Casting the Spell: Druid in Practice

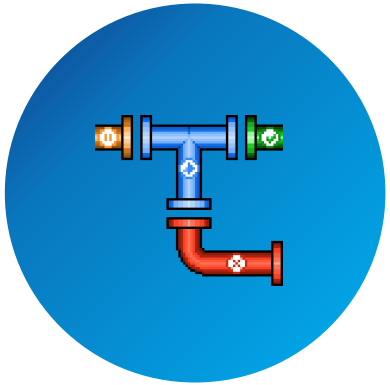
October 2020

Yakir Buskilla, Nielsen
Itai Yaffe, Imply

VIRTUAL
DRUID SUMMIT

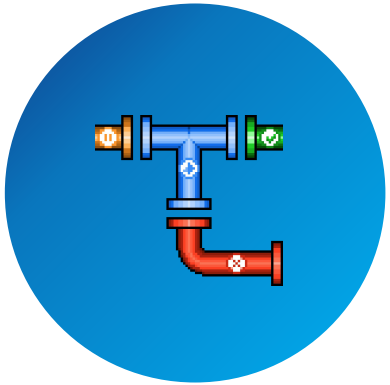
If you ever dealt with big data, you probably ask yourself...

… How to…



Ingest TBs of data

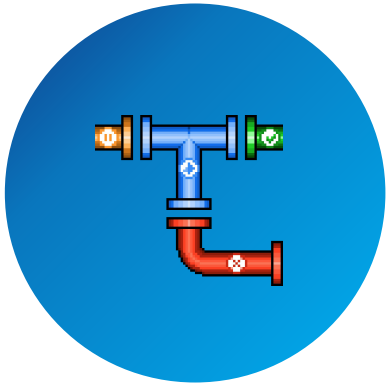
… How to…



Ingest TBs of data

Customer-facing dashboards

… How to…



Ingest TBs of data

Customer-facing dashboards

Cost-efficient

Casting the Spell: Druid in Practice

October 2020

Yakir Buskilla, Nielsen
Itai Yaffe, Imply

VIRTUAL
DRUID SUMMIT

Introduction

Yakir Buskilla

- VP R&D, Nielsen Identity
- Focused on Big Data processing and machine learning solutions
- Yakir Buskilla @yakiro

Itai Yaffe

- Principal Solutions Architect @ Imply
- Prev. Big Data Tech Lead @ Nielsen
- Dealing with Big Data challenges since 2012
- Itai Yaffe @ItaiYaffe

What you will learn?

01. Data Modeling

02. Data Ingestion

03. Retention & Deletion

04. Query Optimization

What you will learn?

01. Data Modeling

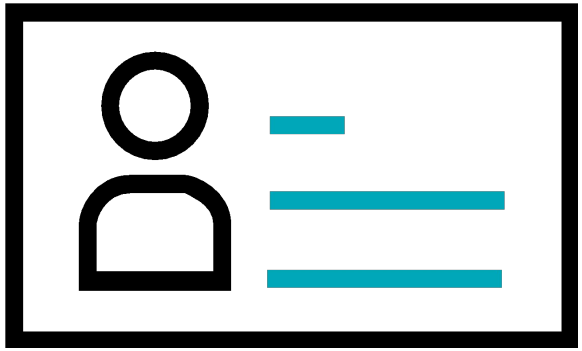
02. Data Ingestion

How to leverage that in your organization

03. Retention & Deletion

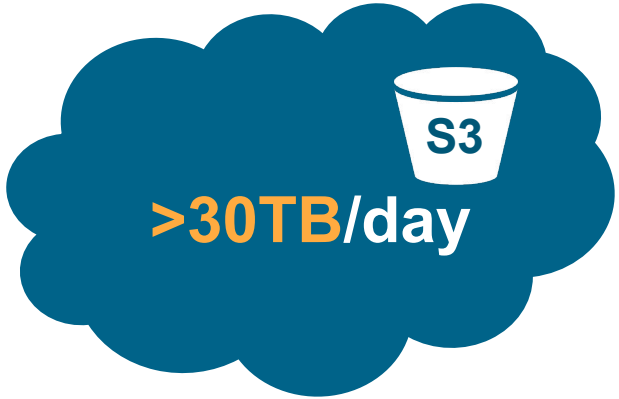
04. Query Optimization

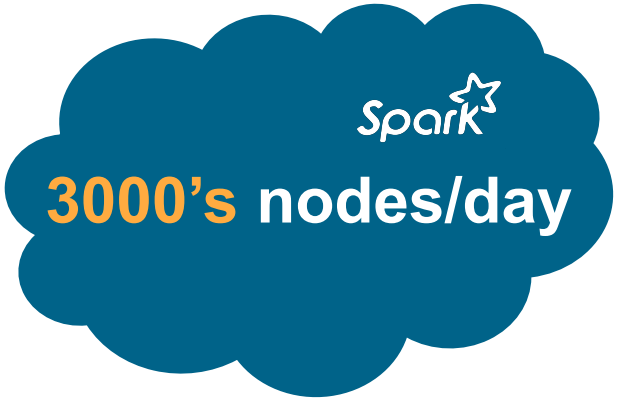
Nielsen Identity

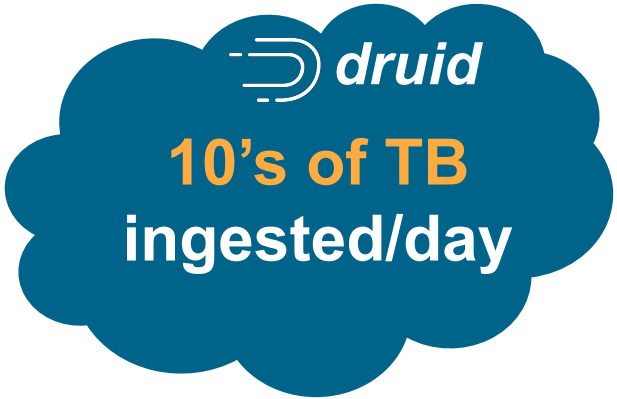


- Data and Measurement company
- Media consumption
- Single source of truth of individuals and households
  - Unifies many proprietary datasets
  - Generates holistic view of a consumer

