Building data applications with Apache Druid

October 2020

Gian Merlino, Cofounder & CTO, Imply

VIRTUAL
DRUID SUMMIT

Who am I?

Gian Merlino
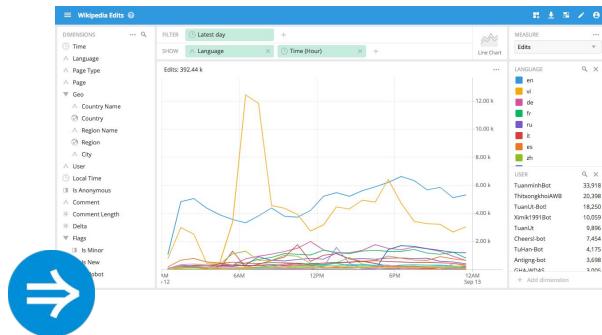
Committer & PMC chair at druid

Cofounder at imply (we're hiring!)

Building data apps

VIRTUAL
DRUID SUMMIT

Druid powers data apps

Druid in the wild

How Netflix uses Druid for Real-time Insights to Ensure a High-Quality Experience

Netflix Technology Blog [Follow]
Mar 3 · 9 min read

By *Ben Sykes*

100+ billion rows/day

1+ trillion rows, 1+ year retained

100s of servers

sub-second to few seconds query latency

mix of streaming and batch ingest

Interactive Analytics at MoPub: Querying Terabytes of Data in Seconds

By Aaron Rolett and Shravana Krishnamurthy
Wednesday, 3 July 2019

How Druid enables analytics at Airbnb

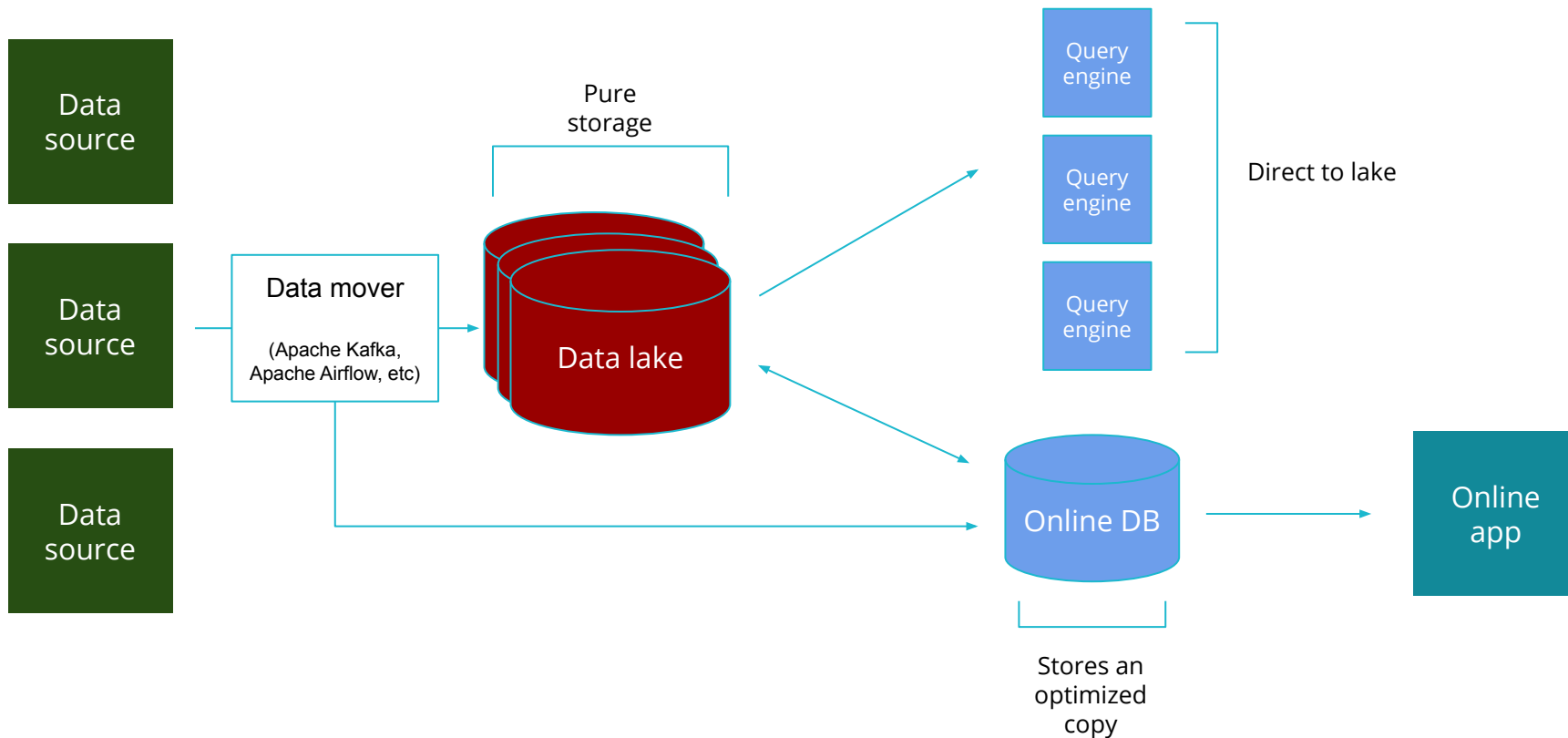Realtime and batch analytics at Airbnb and the role Druid plays in our analytics system architecture
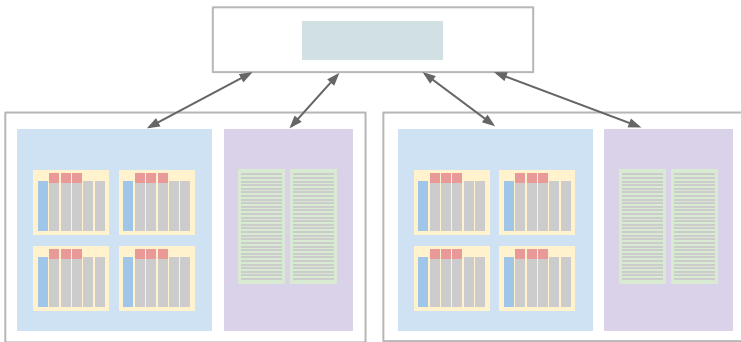
Pala [Follow]
Nov 13, 2018 · 10 min read

