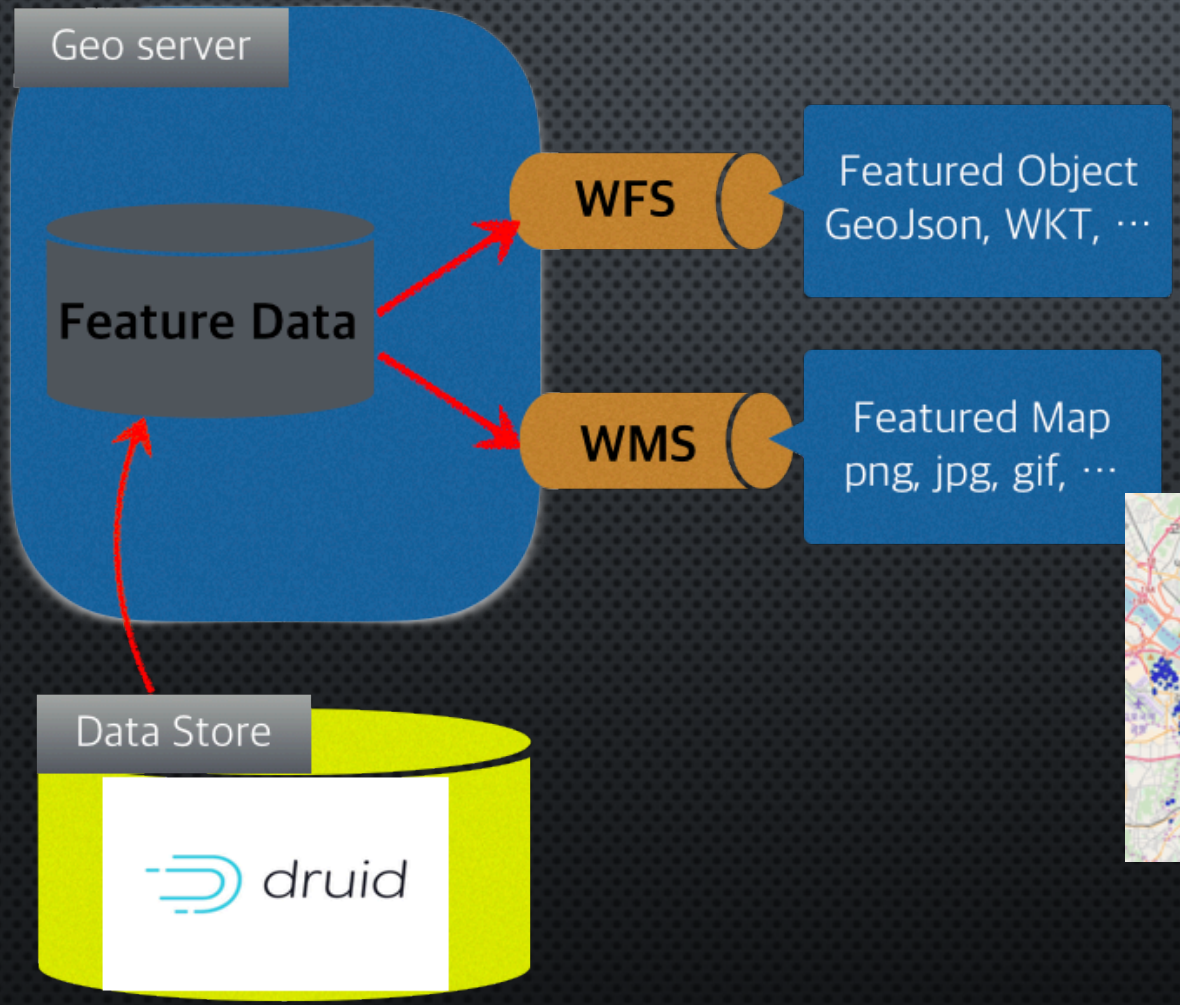
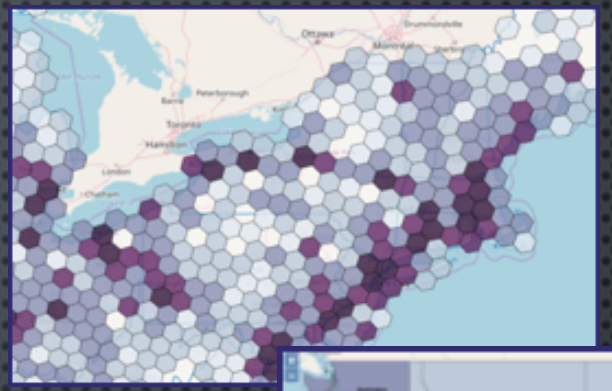
BUILDING GIS ON DRUID

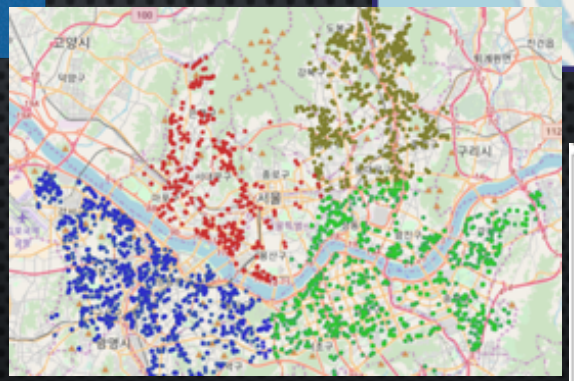
FOSS4G KOREA

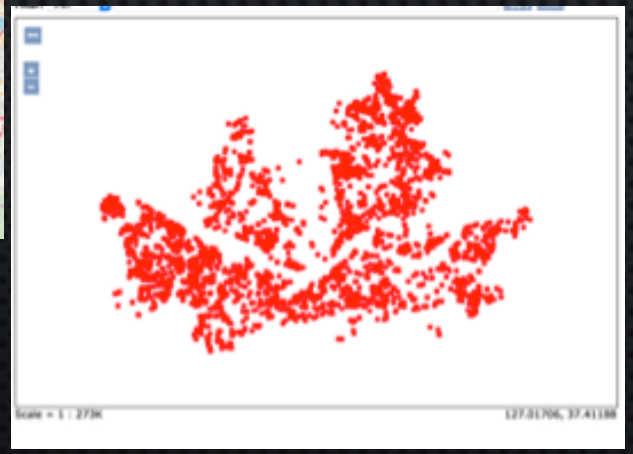
2018.10. 13. NAVIS(SK TELECOM) / LEE, JUNE WOO(SK TELECOM)

SYSTEM OVERVIEW







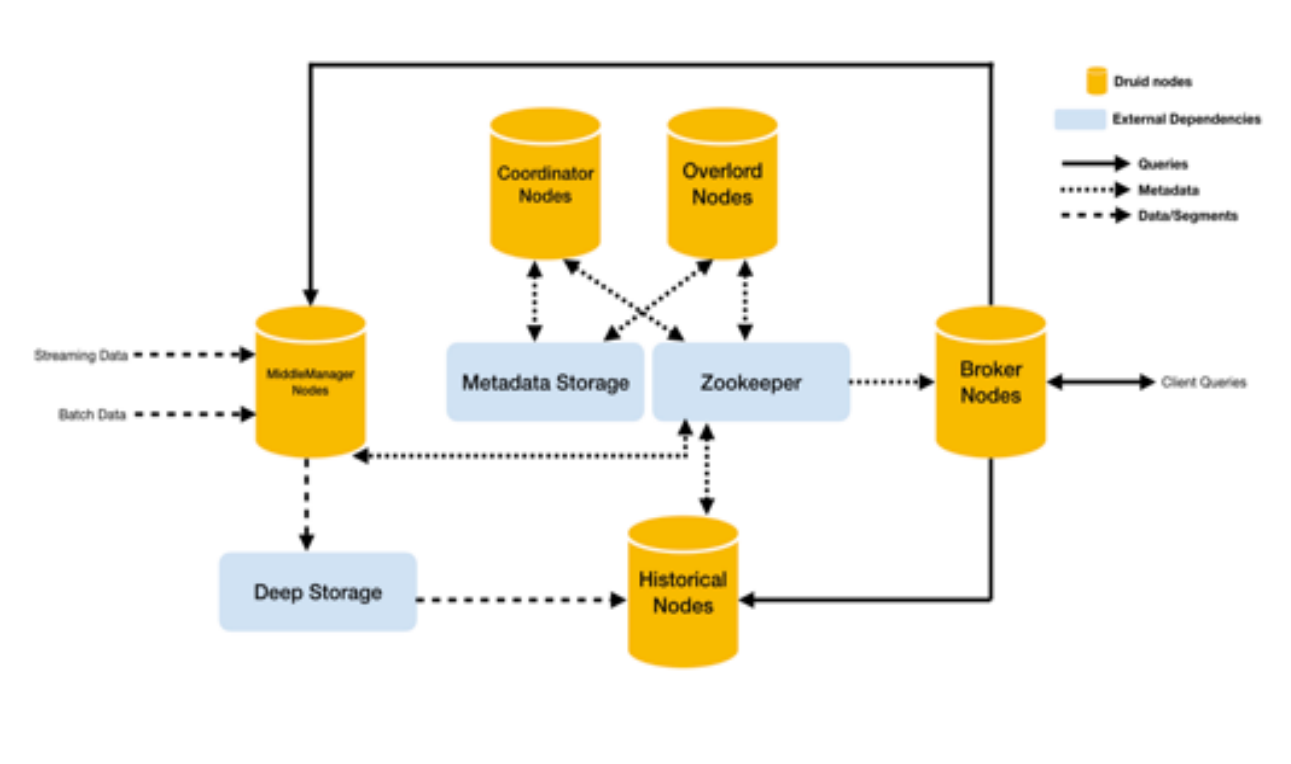


AGENDA

- What & Why Druid
- Druid in SKT
- Building GIS on Druid

WHAT IS DRUID?