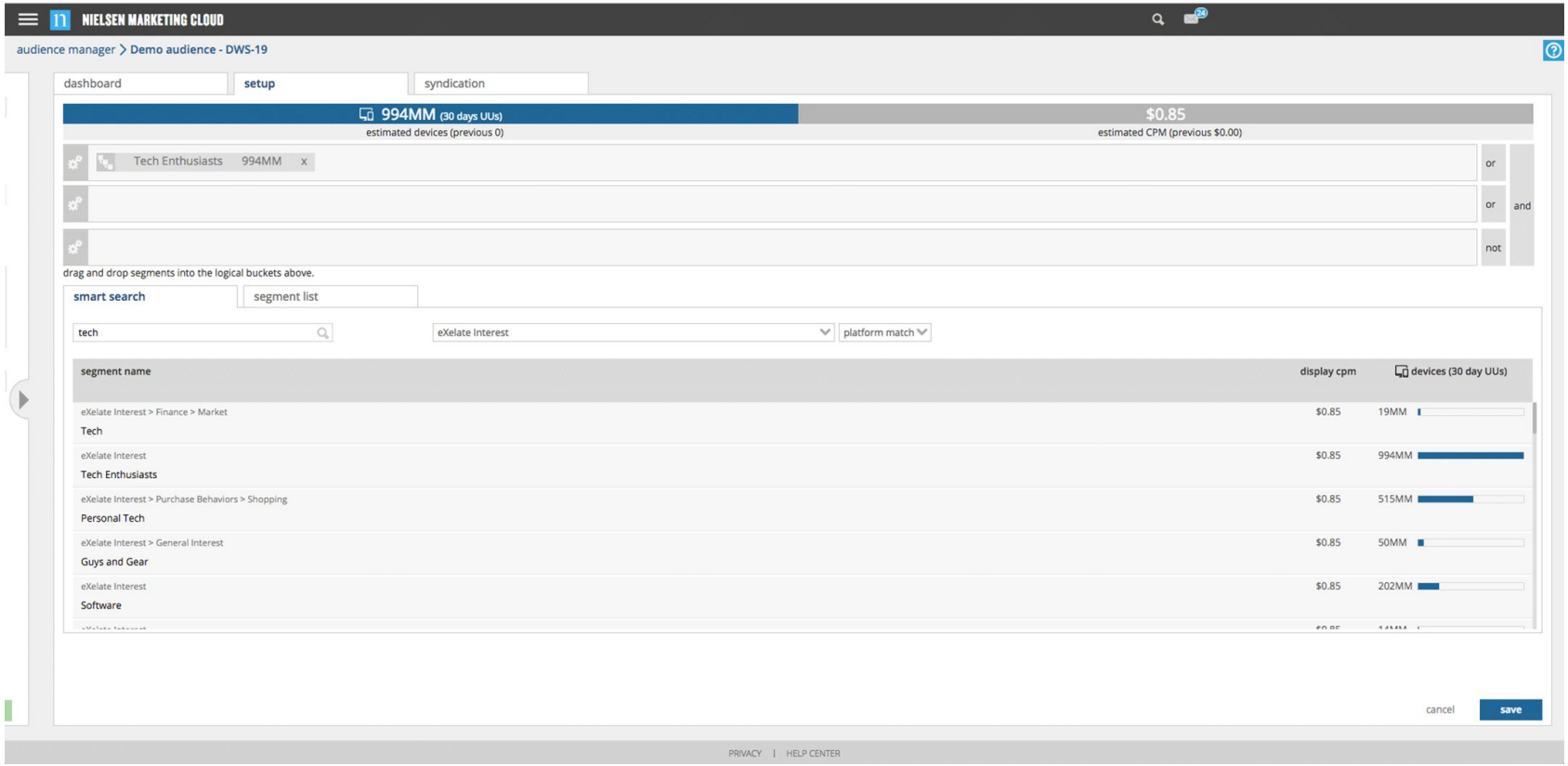
Nielsen Identity in numbers





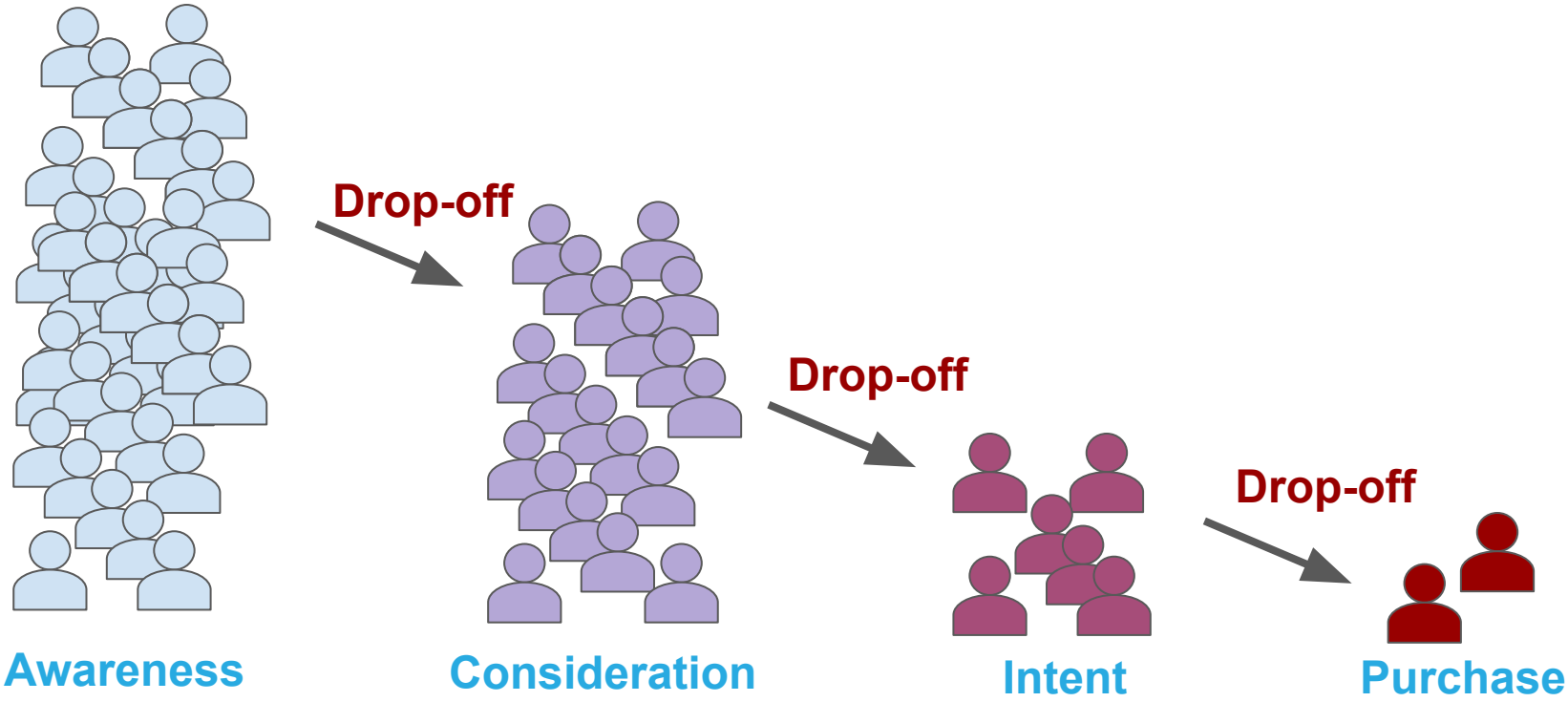


Our use-cases - building target audiences



@ItaiYaffe, @yakiro

Our use-cases - funnel analysis



A jack of all trades

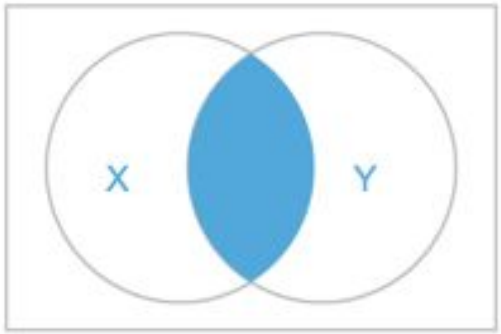
According to Wikipedia, "*The name Druid comes from the shapeshifting