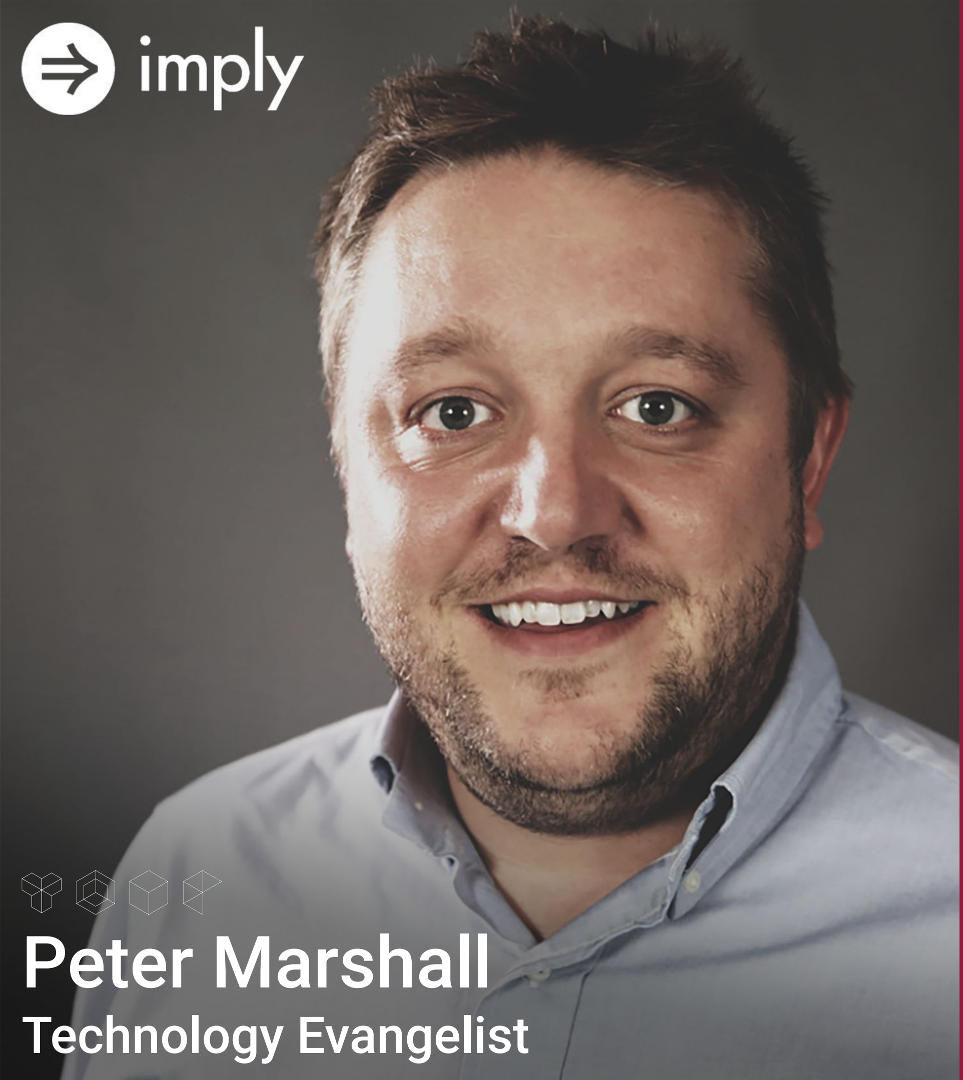
imply

druid®

Adoption Tips & Tricks

Apache®, Apache Druid®, Druid®, and the Druid logo are either registered trademarks or trademarks of The Apache Software Foundation in the United States and other countries.

imply



Peter Marshall

Technology Evangelist

20 years in Enterprise Architecture

CRM, EDRM, ERP, EIP, Digital Services, Security, BI, RI, and MDM

BA Theology (!) and Computer Studies

TOGAF certified

Book collector & A/V buyer
Prime Timeline = *proper* timeline
#werk

petermarshall.io

peter.marshall@imply.io

Community Issues

Tips & Tricks

Call to Action

Questions & Answers

Rachel Pedreschi
Vice President of Community

@rachelpedreschi
rachel@imply.io

Jelena Zanko
Senior Community Manager

@JelenaZanko
jelena.zanko@imply.io

ASSIST

WRITE

EXCHANGE

Matt Sarrel
Developer Evangelist

@msarrel
matt.sarrel@imply.io

Peter Marshall
Technology Evangelist

@petermarshallio
peter.marshall@imply.io

ASSIST

The day-to-day problems people have

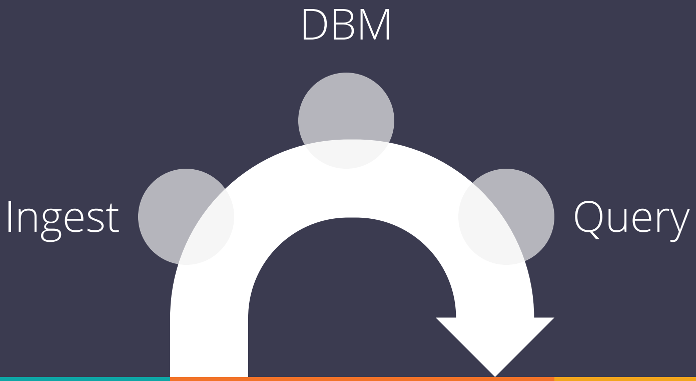
The most common issues

Potential content

Potential docs updates

Potential code changes

Archdruids!



DESIGN	DEPLOY	CREATE	STABILISE	OPERATE
Defining the data pipeline, noting all the building blocks required, noting how only how they will be realised but how and what data objects will flow through that pipeline and with what size, shape, and regularity.	Manually or with automation, assigning Apache Druid components and configurations to the infrastructure - including network services, routing and firewall configuration, encryption certificates - along with the three Druid dependencies: deep storage, Zookeeper, and a metadata database.	Using the features of Apache Druid within the pipeline to achieve the desired design.	Hardening and all those tasks you would associate with good service transition, from defining OLAs / SLAs to training and educating your target audience.	Monitoring, support and maintenance of the transitioned system to meet SLAs.