- OPEN SOURCE DISTRIBUTED DATA STORE
- TIME-SERIES OLAP RUNNING JSON OR SQL QUERY
- LAMBDA ARCHITECTURE

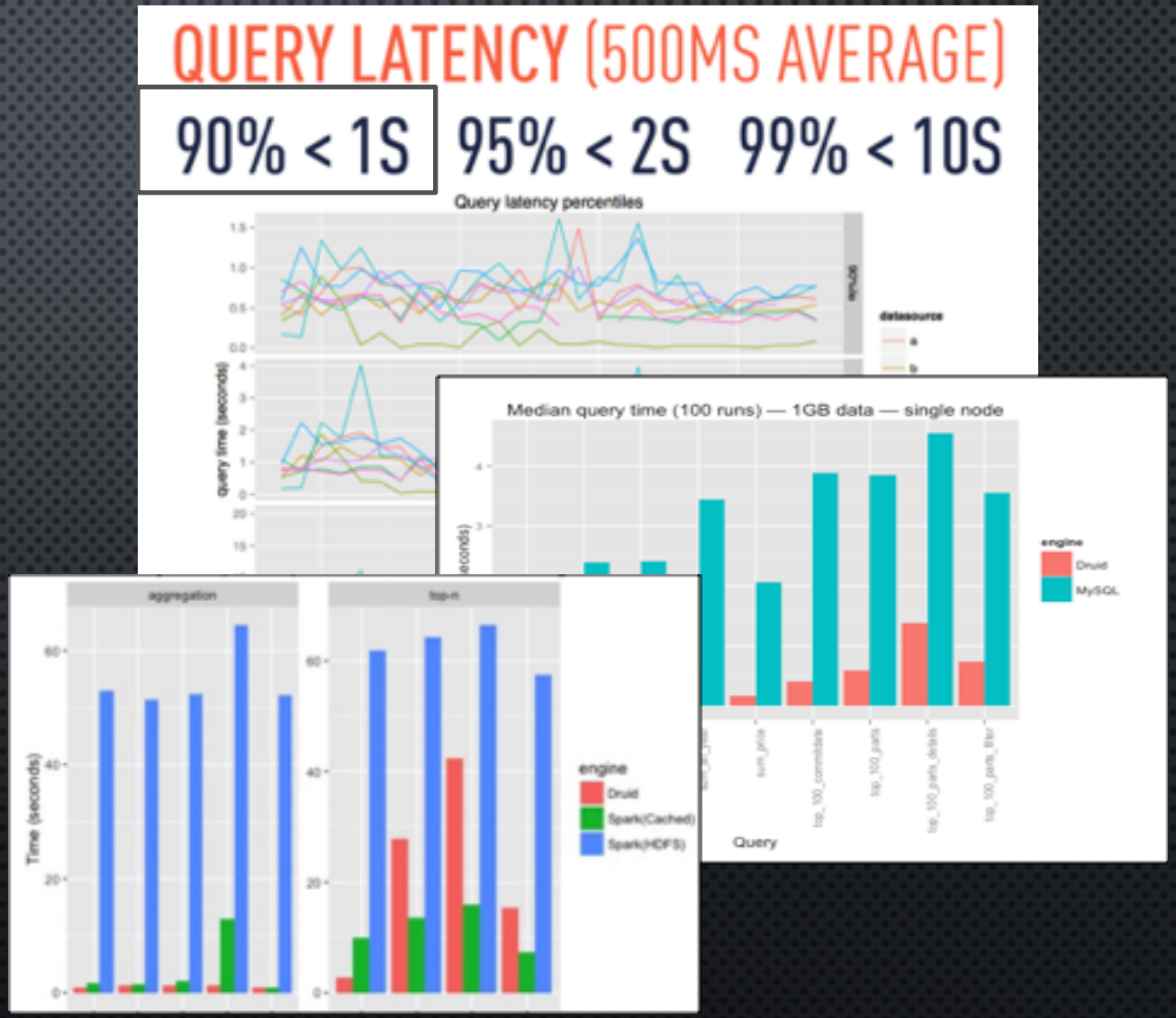


WHAT IS DRUID?

- SCALABLE TO PETA BYTES
- HIGH AVAILABLE
- APPROXIMATION ALGORITHMS
- FEW TO SUB-SECOND QUERY
- EXTENSIBLE

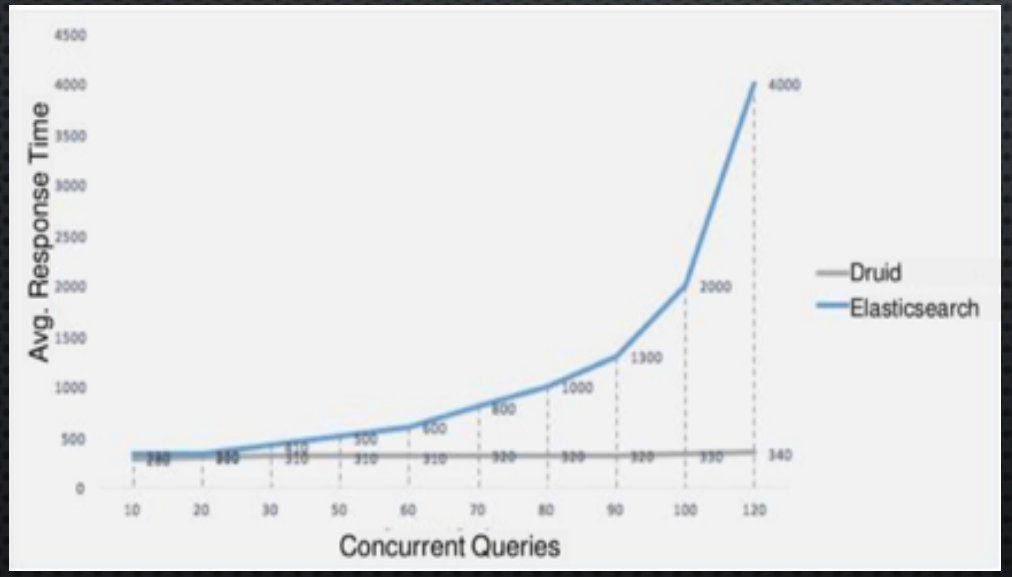
REFERENCE

- AIRBNB, ALIBABA, CISCO

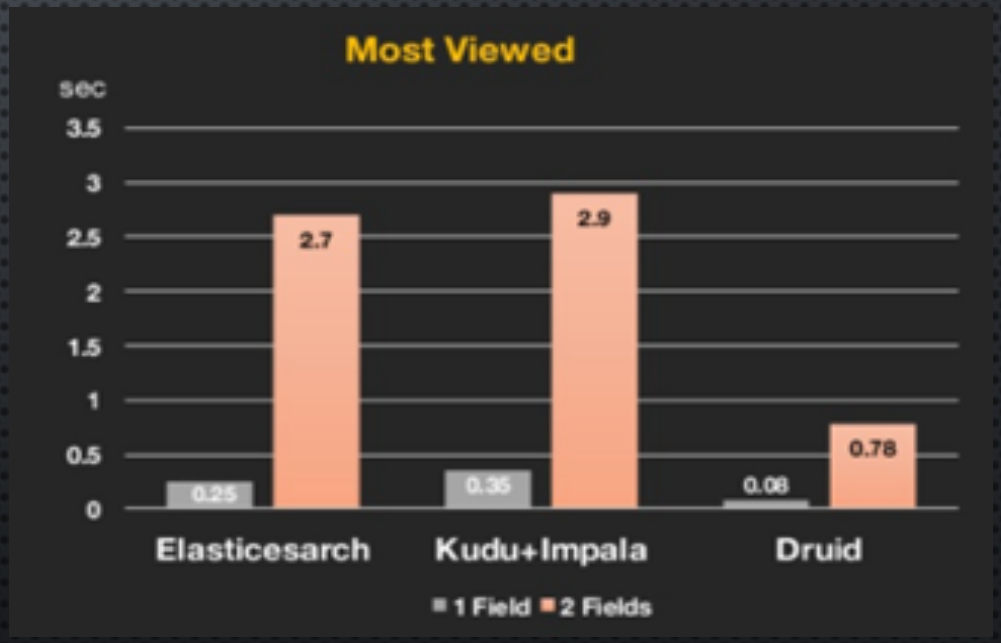


WHY WE DECIDED TO USE DRUID ?

- JAVA BASED & HIGHLY EXTENSIBLE
- LESS PAINFUL TO OPERATE THAN HBASE (+ PHOENIX OR KYLIN)
- FASTER AGGREGATION QUERY THEN ES
- BETTER CONCURRENCY WITH LESS RESOURCE THEN ES



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



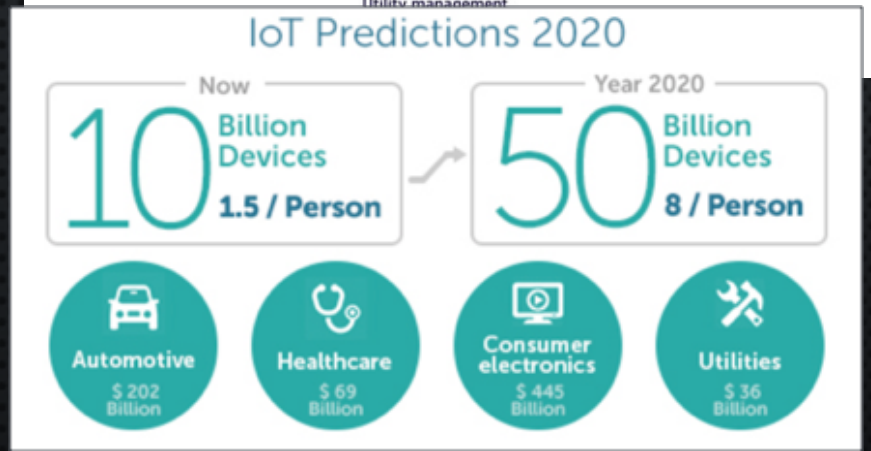
* Jason Heo & DooYong Kim, Web analytics at scale with Druid at naver.com

DRUID IN SKT

- **Tango** : Consolidated network management DW
- **T map** : Navigation service
- **MDM** : Company metadata management system
- **Jive** : De-identification marketing platform
- And also providing technical support to
  - Industrial Bank of Korea, SK Hynix, Bharti Airtel, …

NEW CHALLENGES

- 5G: 2019. 03 (SKT)
- 20B+ CONNECTED DEVICES TO 2020 (GARTNER, 2017)



CAN WE STORE/PROCESS THEM WITH DRUID?