Druid class in many role-playing games, to reflect the fact that* ***the architecture of the system can shift to solve different types of data problems***"

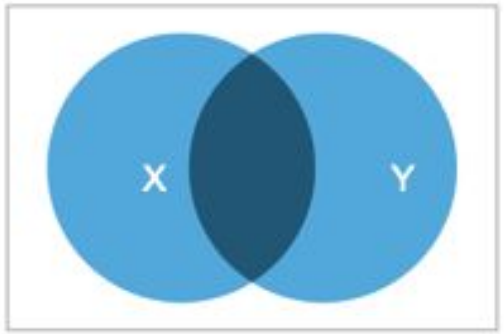
There’s a common thread...

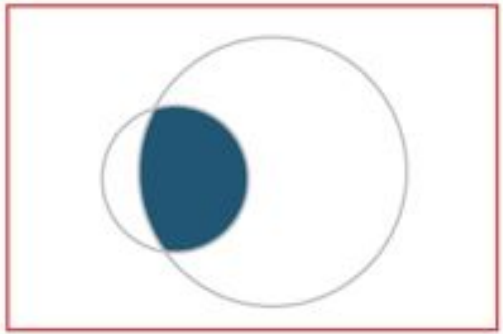
- Counting distinct elements in real-time at scale

What is Theta Sketch?