Ingest

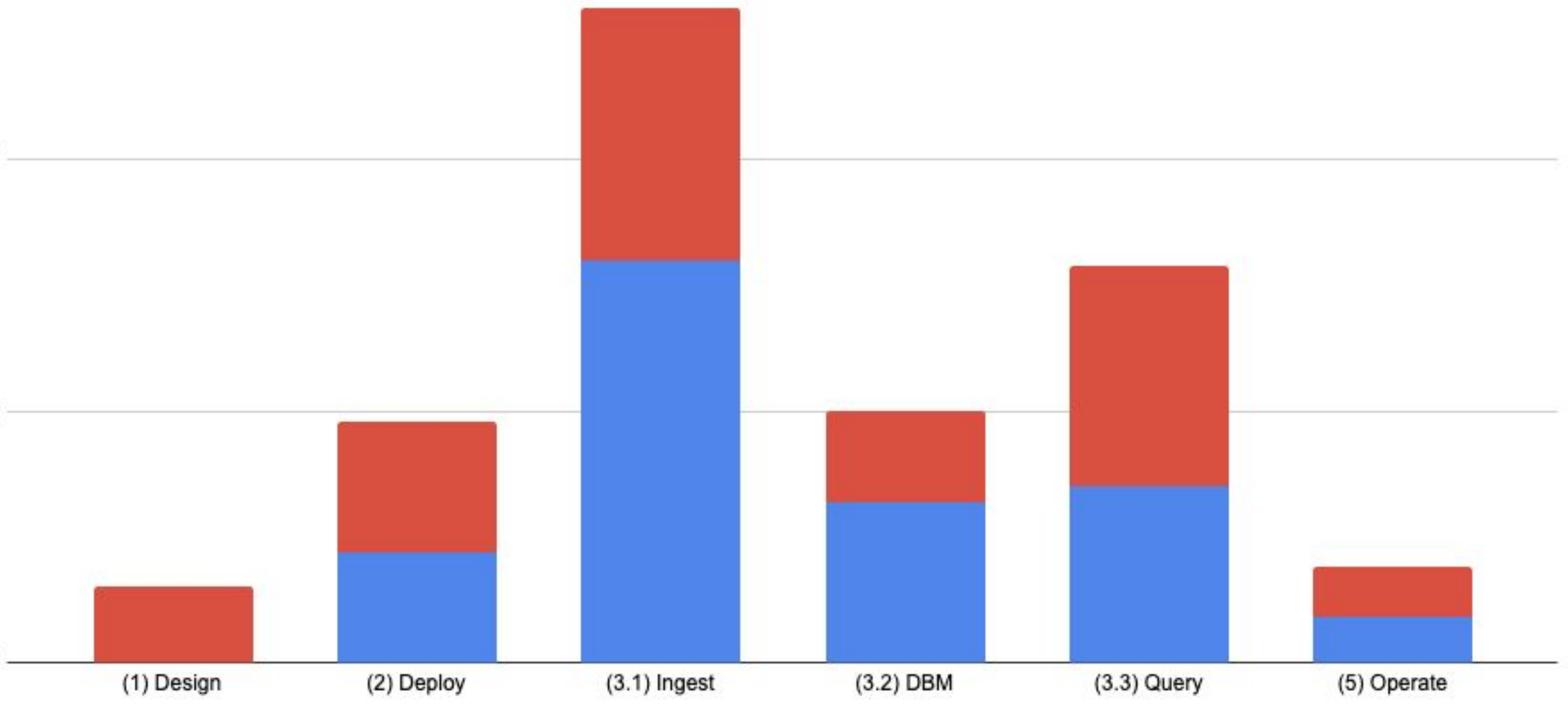
Defining ingestion tasks that will bring statistics-ready data into Druid from storage and delivery services, (including schema, parsing rules, transformation, filtering, connectivity, and tuning options) and ensuring their distributed execution, led by the **overlord**, is performant and complete.

DBM

Led by the **coordinator**, replication and distribution of the ingested data according to rules, while allowing for defragmenting (“compact”), reshaping, heating / cooling, and deleting that data.

Query

Programming SQL / Druid Native code executed by the distributed processes that are led by the **broker** service (possibly via the router process) with security applied.



DESIGN DEPLOY CREATE OPERATE

Ingest

Defining ingestion tasks that will bring statistics-ready data into Druid from storage and delivery services, (including schema, parsing rules, transformation, filtering, connectivity, and tuning options) and ensuring their distributed execution, led by the overlord, is performant and complete.

General Questions

Specifications (ingestion and compaction), and how they are written or generated

Execution of the ingestion

Inbound Integration to things like Hadoop and Kafka

DBM

Led by the coordinator, replication and distribution of the ingested data according to rules, while allowing for defragmenting ("compact"), reshaping, heating / cooling, and deleting that data.

General Questions

Deletion (kill tasks) and **distribution** of ingested data, whether that's immediately afterwards or afterwards

Any **metadata** questions, ie sys.*

Auto Compaction configuration (not the job itself - that's a spec...)

Query

Programming SQL / Druid Native code executed by the distributed processes that are led by the broker service (possibly via the router process) with security applied.