By *Pala Muthiah* and *Jinyang Li*

Online DB pattern

Data source

Data source

Data source

Data mover

(Apache Kafka, Apache Airflow, etc)

Pure storage

Data lake

Query engine

Query engine

Query engine

Direct to lake

Online DB

Online app

Stores an optimized copy
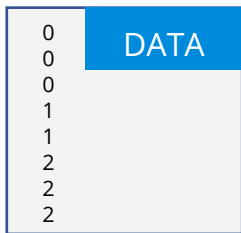
Druid as an online DB

- Scale-out, fault-tolerant architecture

- No downtime for software updates

- No downtime for data management

- Heavily optimized storage format
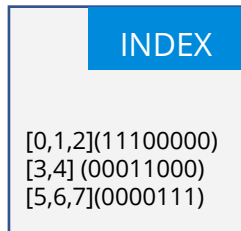
- Integrated storage format and query engine

Data apps

- Interactive query speeds

- Always online

- Fresh data from streams

- Quality of service

- Price/performance

Interactive query speps

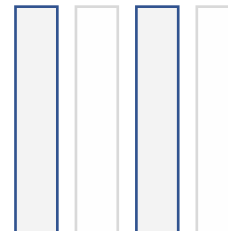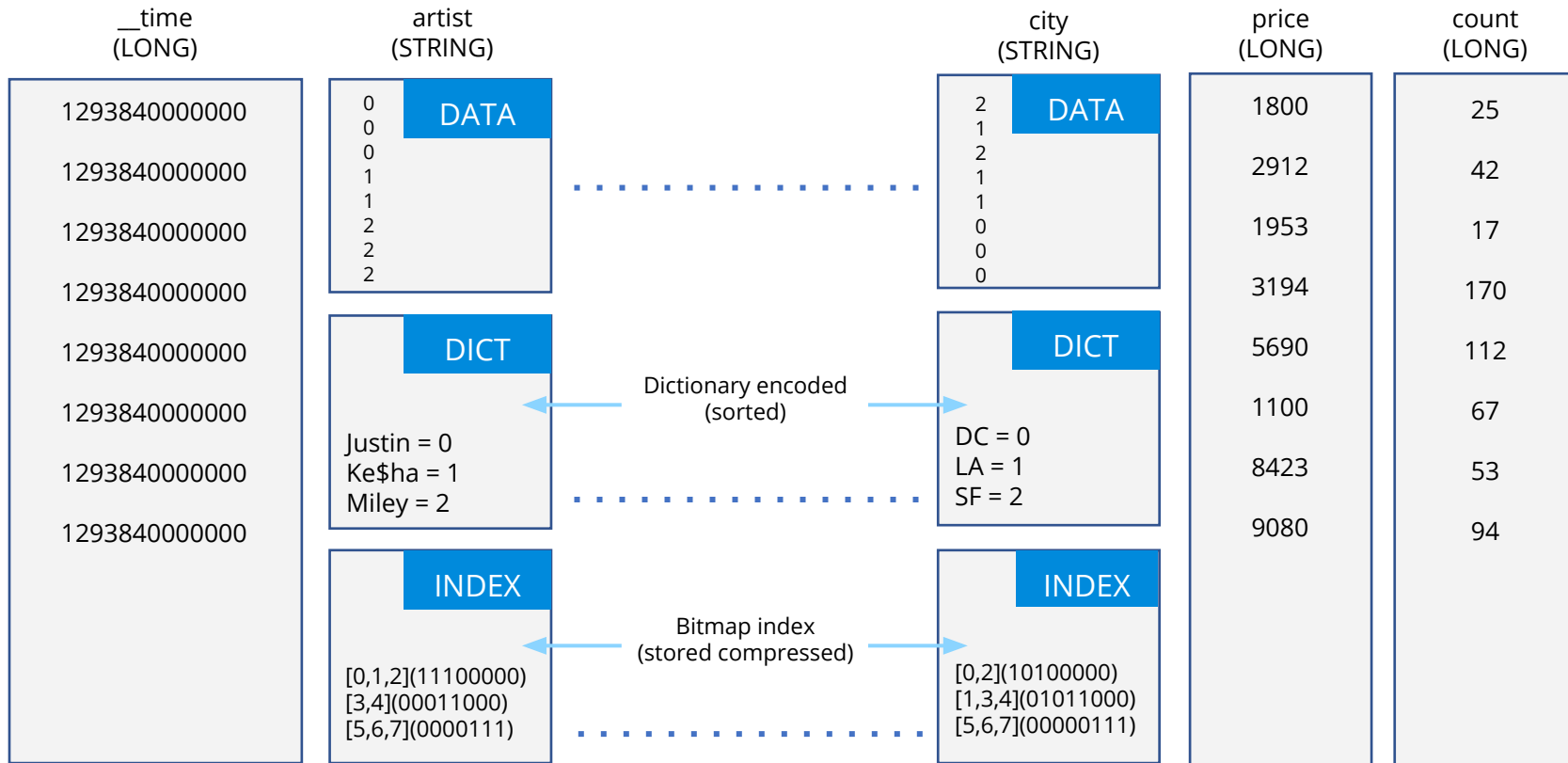| DATA	INDEX	DICT	