- K Minimum Values (KMV)
- Estimate set cardinality
- Supports set-theoretic operations





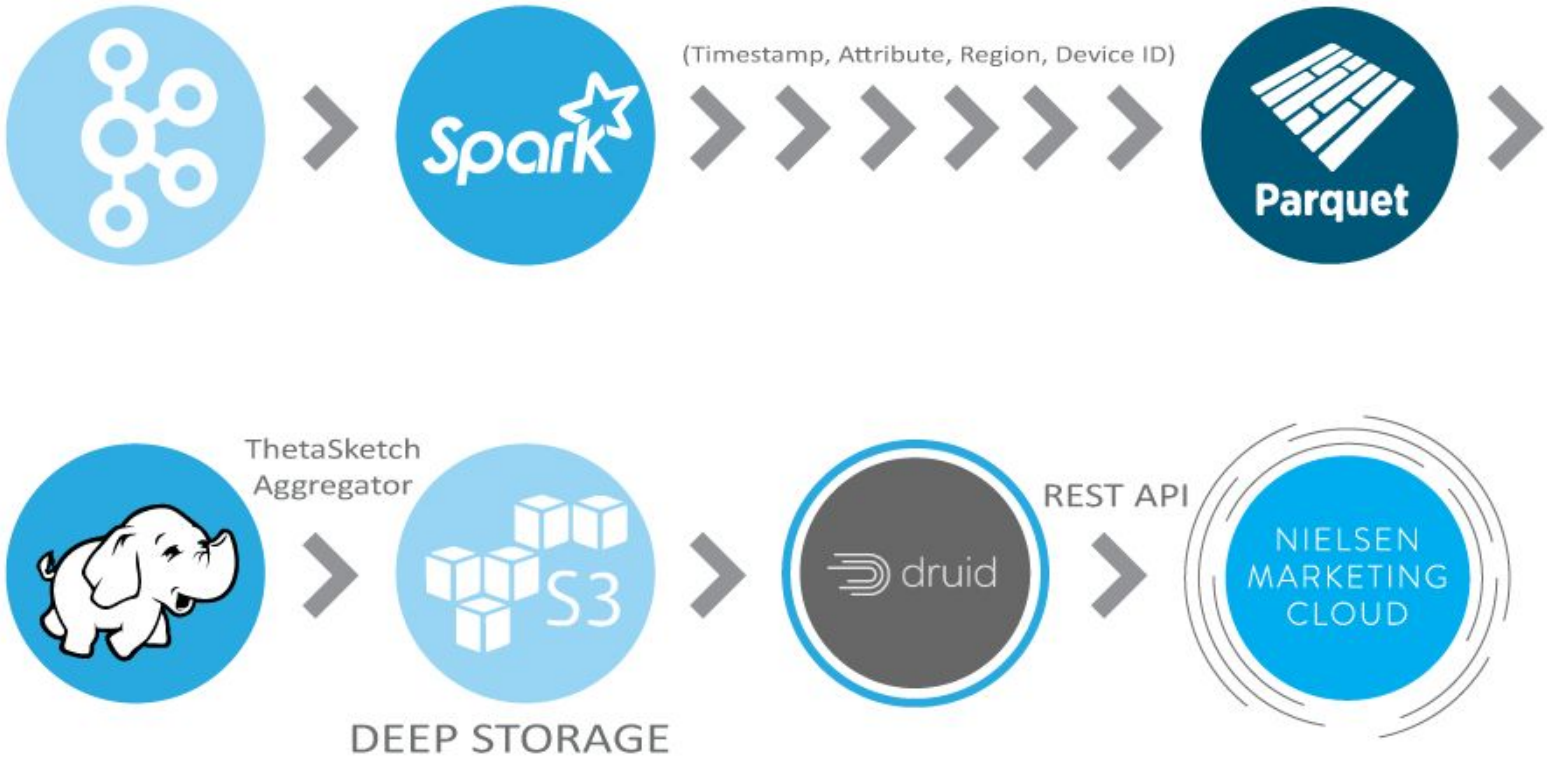


- ThetaSketch mathematical framework - generalization of KMV

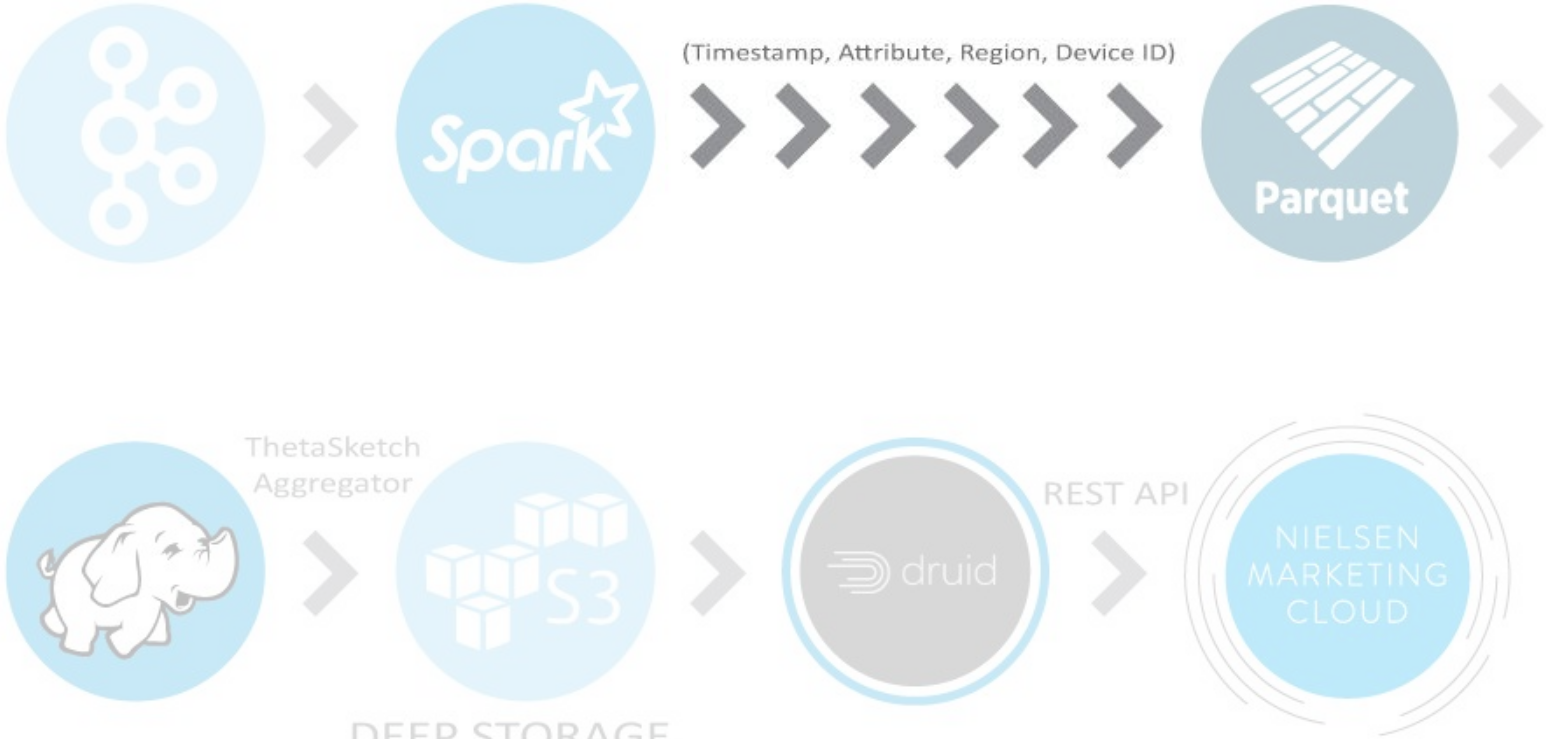
The Theta Sketch module in Druid

- Part of the **DataSketches** library (datasketches.apache.org)
- At **ingestion** time
  - Sketches are **created** and stored in Druid segments
- At **query** time
  - Sketches are **aggregated** (i.e union, intersection or difference between sketches)
  - The result - **estimated number** of **unique entries** in the aggregated sketch
- Also see this short video - tinyurl.com/vdwojh6

High-level overview of the flow



Data modeling



Data model - the naive approach

Timestamp	Audience Name	Device ID
2020-10-05	Mobile	xxx-xxx-xxx
2020-10-05	Football fans	xxx-xxx-xxx
2020-10-05	Tablet	yyy-yyy-yyy
2020-10-05	Druid committers	yyy-yyy-yyy
2020-10-06	Desktop	xxx-xxx-xxx
2020-10-06	Football fans	xxx-xxx-xxx

Data model - introducing Theta Sketch

Timestamp	Audience Name	Theta Sketch
2020-10-05	Mobile	AgMDAAAazJ  AAAAACAP9u
2020-10-05	Football fans	AgMDAAAazJ  AAAAACAP9d
2020-10-05	Tablet	AgMDAAAazJ  AAAAACAPxL
2020-10-05	Druid committers	AgMDAAAazJ  AAAAACAP6J
2020-10-06	Desktop	AgMDAAAazJ  AAAAACAP4h
2020-10-06	Football fans	AgMDAAAazJ  AAAAACAP1

Data model - mitigating Theta Sketch intersections

Timestamp	Audience Name	Device Type	Theta Sketch
2020-10-05	Football fans	Mobile	θ