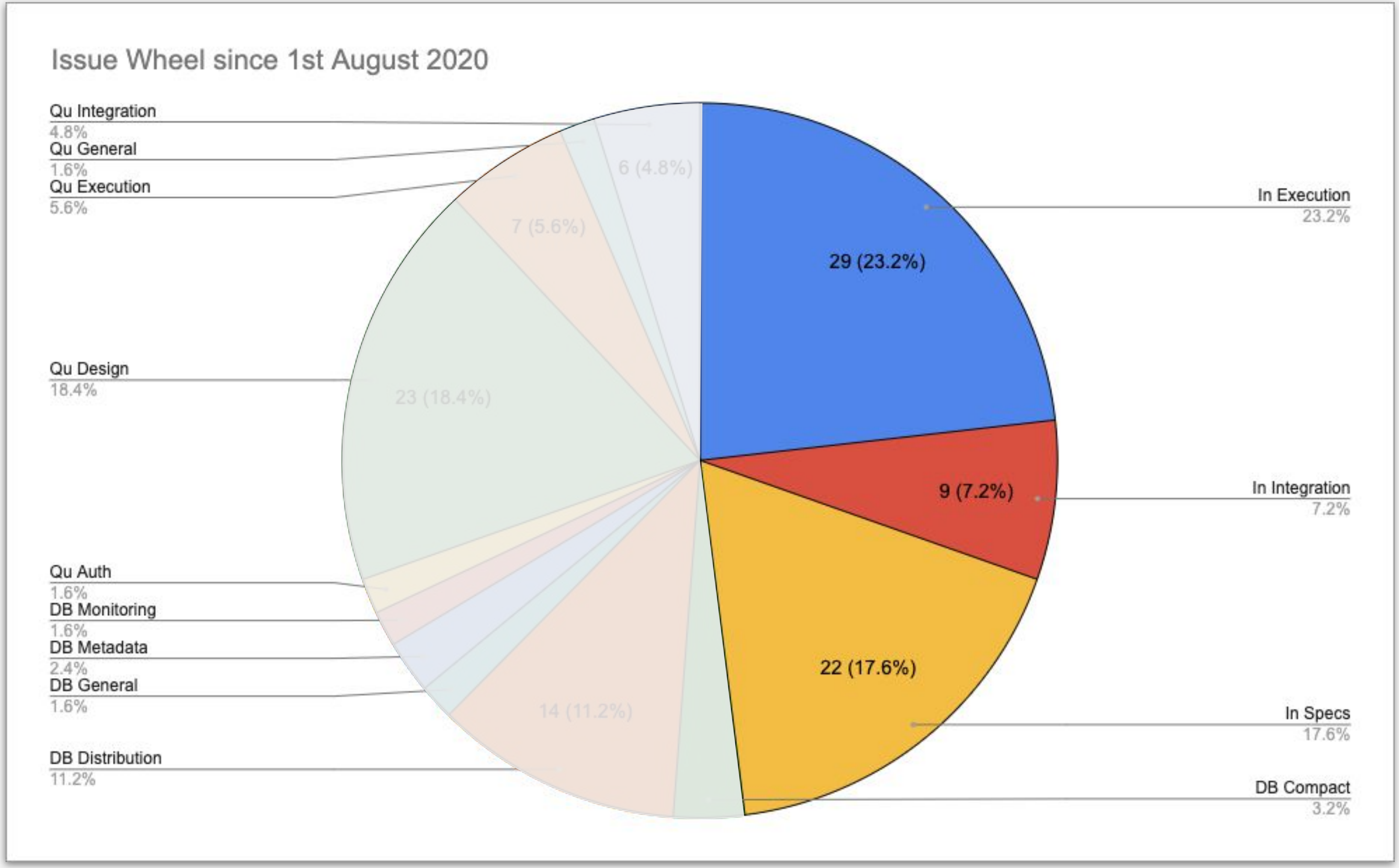
General Questions

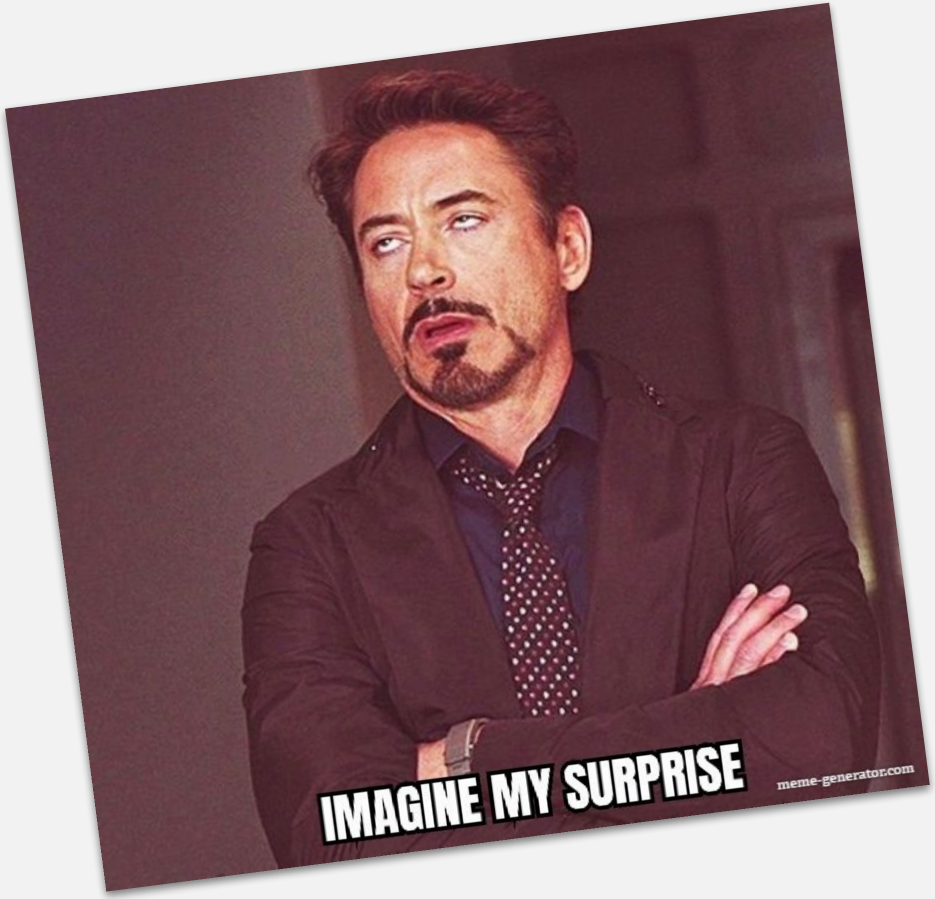
Authorisation and **Authentication** via the broker

Designing fast, effective **queries,** whether that's SQL or Native.

Execution of queries

Outbound Integration of Druid with tools like Superset


Issue Wheel since 1st August 2020
Qu Integration 4.8%
Qu General 1.6%
Qu Execution 5.6%
Qu Design 18.4%
Qu Auth 1.6%
DB Monitoring 1.6%
DB Metadata 2.4%
DB General 1.6%
DB Distribution 11.2%
In Execution 23.2%
In Integration 7.2%
In Specs 17.6%
DB Compact 3.2%
6 (4.8%)
7 (5.6%)
23 (18.4%)
14 (11.2%)
29 (23.2%)
9 (7.2%)
22 (17.6%)


IMAGINE MY SURPRISE
meme-generator.com

Examples of Ingestion Execution problems

“ingestion is not happening to druid even if the data is present in the topic.”

“compact task seems to be blocked by the index task”

“failing task with "runnerStatusCode":"WAITING"”

“Ingestion task fails with RunTime Exception during BUILD_SEGMENTS phase”

“the task is still running until the time limit specified and then is marked as FAILED”

“it seems that the throughput does not cross 1M average”
“its taking more than hour to ingest. When we triggered kill task, its taking forever”
“tips or tricks on improving ingestion performance?”

“Ingestion was throttled for [35,610] millis because persists were pending”

Examples of Ingestion Specification problems

“How to resolve for NULL values when they are coming from source table?”

“Previous sequenceNumber [289] is no longer available for partition [0].”

“Error on batch ingesting from different druid datasource”

“how to do some data formatting while handling schema changes”

“I am not seeing Druid doing any rollups”

“regexp_extract function is causing nullpointerexceptions”

“Anyone tried to hardcode the timeStamp?”

DESIGN | DEPLOY | CREATE | STABILISE | OPERATE

Don’t Walk Alone

Work with your end users to sketch out what your Druid-powered user interface is going to look like.

Grafana
imply pivot
elixir
Microsoft® .NET
R
python™
tableau SOFTWARE
looker
React
ANGULAR
jupyter
Apache Zeppelin
Metabase
Superset

Tips for Druid Design

Real-time data analysis starts with time as a key dimension.

