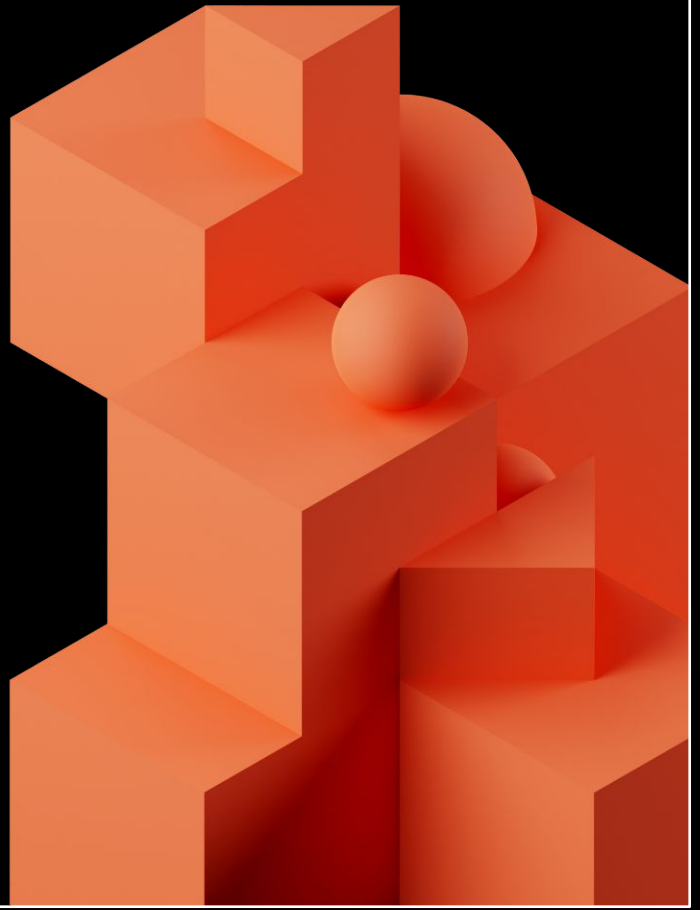
DATA+AI SUMMIT
BY databricks

Rec Room 🤝 RudderStack

How Rec Room Processes Billions of Events Per Day with Databricks and RudderStack