2020-10-05	Druid committers	Tablet	θ
2020-10-06	Football fans	Desktop	θ

Data model - slowly changing dimensions

Timestamp	Audience Name	Device Type	Theta Sketch
2020-10-05	~~Football fans~~ American football fans	Mobile	
2020-10-05	Druid committers	Tablet	
2020-10-06	~~Football fans~~ American football fans	Desktop	

Data model - lookups

Audience ID	Audience Name
1122	Football fans
3344	Druid committers

Data model - lookups

Audience ID	Audience Name
1122	Football fans
3344	Druid committers

Timestamp	Audience ID	Device Type	Theta Sketch
2020-10-05	1122	Mobile	
2020-10-05	3344	Tablet	
2020-10-06	1122	Desktop	

Data model - lookups

Audience ID	Audience Name
1122	~~Football fans~~ American football fans
3344	Druid committers

Timestamp	Audience ID	Device Type	Theta Sketch
2020-10-05	1122	Mobile	
2020-10-05	3344	Tablet	
2020-10-06	1122	Desktop	

Data model - summary

- Use Theta Sketch for fast and efficient count-distinct
  - Pay attention to intersections
- Leverage lookups to handle slowly changing dimensions
- Check out schema design page on Druid's website

Ingestion


(Timestamp, Attribute, Region, Device ID)
Spark
Parquet
ThetaSketch
Aggregator
S3
druid
REST API
NIELSEN
MARKETING
CLOUD
DEEP STORAGE

Ingestion methods

- Streaming (a.k.a real-time)
  - E.g Kafka, Kinesis
- Batch
  - Hadoop-based, native

Ingestion methods - our choice

- Streaming (a.k.a real-time)
  - E.g Kafka, Kinesis
- Batch
  - **Hadoop-based**, native

Why did we choose Hadoop-based ingestion?