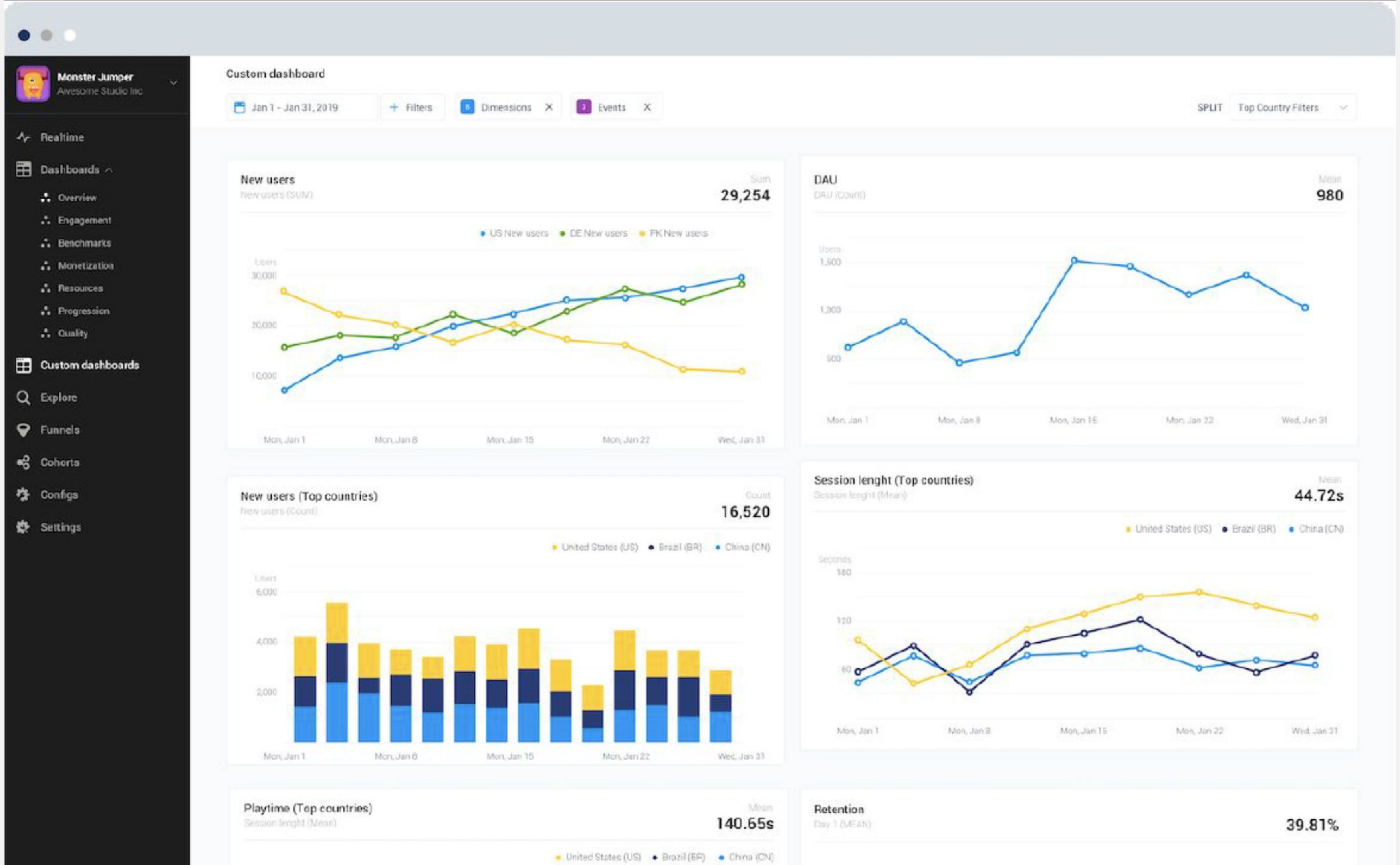
Comparisons make people think differently.

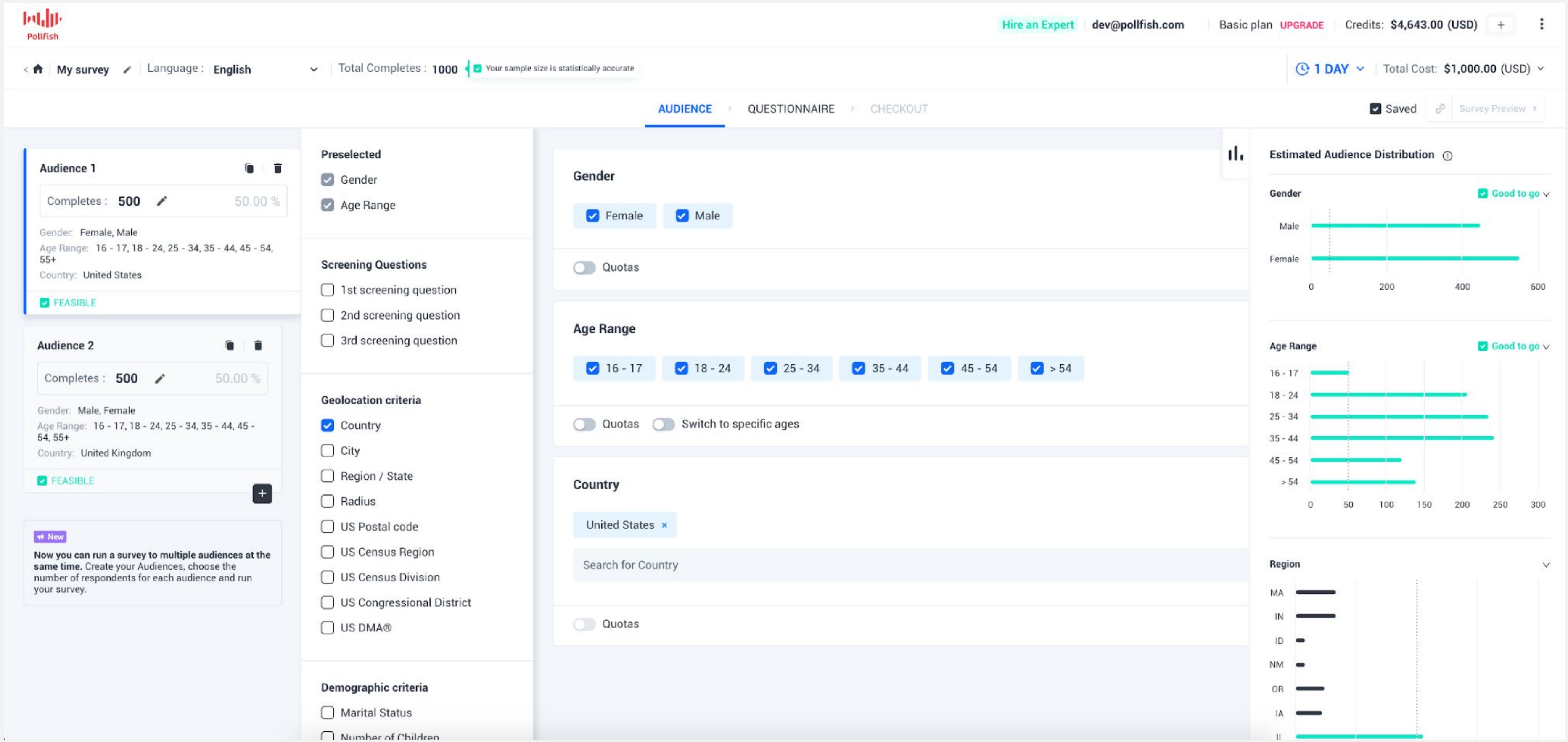
Filters make one visual cover multiple contexts.