Databricks
2023

Albert Hu

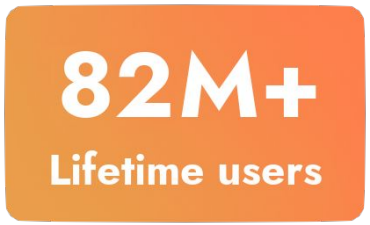
Senior Analytics Engineer
Rec Room

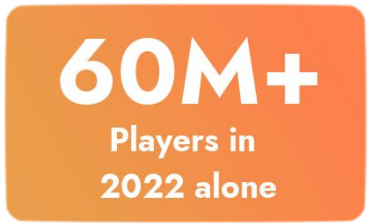
Lewis Mbae

Head of Customer Engineering
RudderStack

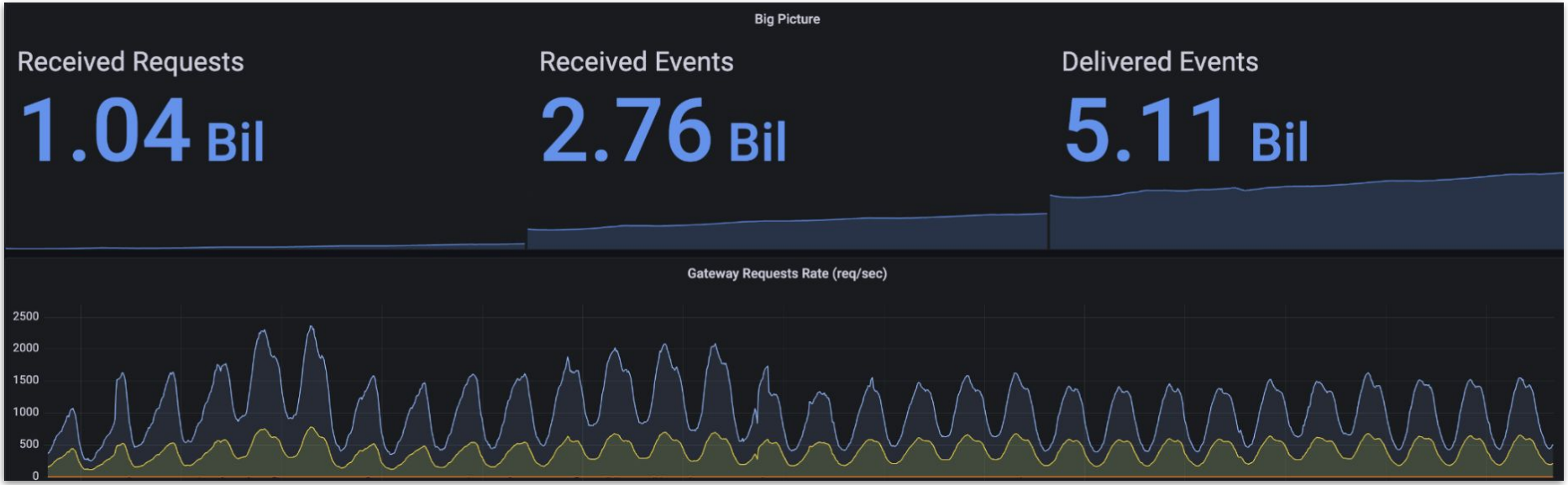
About Rec Room

- Rec Room is the best place to build and play games together.
- With more than 80M lifetime users, you can party up with friends from all around the world to play, hang out, explore MILLIONS of player-created rooms, or build something new!
- Founded in 2016