0 0 0 1 1 2 2 2 2	[0,1,2](11100000) [3,4] (00011000) [5,6,7](0000111)	DC = 0 LA = 1 SF = 2	
Compression	Secondary indexes	Operate on compressed data	Late materialization
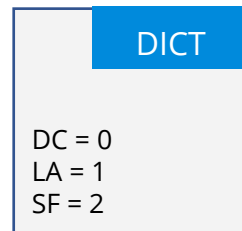
Interactive query speeds

__time (LONG)	artist (STRING)		city (STRING)		price (LONG)	count (LONG)

__time (LONG)

1293840000000

1293840000000

1293840000000

1293840000000

1293840000000

1293840000000

1293840000000

1293840000000

artist (STRING)

DATA

0
0
0
1
1
2
2
2

DICT

Justin = 0
Ke$ha = 1
Miley = 2

INDEX

[0,1,2](11100000)
[3,4](00011000)
[5,6,7](0000111)

city (STRING)

DATA

2
1
2
1
1
0
0
0

DICT

DC = 0
LA = 1
SF = 2

INDEX

[0,2](10100000)
[1,3,4](01011000)
[5,6,7](00000111)

Dictionary encoded (sorted)

Bitmap index (stored compressed)

price (LONG)

1800

2912

1953

3194

5690

1100

8423

9080

count (LONG)