Measures make one visual cover multiple indicators.

Create data sources focused on speed.

Create magic!


Monster Jumper
Awesome Studio Inc
Realtime
Dashboards
Overview
Engagement
Benchmarks
Monetization
Resources
Progression
Quality
Custom dashboards
Explore
Funnels
Cohorts
Configs
Settings
Custom dashboard
Jan 1 - Jan 31, 2019
Filters
Dimensions
Events
SPLIT
Top Country Filters
New users
New users (SUM)
Sum
29,254
US New users
DE New users
PK New users
DAU
DAU (Count)
Mean
980
New users (Top countries)
New users (Count)
Count
16,520
United States (US)
Brazil (BR)
China (CN)
Session lenght (Top countries)
Session lenght (Mean)
Mean
44.72s
Playtime (Top countries)
Session lenght (Mean)
Mean
140.65s
Retention
Day 1 (MEAN)
39.81%
Mon, Jan 1
Mon, Jan 8
Mon, Jan 15
Mon, Jan 22
Wed, Jan 31


Pollfish

Hire an Expert | dev@pollfish.com | Basic plan UPGRADE | Credits: $4,643.00 (USD) +

My survey | Language : English | Total Completes : 1000 | ✓ Your sample size is statistically accurate

1 DAY | Total Cost: $1,000.00 (USD)

AUDIENCE › QUESTIONNAIRE › CHECKOUT

✓ Saved | Survey Preview ›

Audience 1

Completes : 500 | 50.00 %

Gender: Female, Male

Age Range: 16 - 17, 18 - 24, 25 - 34, 35 - 44, 45 - 54, 55+

Country: United States

✓ FEASIBLE

Audience 2

Completes : 500 | 50.00 %

Gender: Male, Female

Age Range: 16 - 17, 18 - 24, 25 - 34, 35 - 44, 45 - 54, 55+

Country: United Kingdom

✓ FEASIBLE

New

Now you can run a survey to multiple audiences at the same time. Create your Audiences, choose the number of respondents for each audience and run your survey.

Preselected

- [x] Gender
- [x] Age Range

Screening Questions

- [ ] 1st screening question
- [ ] 2nd screening question
- [ ] 3rd screening question

Geolocation criteria

- [x] Country
- [ ] City
- [ ] Region / State
- [ ] Radius
- [ ] US Postal code
- [ ] US Census Region
- [ ] US Census Division
- [ ] US Congressional District
- [ ] US DMA®

Demographic criteria

- [ ] Marital Status
- [ ] Number of Children

Gender

- [x] Female
- [x] Male

Quotas

Age Range

- [x] 16 - 17
- [x] 18 - 24
- [x] 25 - 34
- [x] 35 - 44
- [x] 45 - 54
- [x] > 54

Quotas | Switch to specific ages

Country

United States ×

Search for Country

Quotas

Estimated Audience Distribution

DESIGN DEPLOY CREATE STABILISE OPERATE

Druid != Island

Think about how and what you will deploy onto your infrastructure, especially Druid's dependencies

imply

Specialised collectors
Applications & APIs
Machine & Human Data
Environmental Sensors
Stream & Bulk Repositories

Consolidation
Enrichment
Transformation & Stripping
Verification & Validation
Filtering & Sorting

Feature & Structure Discovery
Segmentation & Classification
Recommendation
Prediction & Anomaly Detection
Statistical Calculations

Real-time Analytics
BI Reporting & Dashboards
Buses, Queues & Databases
Search & Filtering UIs
Applications & APIs

Production

Delivery

GUARANTEED DELIVERY
POSSIBLY STORING FOR A LIMITED PERIOD

PULSAR
amazon KINESIS
kafka

Processing

ENRICHMENT AND CLEANSING
REDUCING IN THE SIZE OF THE INPUT

TEZ
samza
APACHE STORM
APACHE Spark

Storage

RETENTION FOR LATER
FOR OPERATIONS & FOR RESEARCH

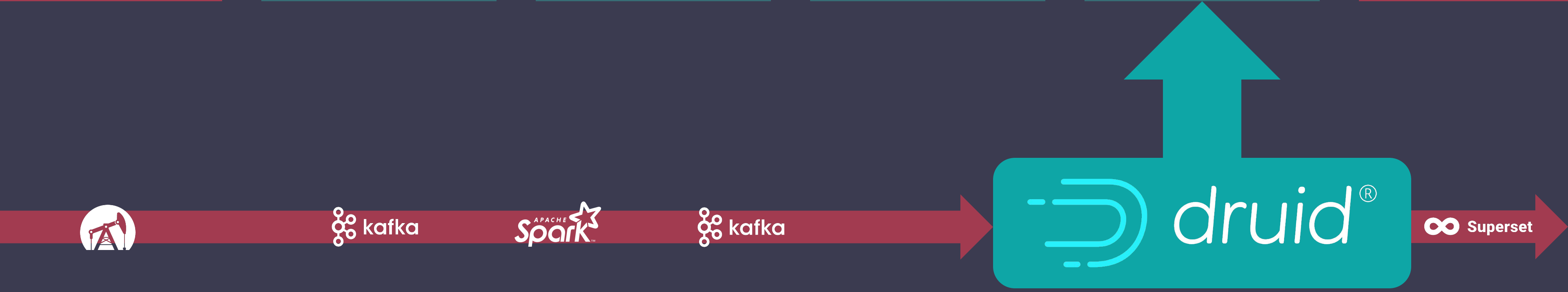
Cassandra
APACHE HBASE
Lucene
Microsoft SQL Server
hadoop HDFS
ORACLE
MySQL
PostgreSQL

Query

EASY AND SPEEDY ANALYSIS
BY STATS, LEARNING, SEARCH...

APACHE DRILL
elasticsearch
Apache Solr
HIVE

Delivery



Jump Right Er... no.