- Technical considerations
  - Maturity
  - Scalability
- Business requirements

Hadoop-based ingestion tips

- Parallel execution of ingestion tasks from seperate Hadoop clusters
  - Can be done using *Affinity*
- Ingestion tasks run **2 MapReduce jobs** by default
  - 1- Determine partitions (based on *targetRowsPerSegment*)
  - 2- Index partitions
  - To **speed-up ingestion**, **set *numShards*** in advance in *partitionsSpec*

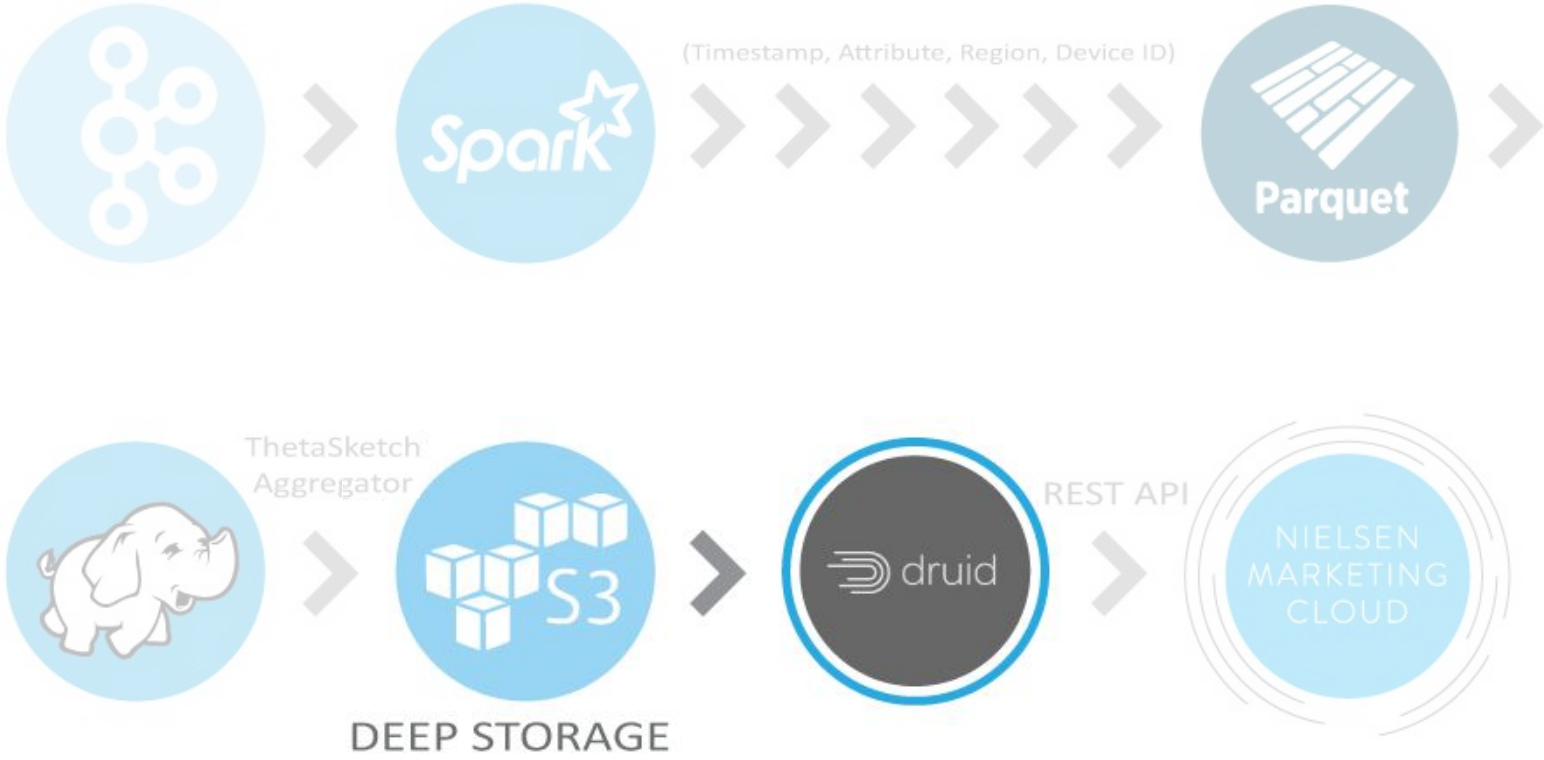
Ingestion - future plans

- Remove Hadoop MapReduce dependency, by either:
  - Ingest data directly from Spark - github.com/apache/druid/issues/9780 (WIP)
  - Use native batch ingestion in a scalable way

Ingestion - summary

- There are multiple options
- We chose Hadoop-based ingestion
  - Affinity
  - targetRowsPerSegment vs numShards

Retention & deletion



Retention & deletion

- Load Rules
  - Which segments to **load** into the cluster (by interval or period)
  - How many **replicas** per segment
- Drop Rules
  - When segments should be **dropped** from the cluster (but not deleted from deep storage)
- Kill Tasks
  - **Permanently delete** all information about a segment and remove it from deep storage

Retention & deletion tips

- Segments have **versions**
  - Versions **take up space** in deep storage
  - Only the latest version is actually used (*used=1* in the metadata store)
- The interval specified in your Kill Task should be **as wide as possible**
  - It'll delete **only** those segments that are marked as "unused" (*used=0*)