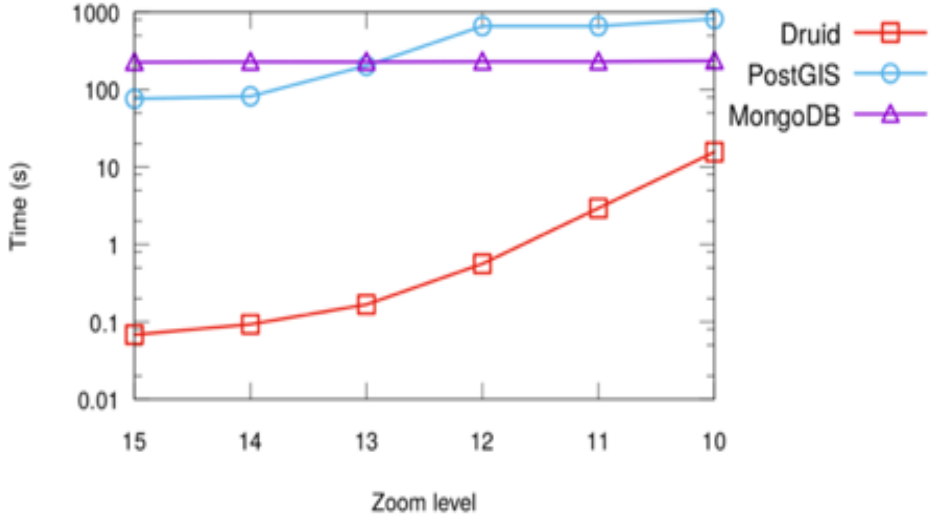
ICWE(International Conference on Web Engineering) 2018

A Case Study on Visualizing Large Spatial Datasets in a Web-based Map Viewer*

Alejandro Cortiñas[0000−0002−2555−6342], Miguel R. Luaces[0000−0003−0549−2000], and Tirso V. Rodeiro[0000−0003−2373−0746]

Universidade da Coruña
Laboratorio de Bases de Datos
A Coruña, Spain
{alejandro.cortinas, luaces, tirso.varela.rodeiro}@udc.es

Abstract. Lately, many companies are using Mobile Workforce Management technologies combined with information collected by sensors from mobile devices in order to improve their business processes. Even for small companies, the information that needs to be handled grows at



What we've done on Druid for spatial data

- Batch Ingestions
- Functions
- Queries
- Index extenstion – "Apache Lucene Index"

INDEX, EXTENSION

```
> Size of Index (except metadata) : 86,513,955 bytes
  Query Granularity : NoneGranularity
  Ingested NumRows : 756352 (rolled-up ? false)
  Aggregators
    RelayAggregatorFactory{name='L_COMMENT', columnName='L_COMMENT', typeName='string'}
    RelayAggregatorFactory{name='L_DISCOUNT', columnName='L_DISCOUNT', typeName='float'}
    RelayAggregatorFactory{name='L_EXTENDEDPRICE', columnName='L_EXTENDEDPRICE', typeName='double'}
    RelayAggregatorFactory{name='L_QUANTITY', columnName='L_QUANTITY', typeName='long'}
    RelayAggregatorFactory{name='L_TAX', columnName='L_TAX', typeName='float'}
  Bitmap Factory : RoaringBitmapFactory

> dimension '__time' (00000.smoosh, 0 ~ 311,084)
  type : LONG, numRows : 756352, hasMultiValue = false, (310,915 bytes, 0.4% of total), stats {min=694278000000
  metric bitmap (25 bitmaps, 0 zeros, 282,292 bytes)

> metric 'L_COMMENT' (00000.smoosh, 311,084 ~ 32,004,261)
  type : STRING, numRows : 756352, hasMultiValue = false, (31,692,999 bytes, 36.6% of total), stats {numNulls=0
  lucene index (5,594,760 bytes)

> metric 'L_DISCOUNT' (00000.smoosh, 32,004,261 ~ 33,818,203)
  type : FLOAT, numRows : 756352, hasMultiValue = false, (1,813,785 bytes, 2.1% of total), stats {min=0.0, numZ
  metric bitmap (10 bitmaps, 68,730 zeros, 1,065,072 bytes)

> metric 'L_EXTENDEDPRICE' (00000.smoosh, 33,818,203 ~ 38,983,835)
  type : DOUBLE, numRows : 756352, hasMultiValue = false, (5,165,470 bytes, 6.0% of total), stats {min=904.0, n
  metric bitmap (32 bitmaps, 0 zeros, 1,516,432 bytes)

> metric 'L_QUANTITY' (00000.smoosh, 38,983,835 ~ 41,732,322)
  type : LONG, numRows : 756352, hasMultiValue = false, (2,748,339 bytes, 3.2% of total), stats {min=1, numZero
  metric bitmap (32 bitmaps, 0 zeros, 1,516,432 bytes)
```

lucene index - Proximity matching

```
"filter": {
  "type": "and",
  "fields": [
    { "type": "regex",
      "dimension": "L_ORDERKEY",
      "pattern": "^[468].*"
    },
    { "type": "expression",
      "expression": "__time > datetime_millis('1992-09-10T22:25:00') && between(L_DISCOUNT, 0.08, 1.0)"
    },
    { "type": "lucene",
      "field": "L_COMMENT",
      "expression": "\"after special\" ~10"
    }
  ]
},
```

Can we store/process them with Druid?

we found Apache Lucene has

- Points
  - GeoPointField : deprecated
  - Geo3DPoint : not required, yet (next year?)
  - LatLonPoint : bbox, distance query
- Shapes
  - RecursivePrefixTree based on GeoHash
  - Supports spatial operations (covers, intersects, etc.)

CAN WE STORE/PROCESS THEM WITH DRUID? - QUERIES

- Point filter : bbox, distance

```
"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)"}
  ]
},
```

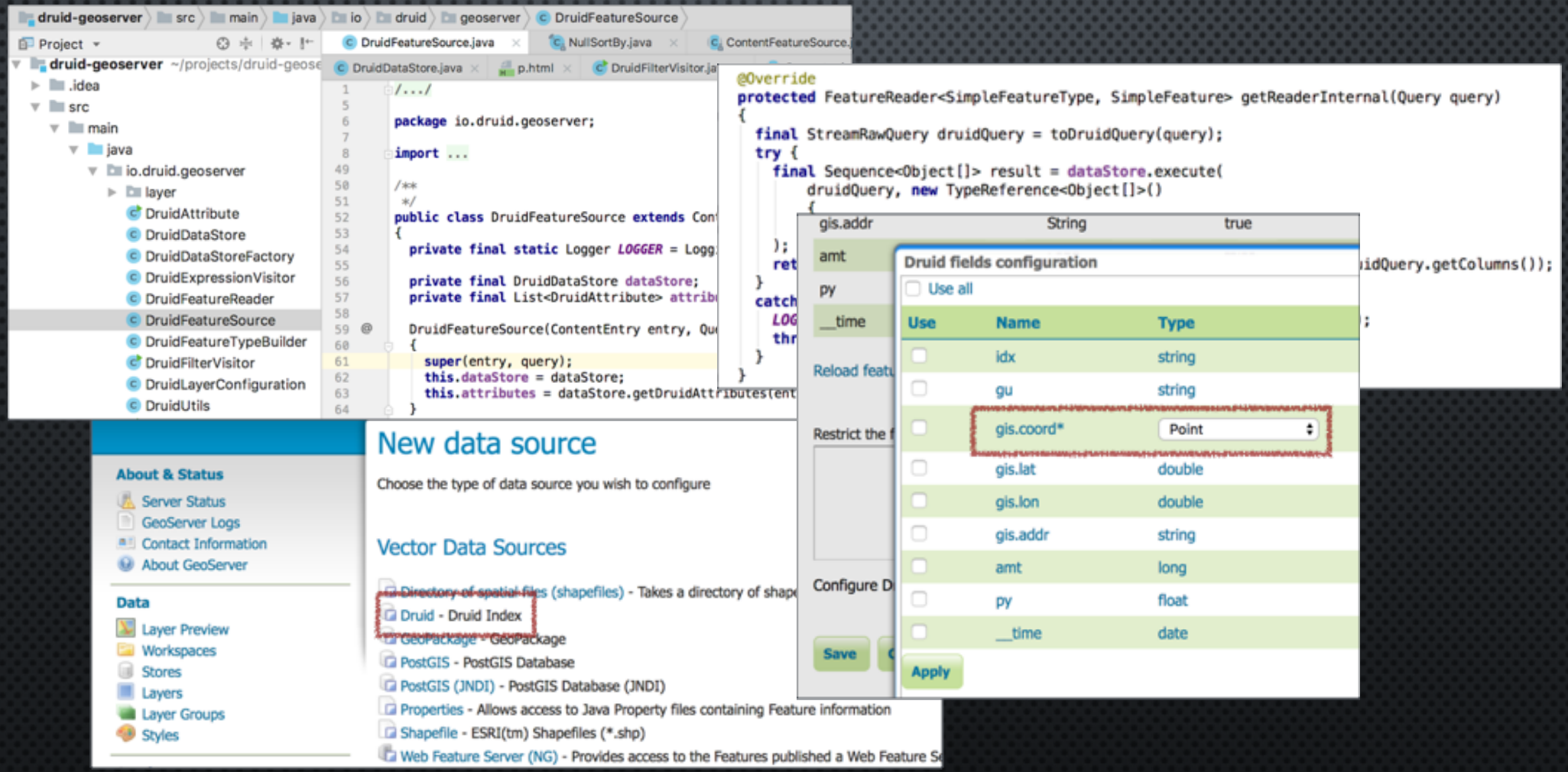
- Point nearest

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

- Shapes:
  - Operation: covered, intersects, …
  - Format: WKT, GeoJson

```
  "filter": {
    "type" : "lucene.spatial",
    "field" : "geom",
    "operation" : "coveredby",
    "shapeFormat" : "wkt",
    "shapeString" : "POLYGON((127.013760 37.493559, 127.014645 37.488400, 127
022991 37.490962, 127.021168 37.495816, 127.013760 37.493559))"
```