Configure JRE properties well - especially heaps - and choose the right hardware

Take time to read the tuning guide

druid.apache.org/docs/latest/operations/basic-cluster-tuning.html
druid.apache.org/docs/latest/configuration/index.html#jvm-configuration-best-practices

Historical Runtime Properties

```
druid.server.http.numThreads
druid.processing.buffer.sizeBytes
druid.processing.numMergeBuffers
druid.processing.numThreads
druid.server.maxSize
druid.historical.cache.useCache
druid.historical.cache.populateCache
druid.cache.sizeInBytes
```

Historical Java Properties

```
-Xms
-Xmx
-XX:MaxDirectMemorySize
```

Middle Manager Runtime Properties

```
druid.worker.capacity
druid.server.http.numThreads
druid.indexer.fork.property.druid.processing.numMergeBuffers
druid.indexer.fork.property.druid.processing.buffer.sizeBytes
druid.indexer.fork.property.druid.processing.numThreads
druid.processing.buffer.sizeBytes
druid.processing.numMergeBuffers
druid.processing.numThreads
```

MiddleManager Java Properties

```
-Xms
-Xmx
```

MiddleManager Peon Java Properties

```
-Xms
-Xmx
-XX:MaxDirectMemorySize
```

Broker Runtime Properties

```
druid.server.http.numThreads
druid.broker.http.numConnections
druid.broker.cache.useCache
druid.broker.cache.populateCache
druid.processing.buffer.sizeBytes
druid.processing.numThreads
druid.processing.numMergeBuffers
druid.broker.http.maxQueuedBytes
```

Broker / Coordinator / Overlord Java Properties

```
-Xms
-Xmx
-XX:MaxDirectMemorySize
```

Stay Connected

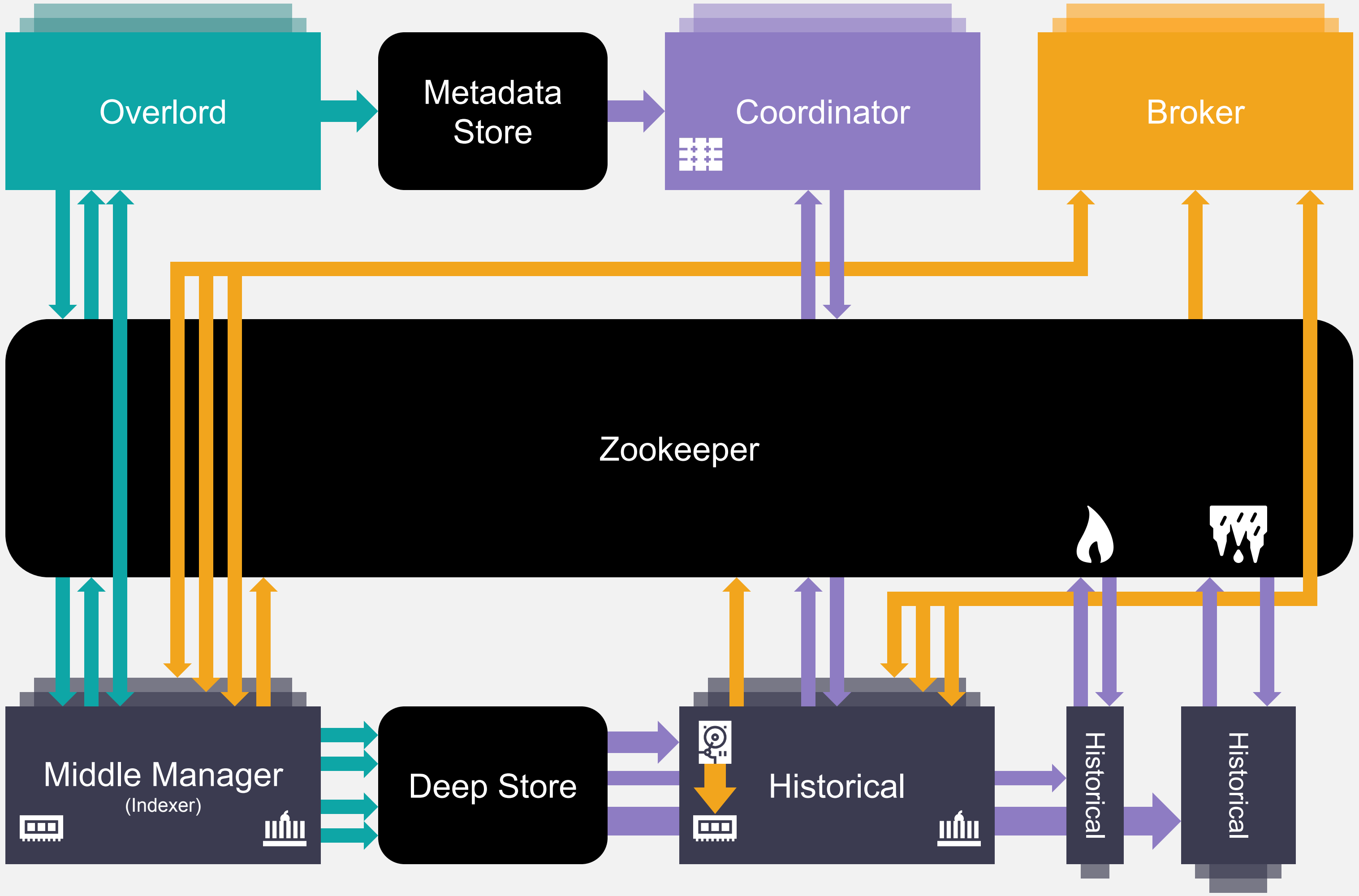
Druid is a highly distributed, loosely coupled system on purpose.

Care for your interprocess communication systems and paths: especially Zookeeper and Http

druid.apache.org/docs/latest/dependencies/zookeeper.html
druid.apache.org/docs/latest/configuration/index.html#zookeeper

Know Your Team

Get to know the core distributed collaborations of Apache Druid


Overlord
Metadata Store
Coordinator
Broker
Zookeeper
Middle Manager
(Indexer)
Deep Store
Historical
Historical
Historical

Love Your Log

Get to know the logs.

For ingestion, particularly the overlord, middle manager and its tasks.

For what happens next, particularly the coordinator and historicals.

K.I.S.S.

Be agile: set up a lab, start simple and start small, working up to perfection

Some Ingestion Spec Tips

Create a target query list

Understand which source data columns you will need at ingestion time (filtering, transformation, lookup) and which are used at query time (filtering, sorting, calculations, grouping, bucketing, aggregations)

Set up your dimension spec and execute queries, recording query performance

Explore what other queries (Time Series, Group By, Top N) you could do with the data

Add more subtasks and monitor the payloads

Add more data and check the lag

Use ingestion-time filter to eke out performance and storage efficiencies

Use transforms to replace or create data closer to the queries that people will execute

Use time granularity and roll-up to generate metrics and datasketches (set, quantile, and cardinality)

DESIGN DEPLOY CREATE STABILISE OPERATE

Digest the Specifics

Learn ingestion specifications in detail through your own exploration and experimentation, from the docs, and from the community

Understand Segments

Historicals serve all used segments, and deep storage stores them

Query time relates directly to segment size: lots of small segments means lots of small query tasks

Segments are tracked in master nodes and registered in the metadata DB

Segment Tips & Tricks

Filter rows and think carefully about what dimensions you need

Use different **segment** granularities and row maximums to control the number of segments generated

Apply time bucketting with **query granularity** and **roll-up**

Think about **tiering** your historicals using drop and load rules

Consider not just initial ingestion but on-going **re-indexing**

Never forget **compaction!**

Check local (`maxBytesInMemory`, `maxRowsInMemory`, `intermediatePersistPeriod`) and deep storage (`maxRowsPerSegment`, `maxTotalRows`, `intermediateHandoffPeriod, taskDuration`) **persists**

Ask Us Anything!