Retention & deletion example

- 1 datasource with 1-year retention period
- Deep storage is AWS S3

Topic	> 30 days
Used Storage	365TB
Costs/month	$8.3K

Retention & deletion example

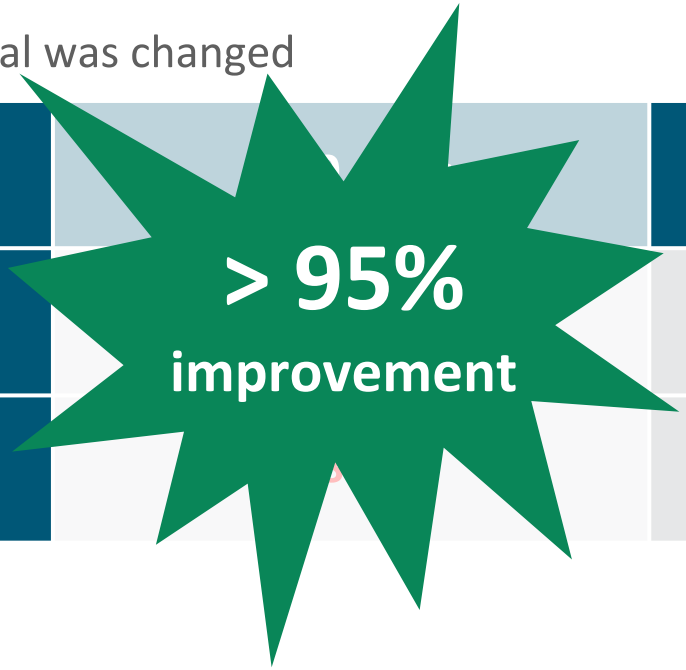
- 1 datasource with 1-year retention period
- Deep storage is AWS S3
- Only the Kill Task interval was changed

Topic	> 30 days	> 2 days
Used Storage	365TB	15TB
Costs/month	$8.3K	$350

Retention & deletion example

- 1 datasource with 1-year retention period
- Deep storage is AWS S3
- Only the Kill Task interval was changed

Topic	> 2 days
Used Storage	15TB
Costs/month	$350



Dimension-based TTL

Timestamp	Country	Device Type	Theta Sketch
2020-09-04	US	Laptop	
2020-09-04	Sweden	Smart TV	
2020-09-05	US	Mobile	
2020-09-05	Sweden	Tablet	
…	…	…	…
2020-10-06	US	Desktop	
2020-10-06	Sweden	Mobile	

Dimension-based TTL

Timestamp	Country	Device Type	Theta Sketch
2020-09-04	US	Laptop	
2020-09-05	US	Mobile	
…	…	…	…
2020-10-06	US	Desktop	
2020-10-06	Sweden	Mobile	
2020-10-07	US	Laptop	
2020-10-07	Sweden	Desktop	

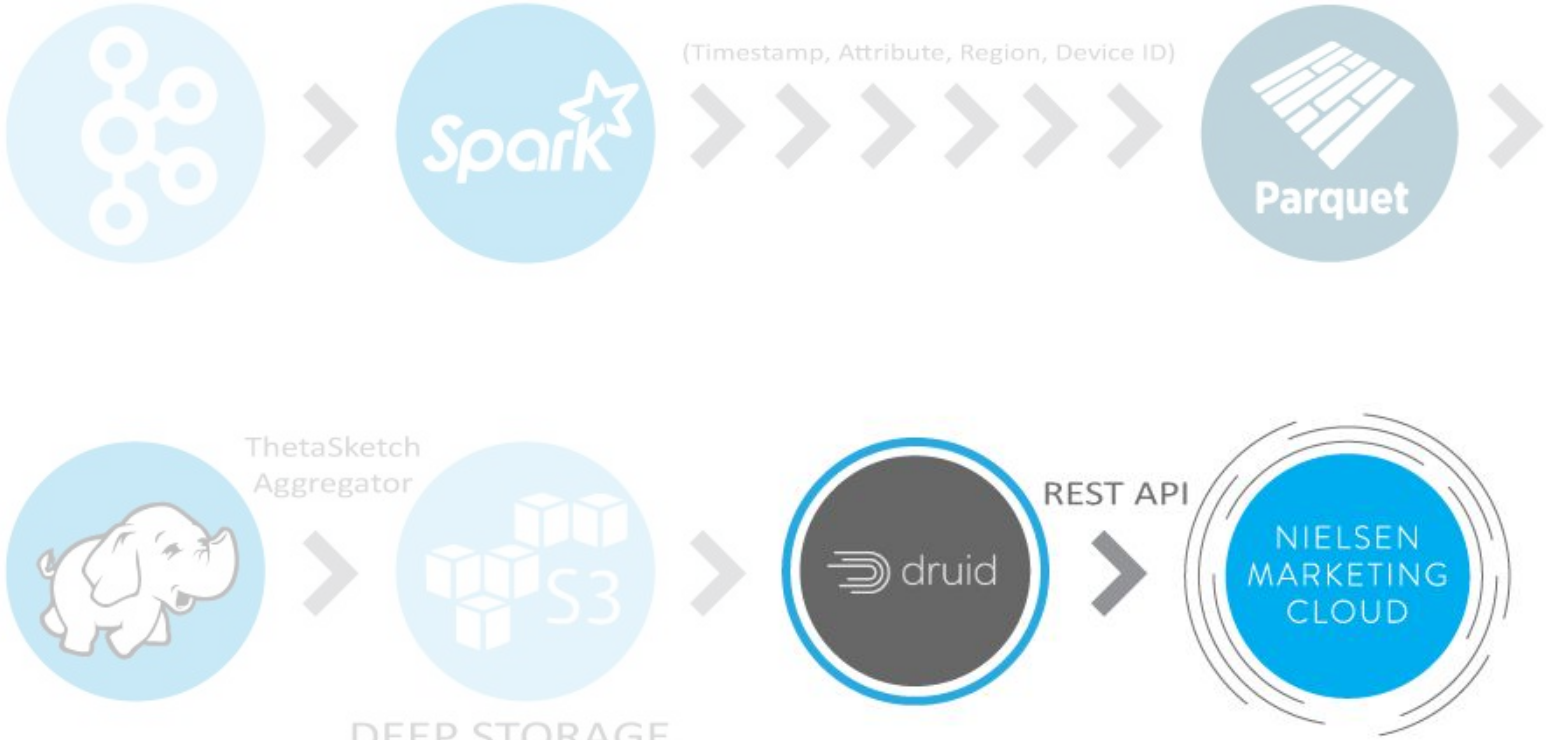
Dimension-based TTL

- Using Hadoop-based ingestion with *multi* type *inputSpec*
  - Allows you to combine other *inputSpecs*
  - In the *"type": "dataSource"*, you can use a filter as part of the *ingestionSpec*
- For more info, check out tinyurl.com/yymrvrn2