IMPLEMENTING DRUID DATASTORE FOR GEOSERVER

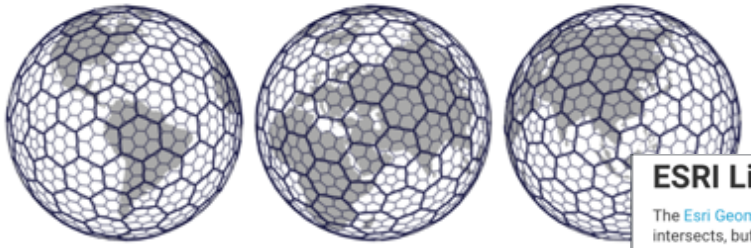


ETC

H3: Uber's Hexagonal Hierarchical Spatial Index

By Isaac Brodsky

June 27, 2018



Spatial4j

build passing | coverage 75% | maven-central v0.7

(note: Spatial4j's official home page is at LocationTech: https://projects.eclipse.org/projects/locationtech.spatial4j but this README has richer information)

Spatial4j is a general purpose spatial / geospatial ASL licensed open-source Java library. It's core capabilities are 3-foid: to provide common shapes that can work in Euclidean and geodesic (surface of sphere) world models, to provide distance calculations and other math, and to read & write shapes from formats like WKT and GeoJSON. Spatial4j is a project of the LocationTech Industry Working Group of the Eclipse Foundation.

If you are working wit
especially high utility

Spatial4j is well teste
for code coverage.

GeoTools

Documentation | D

which provides standards compliant methods for the
nent Geographic Information Systems (GIS). The GeoTools
specifications as they are developed.

ESRI Libraries

The Esri Geometry API for Java includes geometry objects (e.g. points, lines, and polygons), spatial operations (e.g. intersects, buffer), and spatial indexing. By deploying the library (as a jar) within Hadoop, you are able to build custom MapReduce applications using Java to complete analysis on your spatial data. This can be used as a standalone library, or combined with Spatial Framework for Hadoop to create a SQL like experience.

The Spatial Framework for Hadoop includes among others, the Hive Spatial library with User-Defined Functions and SerDes for spatial analysis in Hive. By enabling this library in Hive, you are able to construct queries using Hive Query Language (HQL), which is very similar to SQL. This allows you to avoid complicated MapReduce algorithms and stick to a more familiar workflow. The API used by the Hive UDF's could be used by developers building geometry functions for 3rd-party applications using Storm, Spark, HBase etc.

A good description of ESRI Java Geometry Library here: https://github.com/Esri/geometry-api-java/wiki

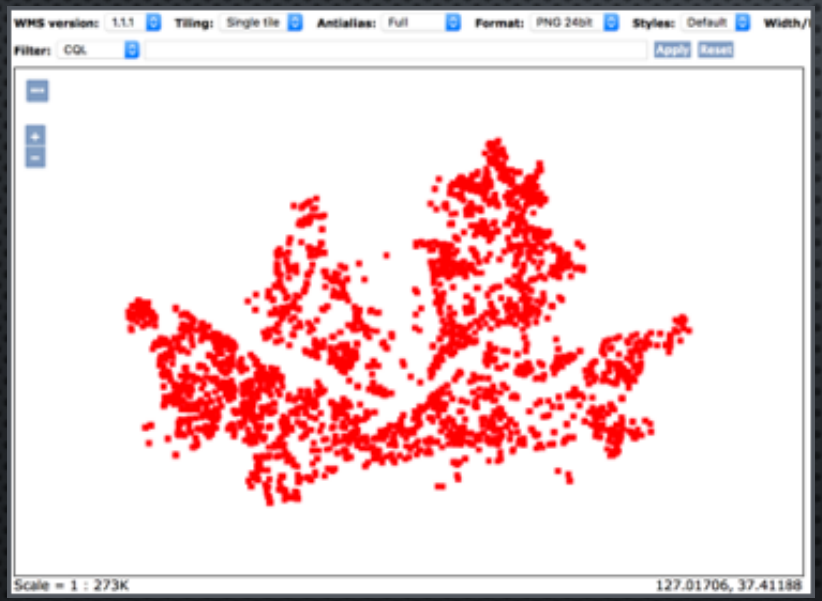
UDF Documentation: https://github.com/Esri/spatial-framework-for-hadoop/wiki/UDF-Documentation

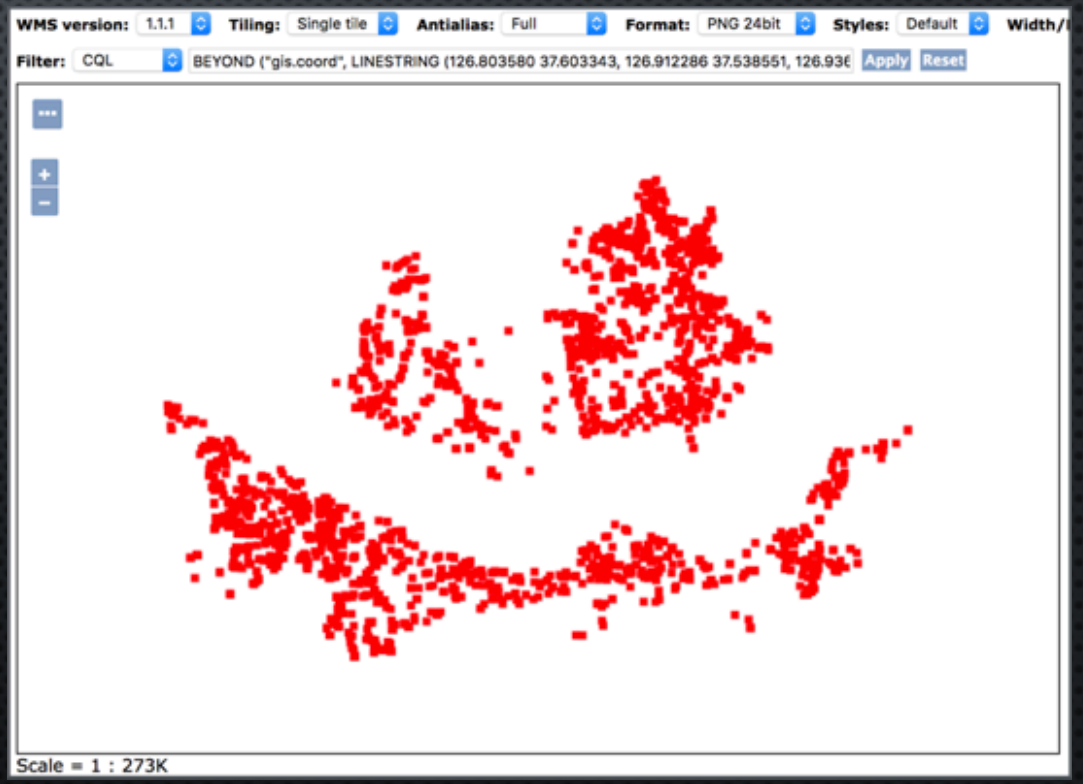
The following chart demonstrates the hierarchy of the ST_Geometry data type and its subclasses.

```
"virtualColumns": [
  {"expression": "haversin_meter(33.917877, -80.345172, gis.lat, gis.lon)", "outputName": "distance"},
  {"expression": "ST_GeodesicLengthWGS84(ST_SetSRID(ST_Linestring(gis.lon, gis.lat, -80.345172, 33.917877),4326))"
  {"expression": "to_geohash(gis.lat, gis.lon)", "outputName": "geohash"}
  {"expression": "to_h3(gis.lat, gis.lon, 10)", "outputName": "h3"}
],
```

WMS, CQL

BEYOND ("gis.coord", LINESTRING (126.803580 37.603343, 126.912286 37.538551, 126.936128 37.533877, 126.952270 37.520025, 126.981695 37.509739, 126.996454 37.514651, 127.025470 37.538023, 127.057066 37.530044, 127.077810 37.521918, 127.091702 37.523798, 127.104326 37.533463, 127.113816 37.544541, 127.124523 37.564985, 127.137214 37.571161, 127.149065 37.572075, 127.173098 37.583063), 3, kilometers)