More Users, More Data



Making Impact with Data at Rec Room

Data-driven culture powered by high data volume and quick adoption

A/B Test as many decisions and features as possible

Recommend new, interesting rooms and items to players

Share metrics with Creators to help grow their player base

Rec Room's Data Challenges

Disparate data sources


unity

javascript

microsoft .net


webhook


flutter

Disparate data destinations


databricks


amplitude


facebook


azure event hubs


statsig

Multiple environments

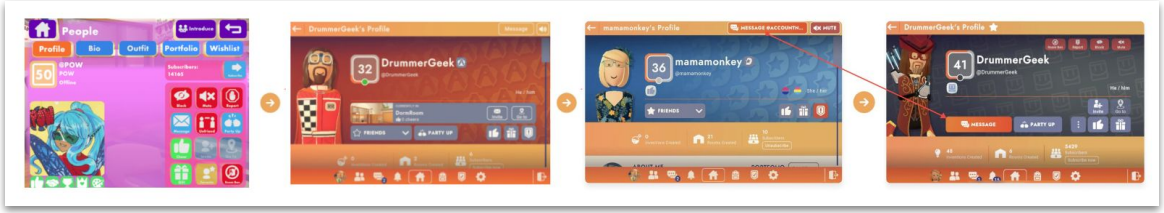

dev

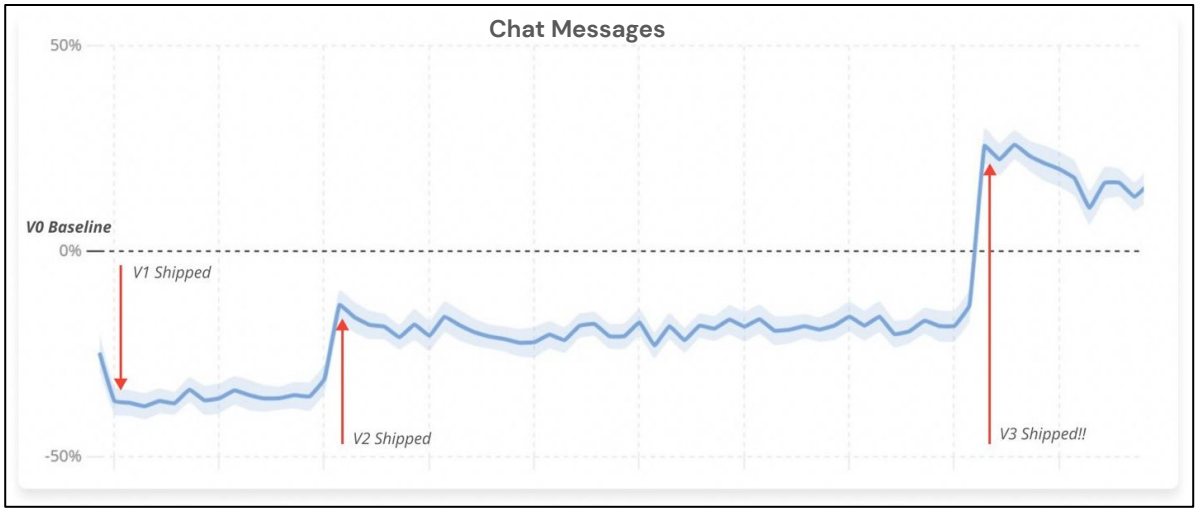

prod


150+ experiments