25

42

17

170

112

67

53

94

Always online

Query server

Broker

Master server

Data server

Coordinator

Apache ZooKeeper

Historical

Indexer

Historical

Indexer

Streaming data

Batch data

Deep storage

Fresh data from streams

Query server

Broker

Master server

Coordinator

Apache
ZooKeeper

Data server

Historical

Indexer

Historical

Indexer

Streaming
data

Batch
data

Deep storage

Quality of service

Quality of service

Query server

Broker

Master server

Data server

Coordinator

Apache ZooKeeper

Historical

Indexer

Historical

Indexer

Streaming data

Batch data

Deep storage
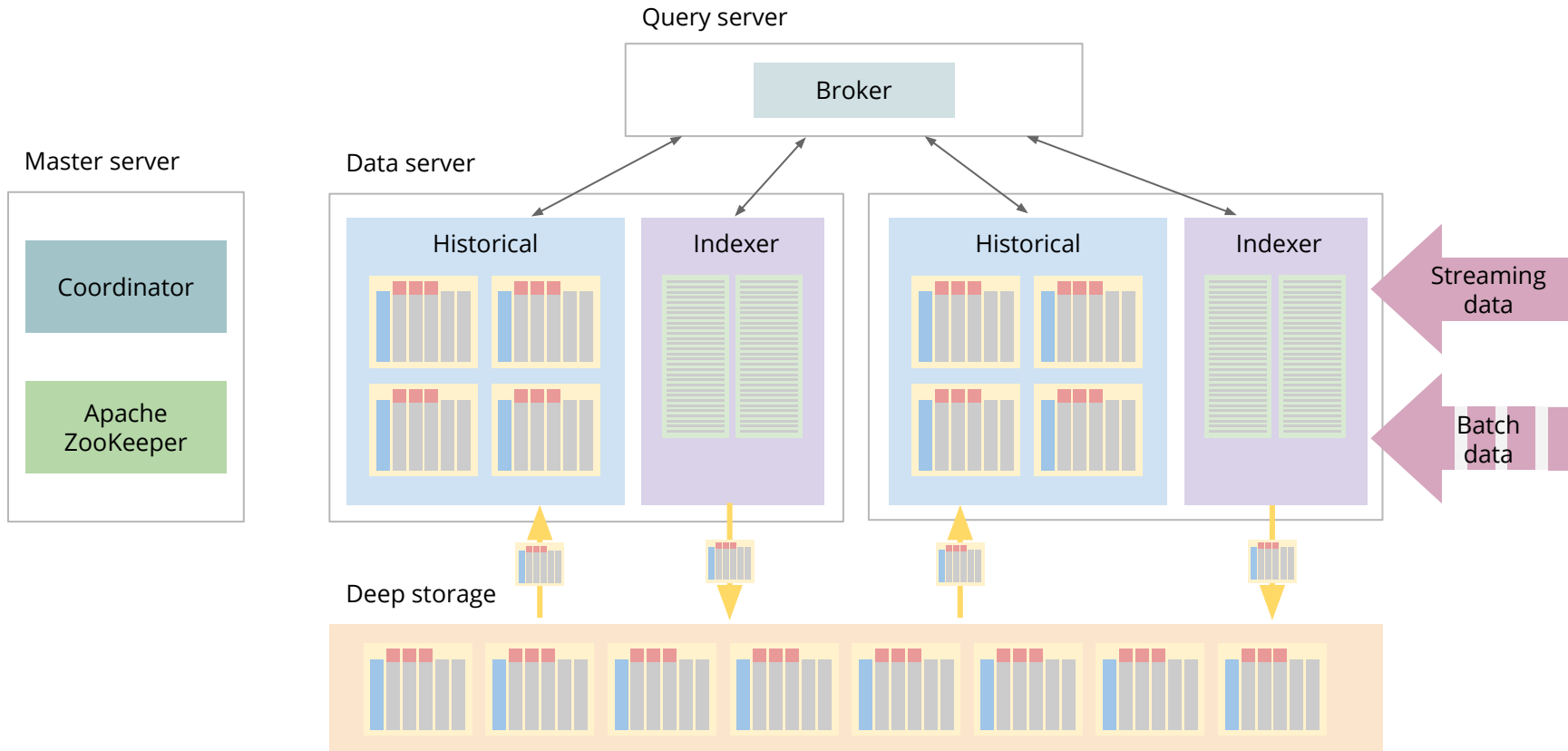