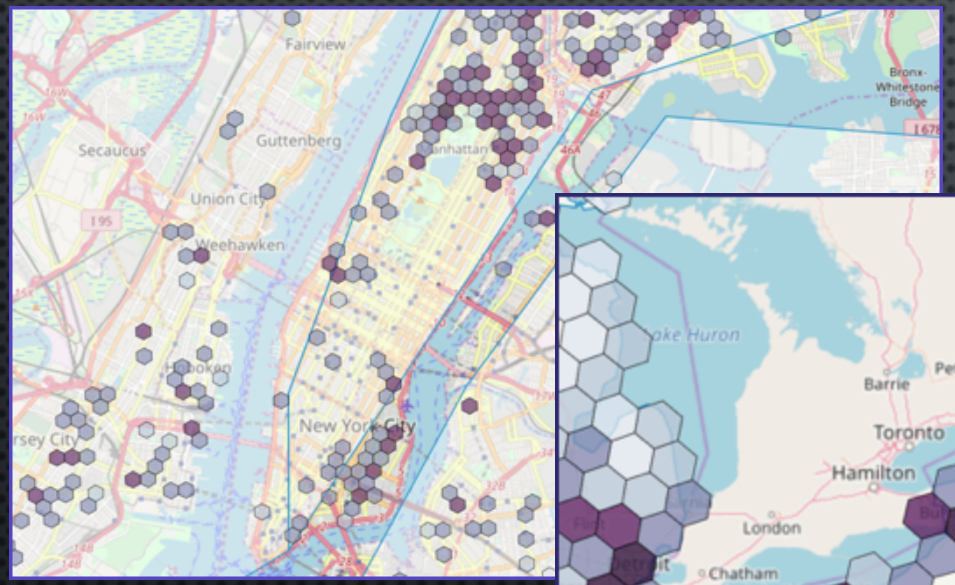




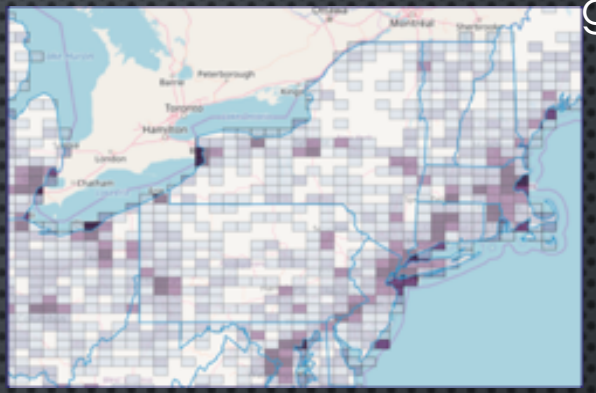
UBER H3 AND GEOHASH

SOURCE: 98783ROWS

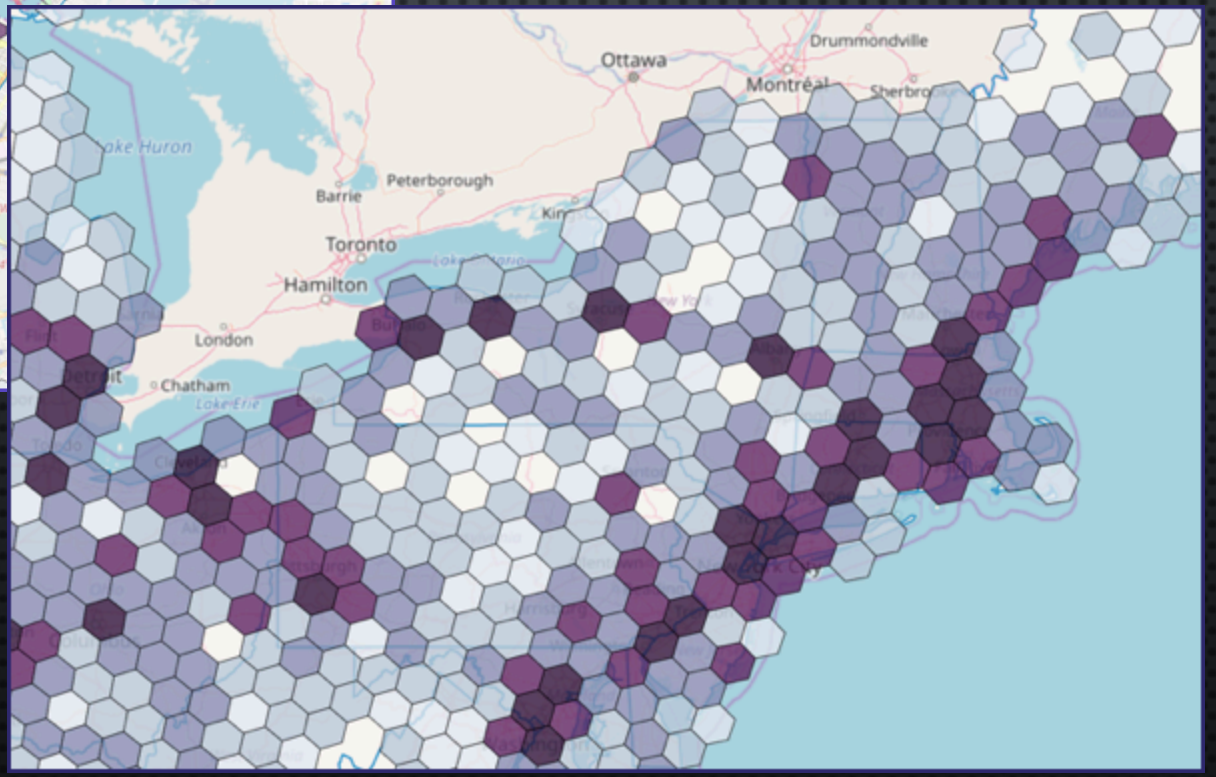
H3, precision=9, 23893 features



geohash



H3, precision=4, 2559 features



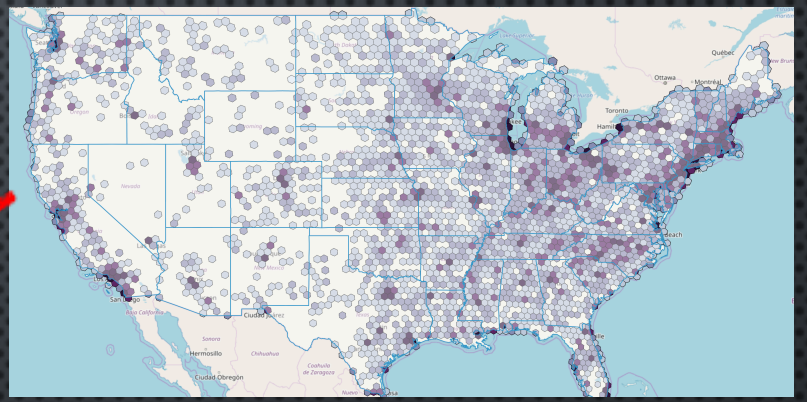
CHOROPLETH MAP (OR REGION MAP)

SOURCE: 98783ROWS

Boundary-join

Record - ((lat, lon), “California”, 91.81)

USA_State inspection_score at point

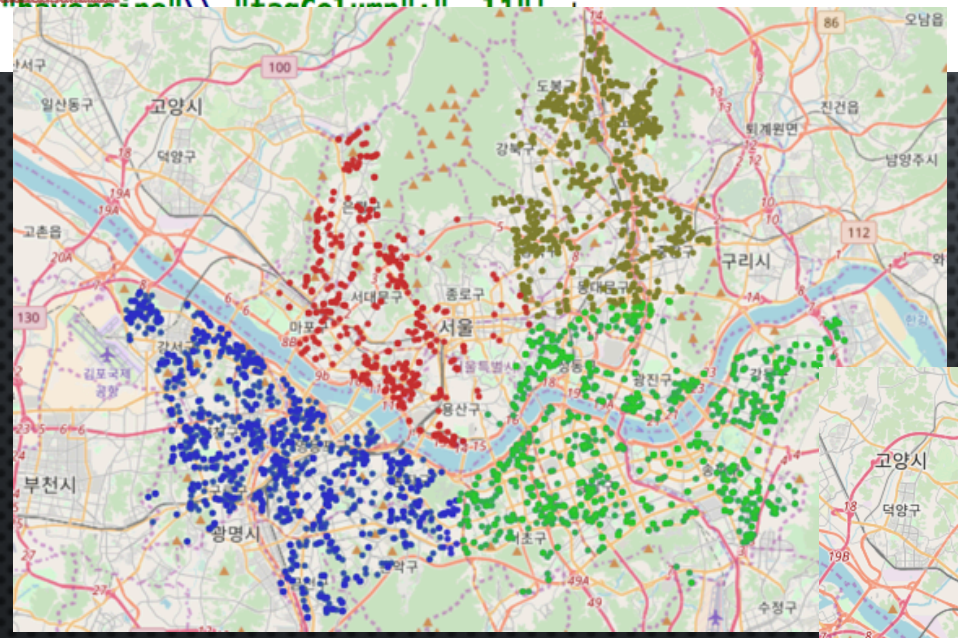


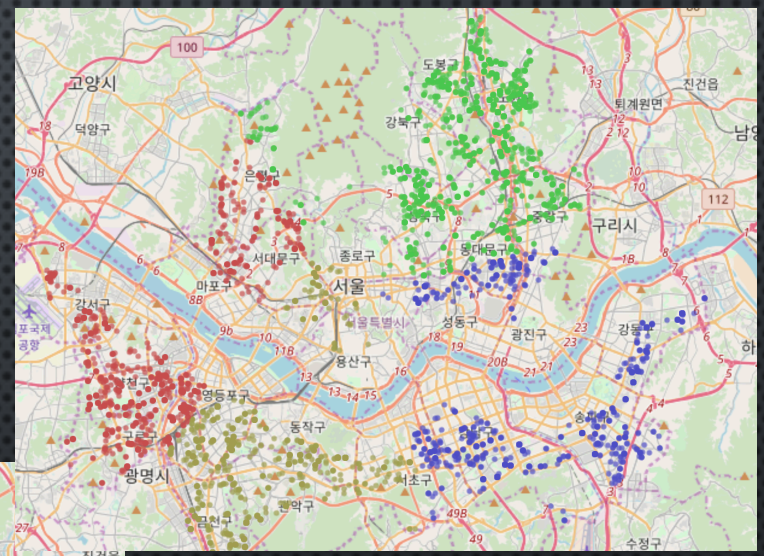
Record –(“California”, Polygon(lat lon))