Retention & deletion - summary

- Load & drop rules
- Kill tasks
- Dimension-based TTL

Queries


(Timestamp, Attribute, Region, Device ID)
Spark
Parquet
ThetaSketch
Aggregator
S3
DEEP STORAGE
druid
REST API
NIELSEN
MARKETING
CLOUD

Query methods

- Native queries
- Druid SQL

Query methods - our choice

- Native queries
- Druid SQL

Why did we choose native queries?

- Mainly because Druid SQL didn’t exist yet…
- Druid SQL is expanding with each version
  - Querying Theta Sketches from SQL was added in 0.14.0
  - github.com/apache/druid/pull/6951
  - github.com/apache/druid/issues/7126

Query optimization

- Tune Theta Sketch size for **performance**
  - Queries with size=65536 are “heavy” thus take more time
  - Switching to size=4096 will **improve speed**
    - When there are no intersections
    - Relatively small effect on accuracy

Query optimization

- Tune Theta Sketch size for **accuracy**
  - ```
    SELECT APPROX_COUNT_DISTINCT_DS_THETA(user_id_sketch)
    FROM campaign_1012
    WHERE tactic = 1 AND
    __time BETWEEN TIMESTAMP '2018-02-01' AND TIMESTAMP '2020-09-08'
    ```
    Result = **28945757**
    ```

Query optimization

- Tune Theta Sketch size for **accuracy**
  - SELECT **APPROX_COUNT_DISTINCT_DS_THETA**(user_id_sketch)
    FROM campaign_1012
    WHERE tactic = 1 AND
    __time BETWEEN TIMESTAMP '2018-02-01' AND TIMESTAMP '2020-09-08'
    Result = **28945757**
  - SELECT **APPROX_COUNT_DISTINCT_DS_THETA**(user_id_sketch,**65536**)...
    Result = **29356320** => **more accurate**

Query optimization

- Tune Theta Sketch size for **accuracy**
  - SELECT **APPROX_COUNT_DISTINCT_DS_THETA**(user_id_sketch)
    FROM campaign_1012
    WHERE tactic = 1 AND
    __time BETWEEN TIMESTAMP '2018-02-01' AND TIMESTAMP '2020-09-08'
    Result = **28945757**
  - SELECT **APPROX_COUNT_DISTINCT_DS_THETA**(user_id_sketch,**65536**)...
    Result = **29356320 => more accurate**
  - APPROX_COUNT_DISTINCT_DS_THETA(expr, [size]) - size **defaults to 16384**

Queries - summary

- 2 query methods
- We chose native queries
- Tune Theta Sketch
  - Balance between performance and accuracy

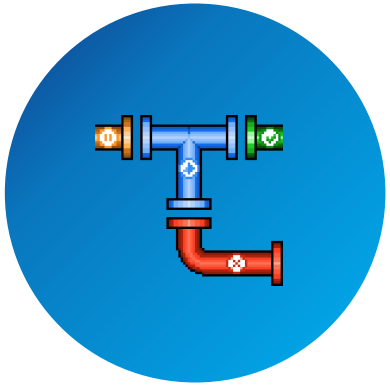
Bonus slide - hardware

- There are different considerations when choosing your hardware, e.g
  - Cost
  - Performance
  - Usage patterns
- Actual example:
  - No. of datasources - <10 (all are ThetaSketch)
  - Data size on cluster - ~45TB
  - Broker nodes - 3 X r4.8xlarge (32 cores, 244GB RAM each)
  - Historical nodes - 20 X i3.8xlarge (32 cores, 244GB RAM each, NVMe SSD)

Bonus slide - Dev cluster

- Setting-up Dev cluster using the **same deep storage** as Prod cluster
  - Create a **read & write** role for the **Prod** cluster
    - Allows you to ingest new data
  - Create a **read-only** role for the **Dev** cluster
    - Prevents you from (accidentally) ingesting new data
  - Periodically restore metadata store from Prod backup to Dev

So how can you…



Ingest TBs of data

Customer-facing dashboards

Cost-efficient

Want to know more?

- Women in Big Data
  - A world-wide program that aims :
    - To inspire, connect, grow, and champion success of women in the Big Data & analytics field
  - 30+ chapters and 17,000+ members world-wide
  - Everyone can join (regardless of gender), so find a chapter near you - https://www.womeninbigdata.org/wibd-structure/

- Funnel Analysis with Spark and Druid - tinyurl.com/y5qboqpj

- Our Tech Blog - medium.com/nmc-techblog
  - Data Retention and Deletion in Apache Druid - tinyurl.com/yymrvrn2

Thank you!

Time for questions

@ItaiYaffe
@yakiro

VIRTUAL
DRUID SUMMIT

Register Now for the Next Druid Virtual Summit

VIRTUAL
DRUID SUMMIT

Dates: TBD

imply.io