Price/performance

vs. leading open source SQL engines

Data sourced from: *Correia, José & Costa, Carlos & Santos, Maribel. (2019). Challenging SQL-on-Hadoop Performance with Apache Druid.*
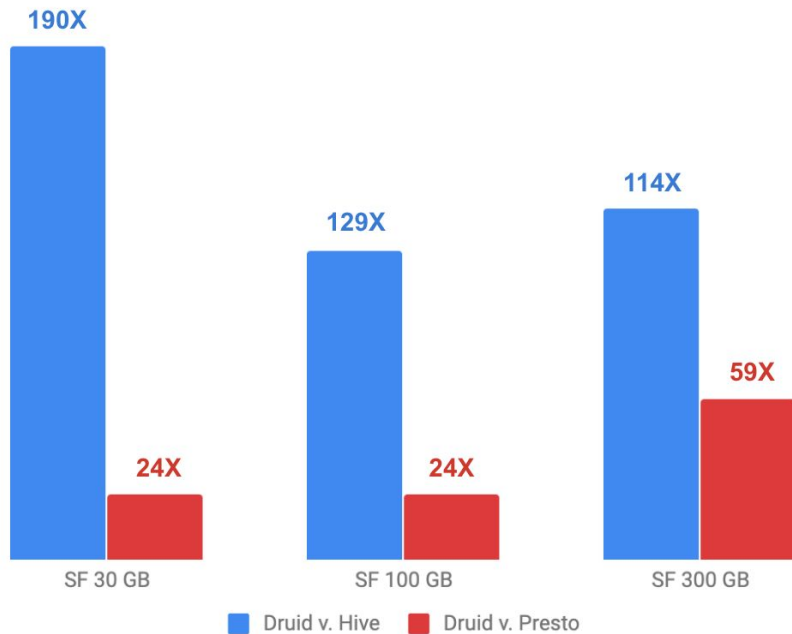
Price/performance

vs. a leading cloud-based data warehouse

Data sourced from Imply benchmarks.
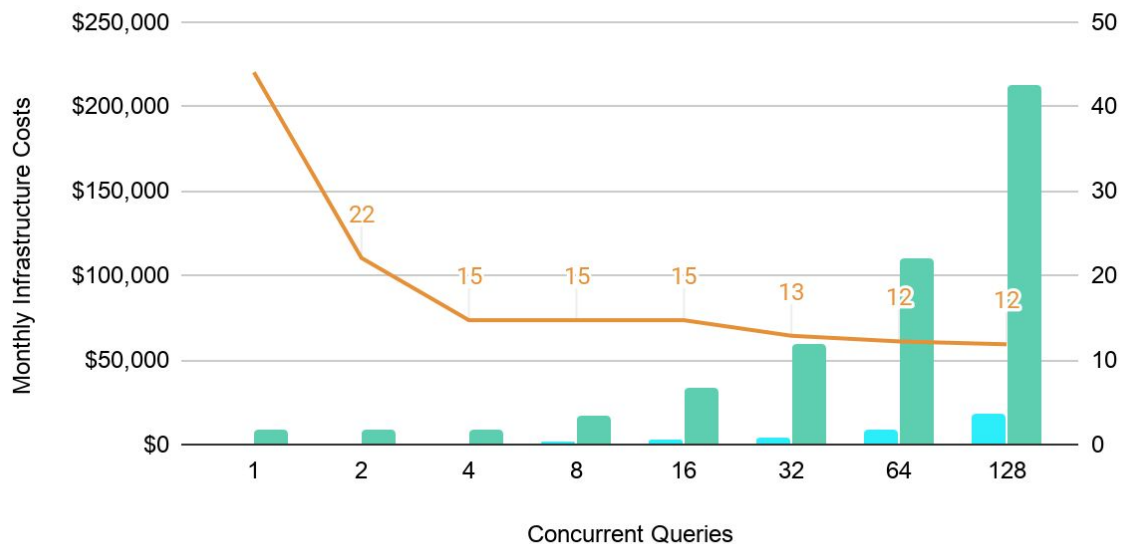
Wrapping up

VIRTUAL
DRUID SUMMIT

Stay in touch

Follow the Druid project on Twitter!

@druidio

Join the community
(Mailing lists, Slack, meetups)
https://druid.apache.org/community/

Thank you!

Time for questions

@gianmerlino

VIRTUAL
DRUID SUMMIT