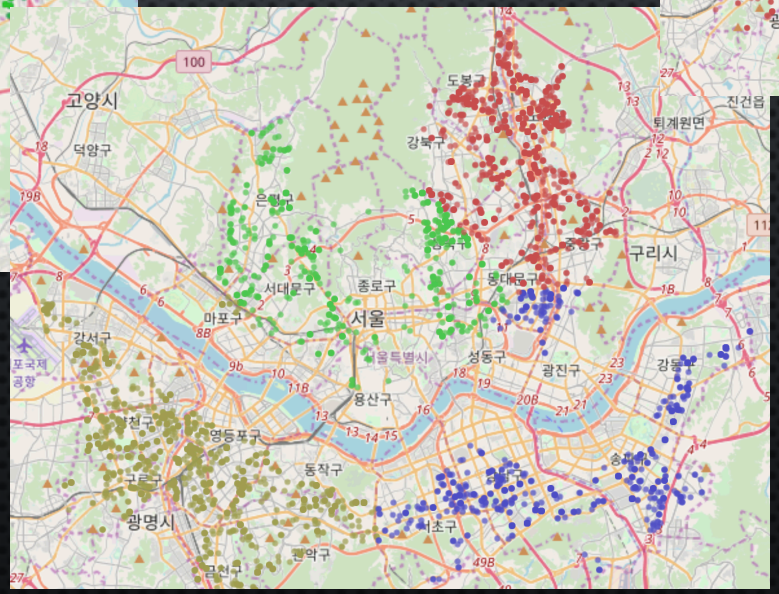
USA_State geometry information

K-MEANS ON COORDINATES

```
'viewParams=druid:{' +
  '"type":"kmeans"\\,"numK":4\\,"metrics":["gis.lon"\\,"gis.lat"]\\,' +
  '"measure":"
'}&' +
```







Conclusion

- Druid : Petabyte scalable, fast processing engine
- Lucene : Performant geo-indexing and geo-query
- By integrating two, we've got BIG GIS

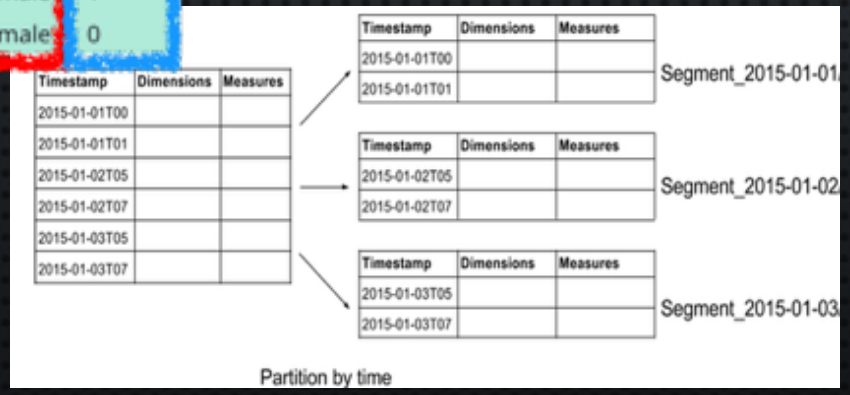
Thank you!

HOW DRUID STORE THE DATA

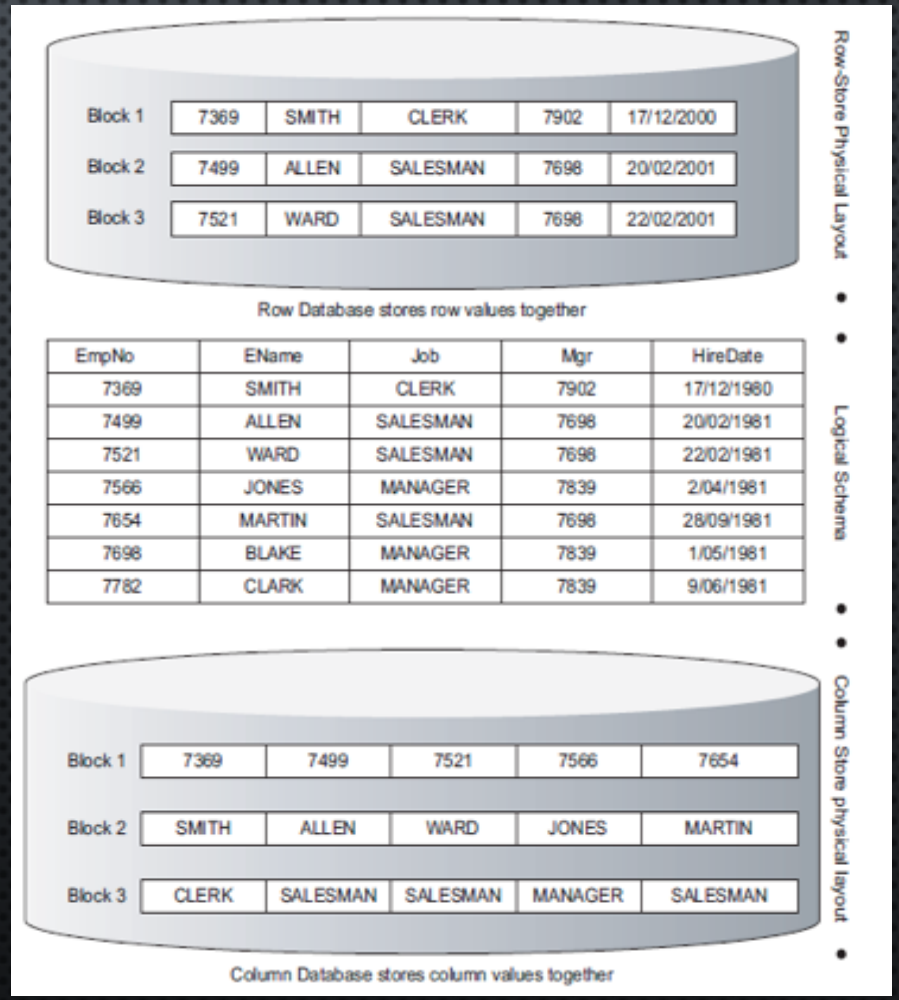
- COLUMN-ORIENTED DATA STORAGE
- PARTITION BY TIME

timestamp	domain	gender	clicked
2011-01-01T00:01:35Z	bieber.com	Female	1
2011-01-01T00:03:03Z	bieber.com	Female	0
2011-01-01T00:04:51Z	ultra.com	Male	1
2011-01-01T00:05:33Z	ultra.com	Male	1
2011-01-01T00:05:53Z	ultra.com	Female	0
2011-01-01T00:06:17Z	ultra.com	Female	1
2011-01-01T00:23:15Z	bieber.com	Female	0
2011-01-01T00:38:51Z	ultra.com	Male	1
2011-01-01T00:49:33Z	ultra.com	Female	1
2011-01-01T00:49:53Z	ultra.com	Female	0

timestamp	domain	gender	clicked
2011-01-01T00:00:00Z	bieber.com	Female	1
2011-01-01T00:00:00Z	ultra.com	Male	3
2011-01-01T00:00:00Z	ultra.com	Female	2