Find other people in the community
that have had the same issue as you

Learn from the best!

Find other people in the community
who have walked your walk!

COMMUNITY@IMPLY.IO

druid.apache.org/community

Get Meta

Collect metrics and understand how your work affects them

imply

Metrics & Measures

Infrastructure

Host
Druid Service
Instance Type
Imply Version
Druid Version

Tasks

Task Id
Task Status

Query Data

Data Source	Num Metrics
Query Type	Num Complex Metrics
Native / Query ID	Interval
Successful?	Duration
Identity	Filters?
Context	Remote Address

Ingestion

Events Processed
Unparseable Events
Thrown Away Events
Output Rows
Persists
Total Back Pressure
Message Gap
Kafka Lag

Infrastructure

Memory Used
Maximum Memory Used
Garbage Collections
Total / Average GC Time
Total CPU Time
User CPU Time
CPU Wait Time
CPU System Use
Avg Jetty Connections
Min / Max Jetty Conns

Query Cache

Hit Rate
Hits
Misses
Timeouts
Errors
Size
Number of Entries
Average Entry Size
Evictions

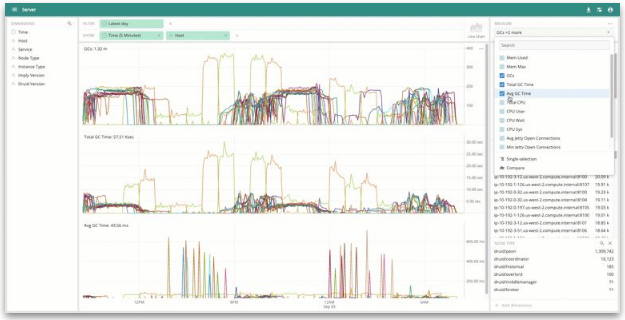
Query Patterns

Query Count	Avg Return Size
Average Query Time	Total CPU Time
98%ile Query Time	Subquery Count
Max Query Time	Avg Subquery Time

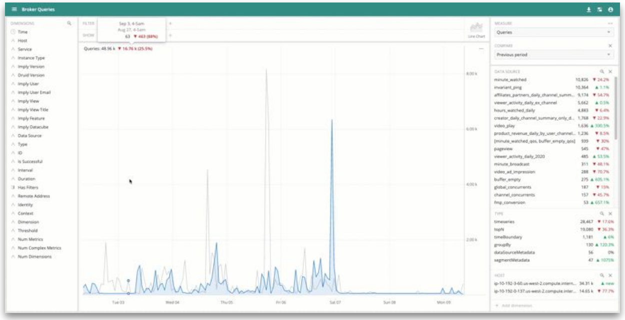
https://druid.apache.org/docs/latest/operations/metrics.html

imply clarity

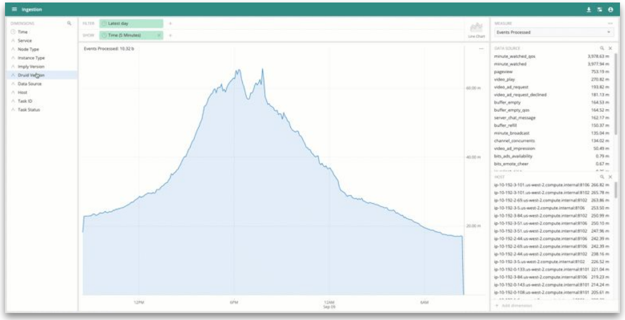
Infrastructure



Use & Experience



Ingestion



Inform **capacity planning**
Isolate potential execution **bottlenecks**
Check and investigate cluster **performance**
Flatten the **learning curve** for running Druid at scale

How can we all help each other?

1

Come with us, join us...

Join ASF Slack and the Google User Group, say hi, give people (socially distanced) hugs - and link back to the docs

2

Tippy-Tappy-Typey-Type-Type

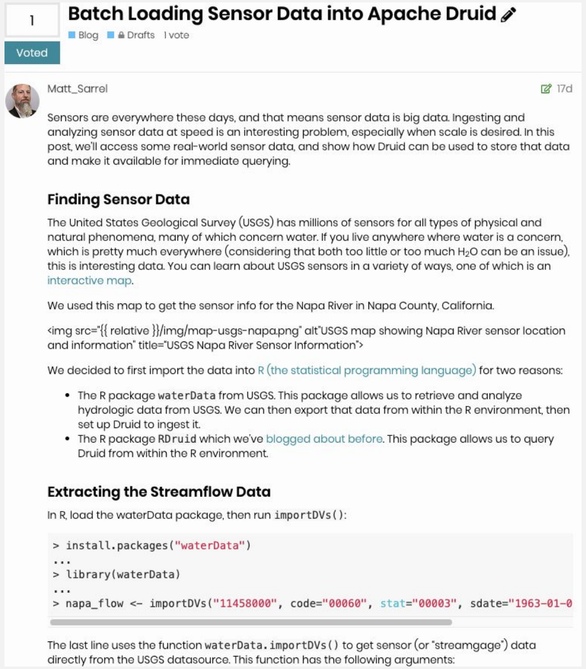
Blog helpful tips and tricks and walkthroughs of your own ingestion integrations, specifications, and execution configurations. Contribute code and doc updates :-D

3

Make Pretty Slides

Take part in Ask Me Anything, Town Hall, and Druid meetups about ingestion.

Walkthroughs



1 Voted

Batch Loading Sensor Data into Apache Druid

Blog · Drafts · 1 vote

Matt_Sarrel 17d

Sensors are everywhere these days, and that means sensor data is big data. Ingesting and analyzing sensor data at speed is an interesting problem, especially when scale is desired. In this post, we'll access some real-world sensor data, and show how Druid can be used to store that data and make it available for immediate querying.

Finding Sensor Data

The United States Geological Survey (USGS) has millions of sensors for all types of physical and natural phenomena, many of which concern water. If you live anywhere where water is a concern, which is pretty much everywhere (considering that both too little or too much H_2O can be an issue), this is interesting data. You can learn about USGS sensors in a variety of ways, one of which is an interactive map.