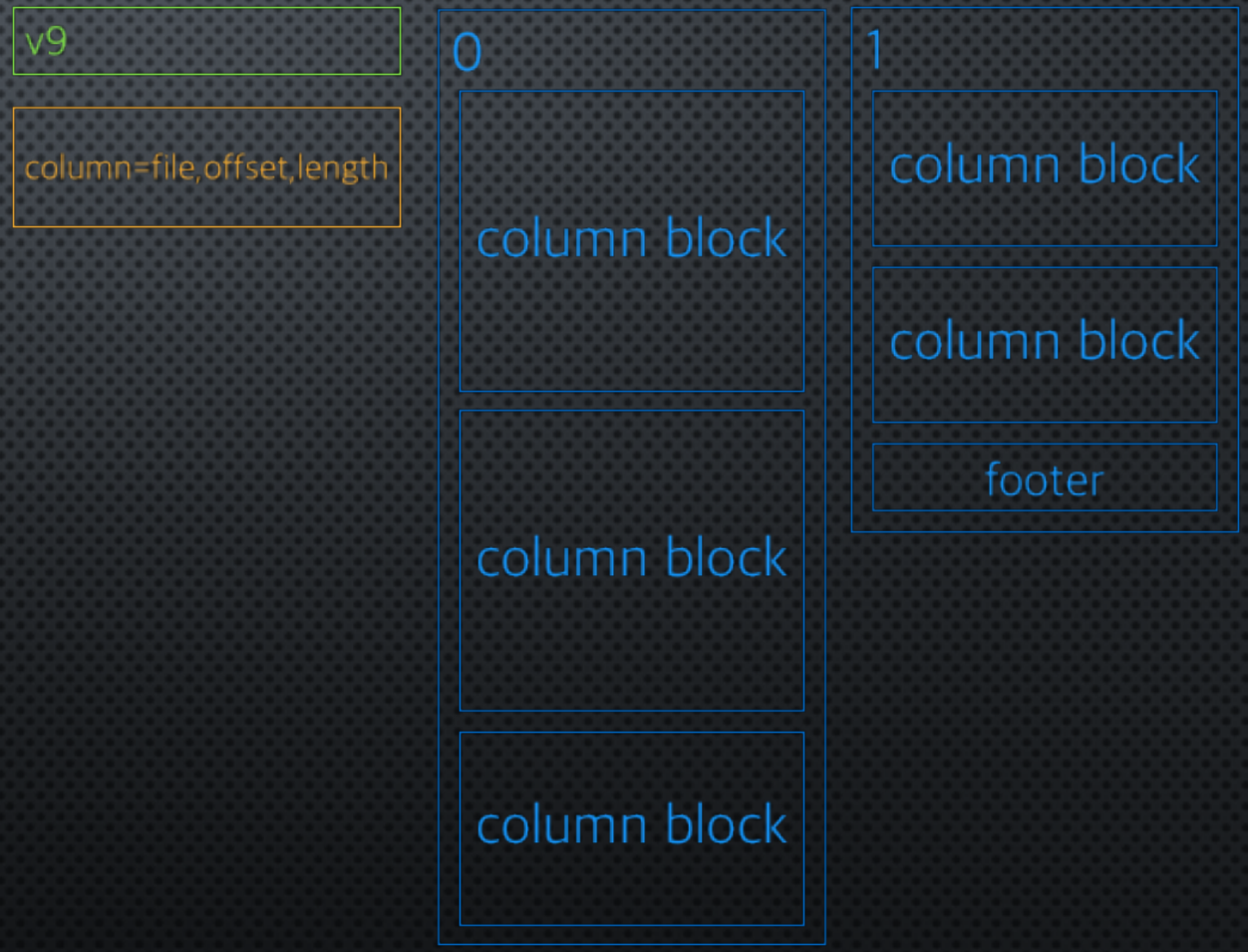


Partition by time



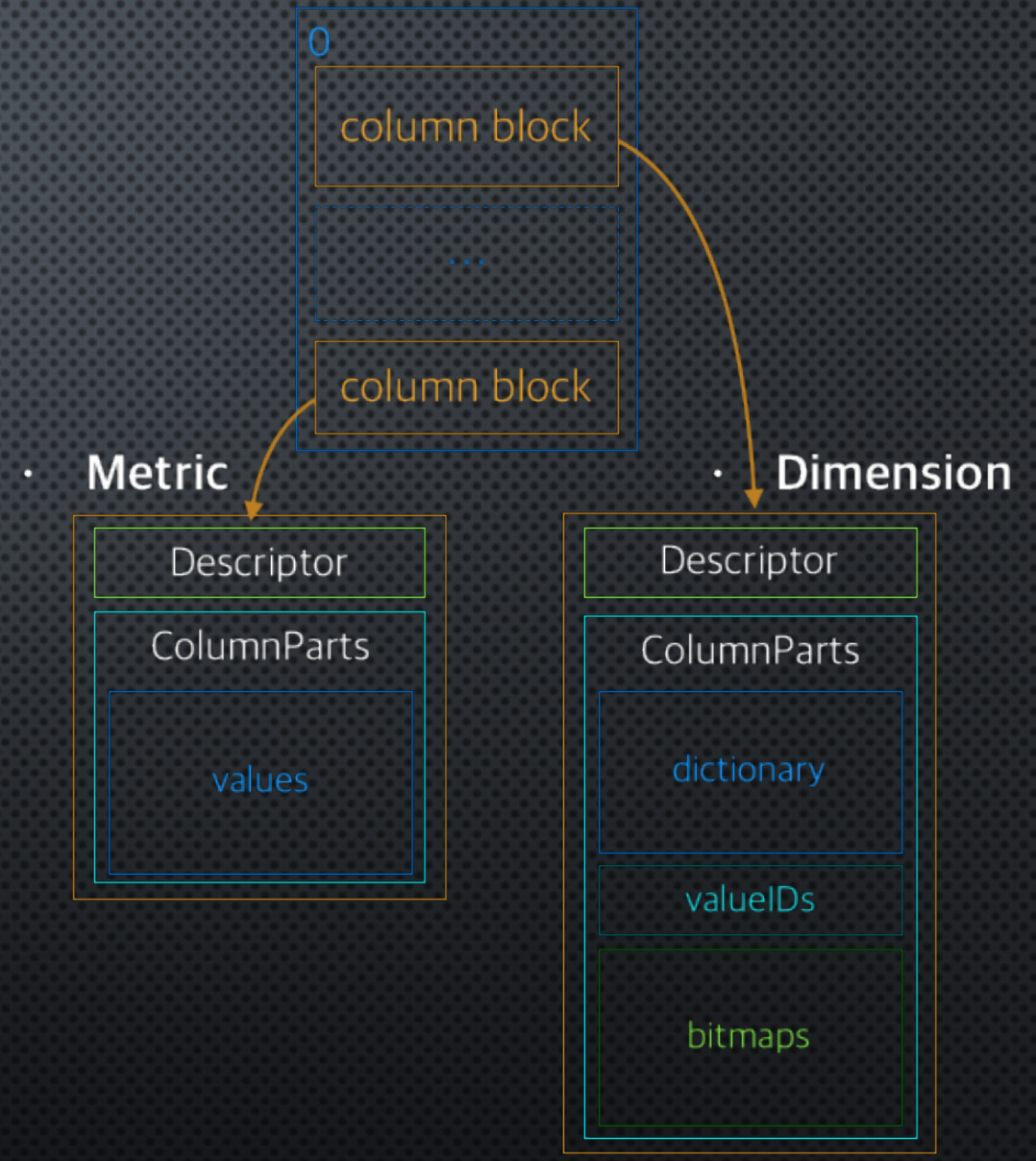
Druid index format : segment

- Immutable, read-only
- Consist of
  - Version file
  - Column offset file(text)
  - Data file(s)
    - Columm blocks + Footer



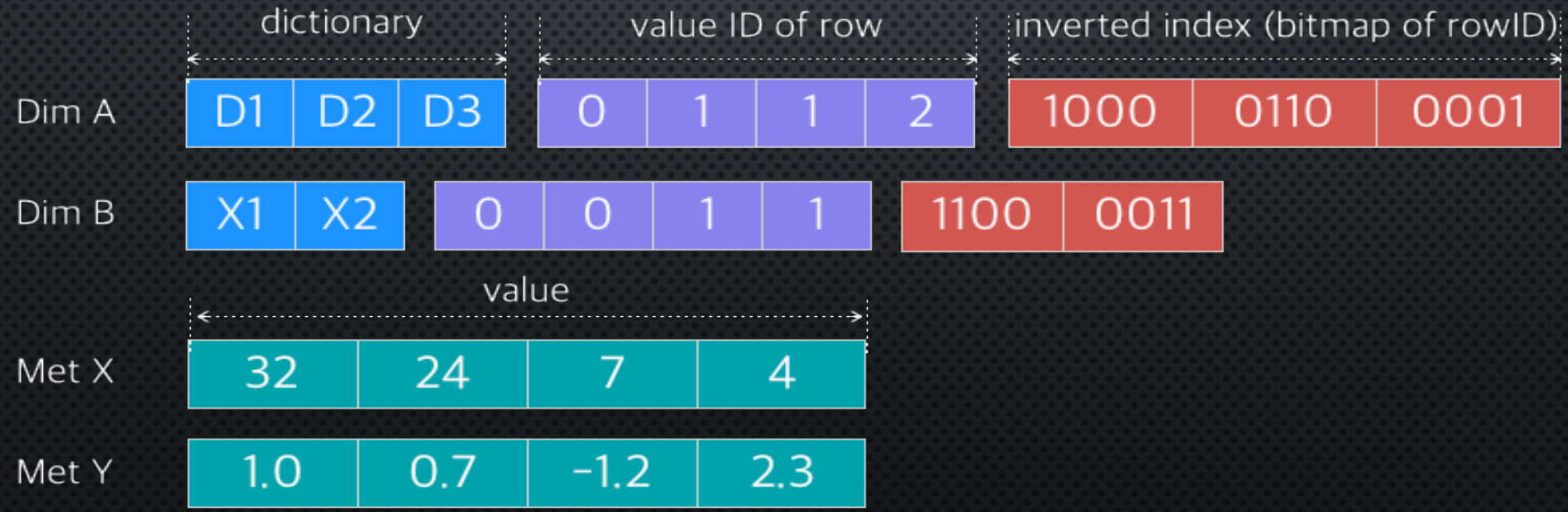
Column Block

- Column Descriptor
  - Type of column
  - SerDes of column part
- Column Part(s)
  - Metric: values Only
  - Dimension: dictionary + valueIDs + bitmaps



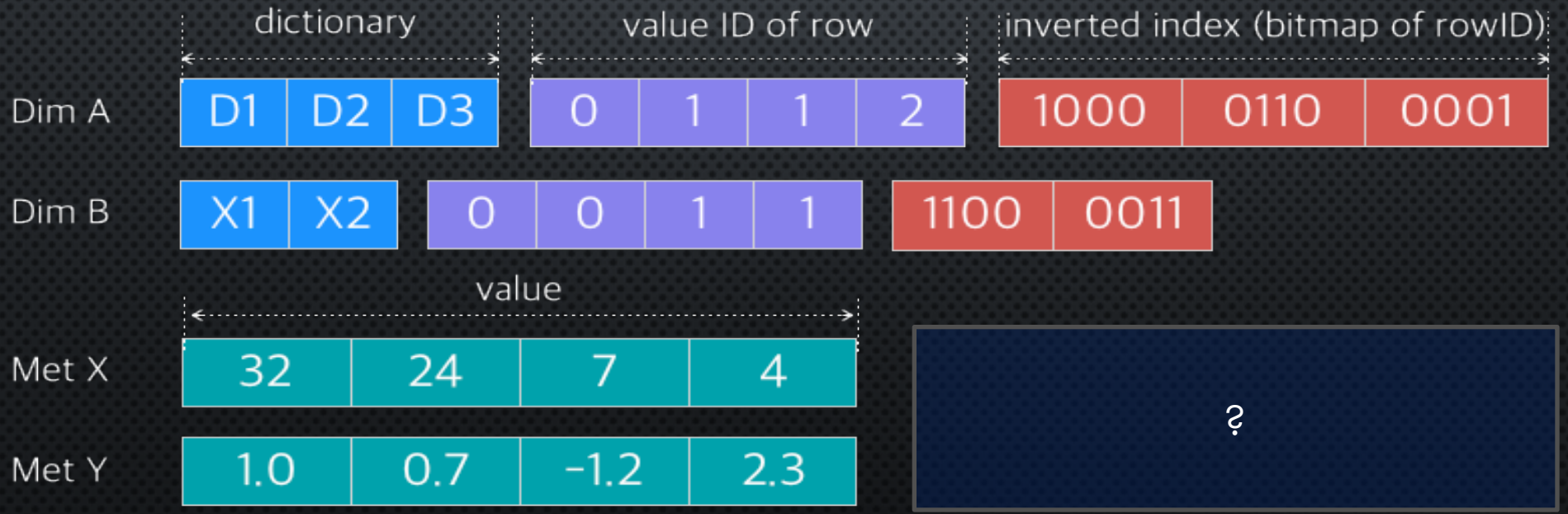
COLUMN PART IN DRUID

Dim A	Dim B	Met X	Met Y
D1	X1	32	1.0
D2	X1	24	0.7
D2	X2	7	-1.2
D3	X2	4	2.3



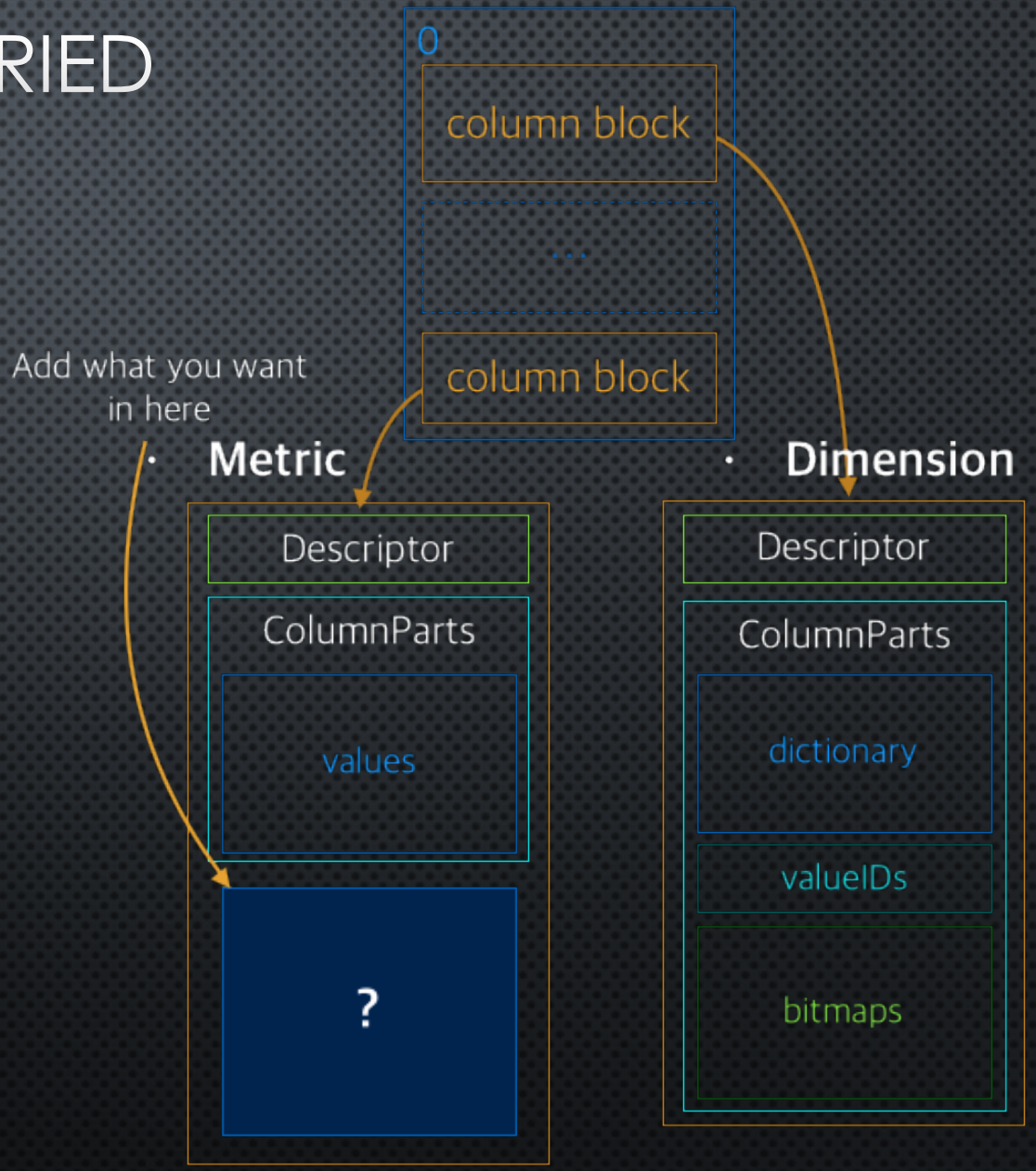
COLUMN PART IN DRUID

Dim A	Dim B	Met X	Met Y
D1	X1	32	1.0
D2	X1	24	0.7
D2	X2	7	-1.2
D3	X2	4	2.3



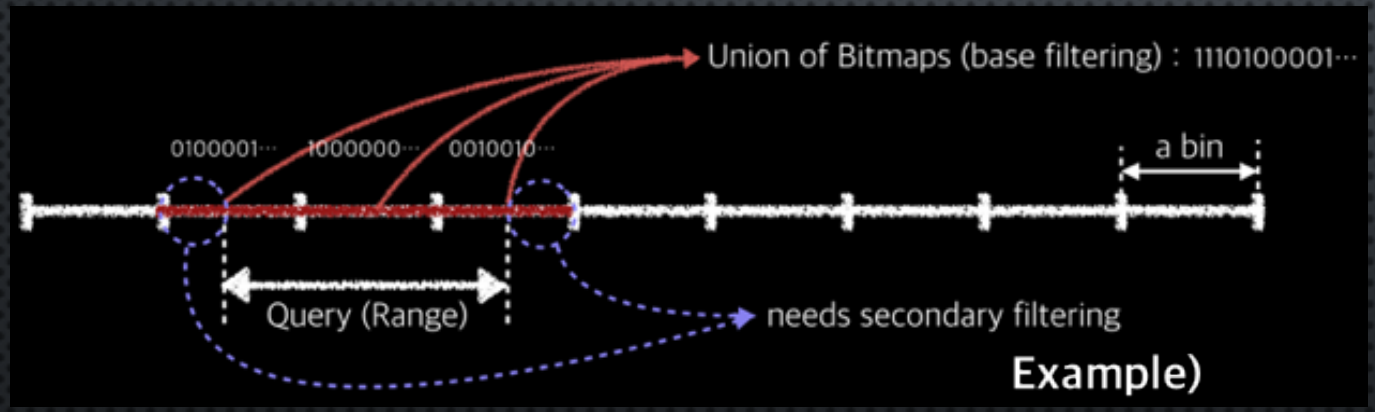
COLUMN PARTS WE'VE TRIED

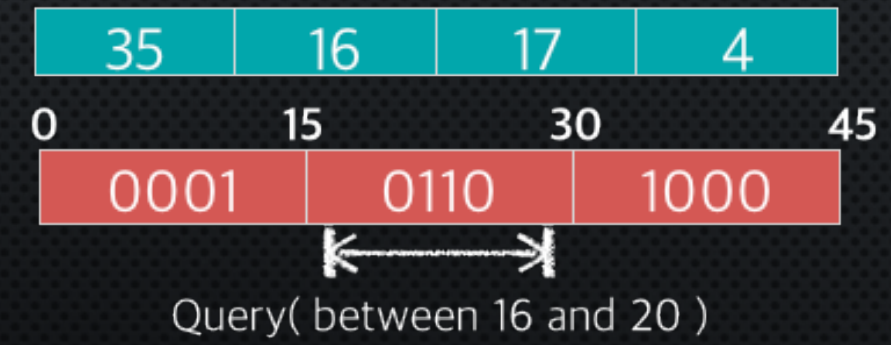
- RANGED HISTOGRAM
- BIT-SLICED BITMAP
- LUCENE INDEX



RANGED HISTOGRAM

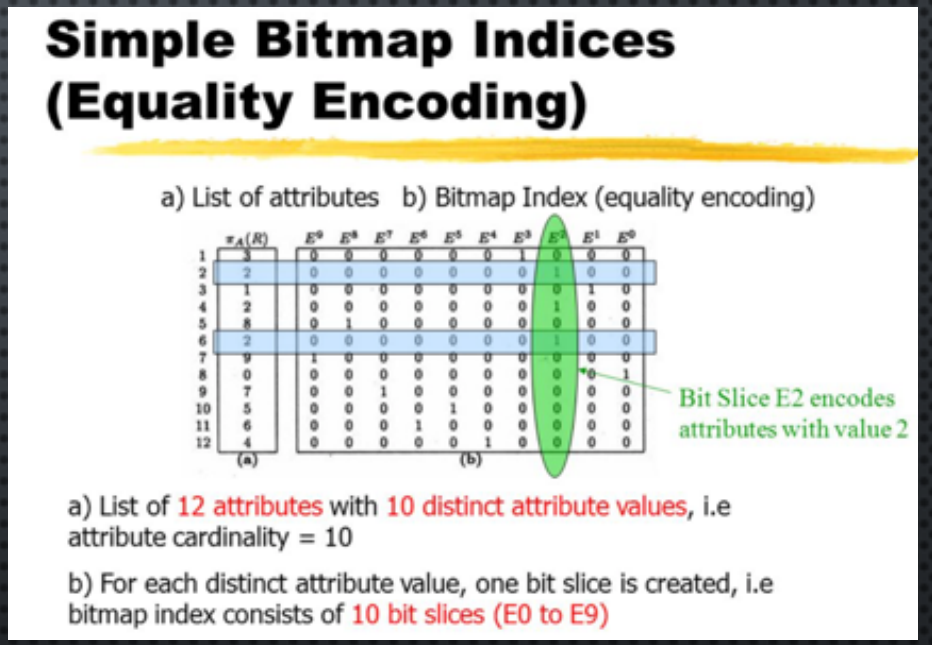
- (BITMAP OF ROWID IN 'SPECIFIC RANGE : BUCKET') X N
- NOT EXACT (NEEDS SECONDARY FILTERING)
- QUICK & CAN APPLY TO ANY COMPARABLE TYPES





BIT-SLICED BITMAP

- EXACT
- EASY TO IMPLEMENT
- LOW COST FOR BUILDING
- ONLY APPLICABLE TO FIXED-LENGTH NUMERIC TYPES



Lucene Index (Text)

- Log messages, SQLs, medical charts, etc.
- Use dimension?: inefficient or even useless → Metric
- Strategy
  - String type metric + Lucene index
  - Rowid as Docid & String value as Text field
    - → Column as lucene Document