We used this map to get the sensor info for the Napa River in Napa County, California.

<img src="{{ relative }}/img/map-usgs-napa.png" alt"USGS map showing Napa River sensor location and information" title="USGS Napa River Sensor Information">

We decided to first import the data into R (the statistical programming language) for two reasons:

- The R package `waterData` from USGS. This package allows us to retrieve and analyze hydrologic data from USGS. We can then export that data from within the R environment, then set up Druid to ingest it.
- The R package `RDruid` which we've blogged about before. This package allows us to query Druid from within the R environment.

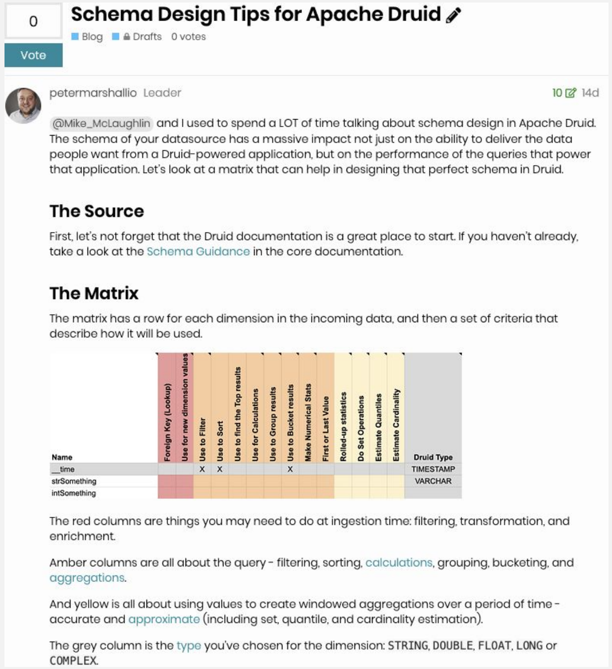
Extracting the Streamflow Data

In R, load the waterData package, then run `importDVs()`:

```
> install.packages("waterData")
...
> library(waterData)
...
> napa_flow <- importDVs("11458000", code="00060", stat="00003", sdate="1963-01-0
```

The last line uses the function `waterData.importDVs()` to get sensor (or "streamgage") data directly from the USGS datasource. This function has the following arguments:

Tips & Tricks



0 Vote

Schema Design Tips for Apache Druid

Blog · Drafts · 0 votes

petermarshallio Leader 10 14d

@Mike_McLaughlin and I used to spend a LOT of time talking about schema design in Apache Druid. The schema of your datasource has a massive impact not just on the ability to deliver the data people want from a Druid-powered application, but on the performance of the queries that power that application. Let's look at a matrix that can help in designing that perfect schema in Druid.

The Source

First, let's not forget that the Druid documentation is a great place to start. If you haven't already, take a look at the Schema Guidance in the core documentation.

The Matrix

The matrix has a row for each dimension in the incoming data, and then a set of criteria that describe how it will be used.

Name	Foreign Key (Lookup)	Use for new dimension values	Use to Filter	Use to Sort	Use to find the Top results	Use for Calculations	Use to Group results	Use to Bucket results	Make Numerical Stats	First or Last Value	Rolled-up statistics	Do Set Operations	Estimate Quantiles	Estimate Cardinality	Druid Type
__time			X	X				X							TIMESTAMP
strSomething															VARCHAR
intSomething															

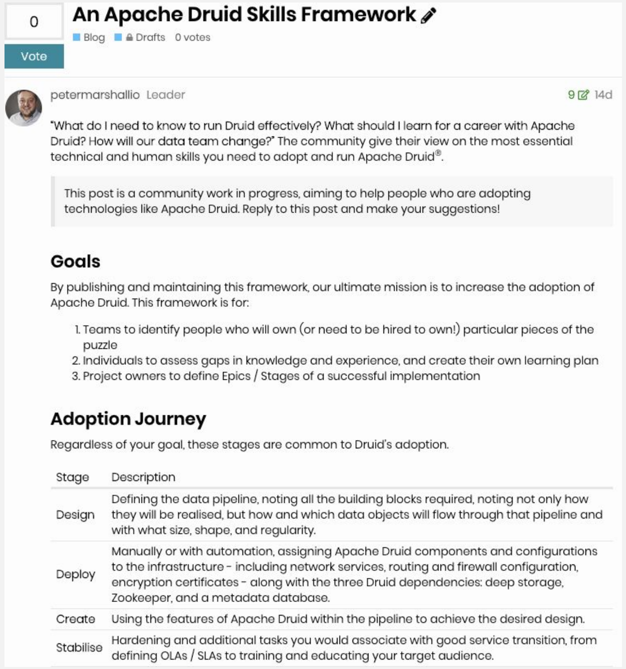
The red columns are things you may need to do at ingestion time: filtering, transformation, and enrichment.

Amber columns are all about the query - filtering, sorting, calculations, grouping, bucketing, and aggregations.

And yellow is all about using values to create windowed aggregations over a period of time - accurate and approximate (including set, quantile, and cardinality estimation).

The grey column is the type you've chosen for the dimension: `STRING`, `DOUBLE`, `FLOAT`, `LONG` or `COMPLEX`.

druid-DDCSO Content



0 Vote

An Apache Druid Skills Framework

Blog · Drafts · 0 votes

petermarshallio Leader 9 14d

"What do I need to know to run Druid effectively? What should I learn for a career with Apache Druid? How will our data team change?" The community give their view on the most essential technical and human skills you need to adopt and run Apache Druid®.

> This post is a community work in progress, aiming to help people who are adopting technologies like Apache Druid. Reply to this post and make your suggestions!

Goals

By publishing and maintaining this framework, our ultimate mission is to increase the adoption of Apache Druid. This framework is for:

1. Teams to identify people who will own (or need to be hired to own!) particular pieces of the puzzle
2. Individuals to assess gaps in knowledge and experience, and create their own learning plan
3. Project owners to define Epics / Stages of a successful implementation

Adoption Journey

Regardless of your goal, these stages are common to Druid's adoption.

Stage	Description
Design	Defining the data pipeline, noting all the building blocks required, noting not only how they will be realised, but how and which data objects will flow through that pipeline and with what size, shape, and regularity.
Deploy	Manually or with automation, assigning Apache Druid components and configurations to the infrastructure - including network services, routing and firewall configuration, encryption certificates - along with the three Druid dependencies: deep storage, Zookeeper, and a metadata database.
Create	Using the features of Apache Druid within the pipeline to achieve the desired design.
Stabilise	Hardening and additional tasks you would associate with good service transition, from defining OLAs / SLAs to training and educating your target audience.

imply

Community Issues

Tips & Tricks

Call to Action

Questions & Answers