launched as of Q2 2023
Started A/B testing March 2022

One A/B test, big impact

From a negative to positive 25% change in chat messages between players



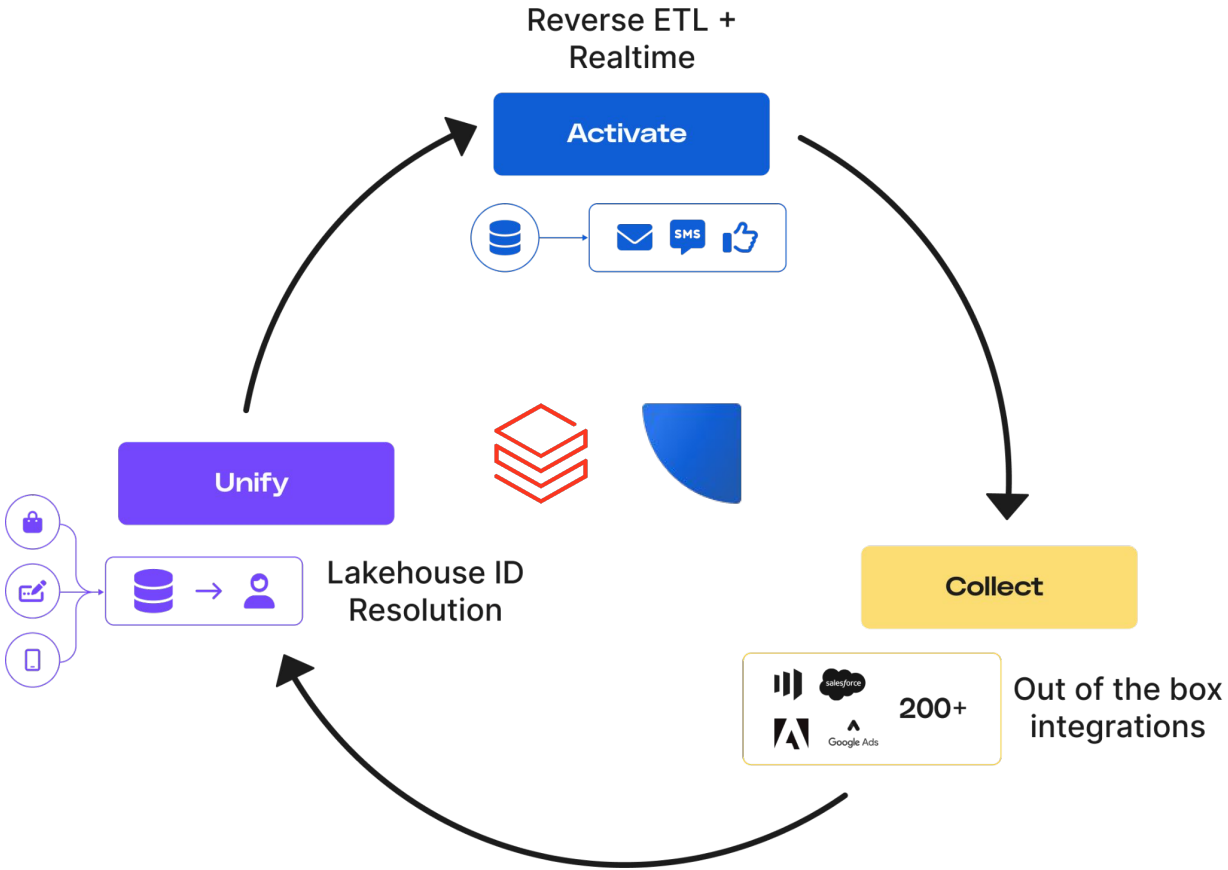


About RudderStack

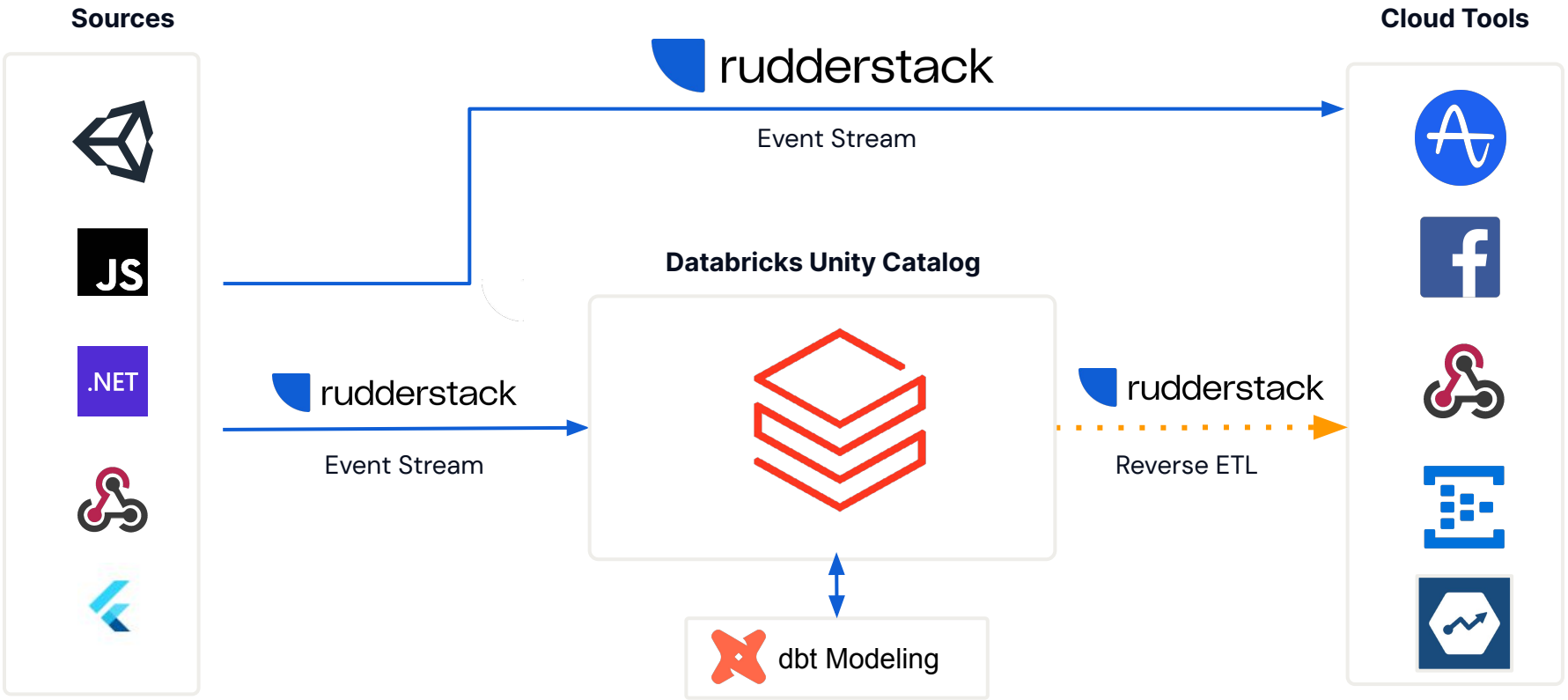
- RudderStack is the leading **lakehouse native customer data platform**.
- RudderStack runs on top of your lakehouse and does not store data, alleviating security concerns, reducing costs, and unlocking the value of your lakehouse investment
- Founded in 2019

allbirds stripe Hinge Crate&Barrel Wynn LAS VEGAS Anheuser-Busch

The Data Activation Lifecycle



Rec Room Data Architecture

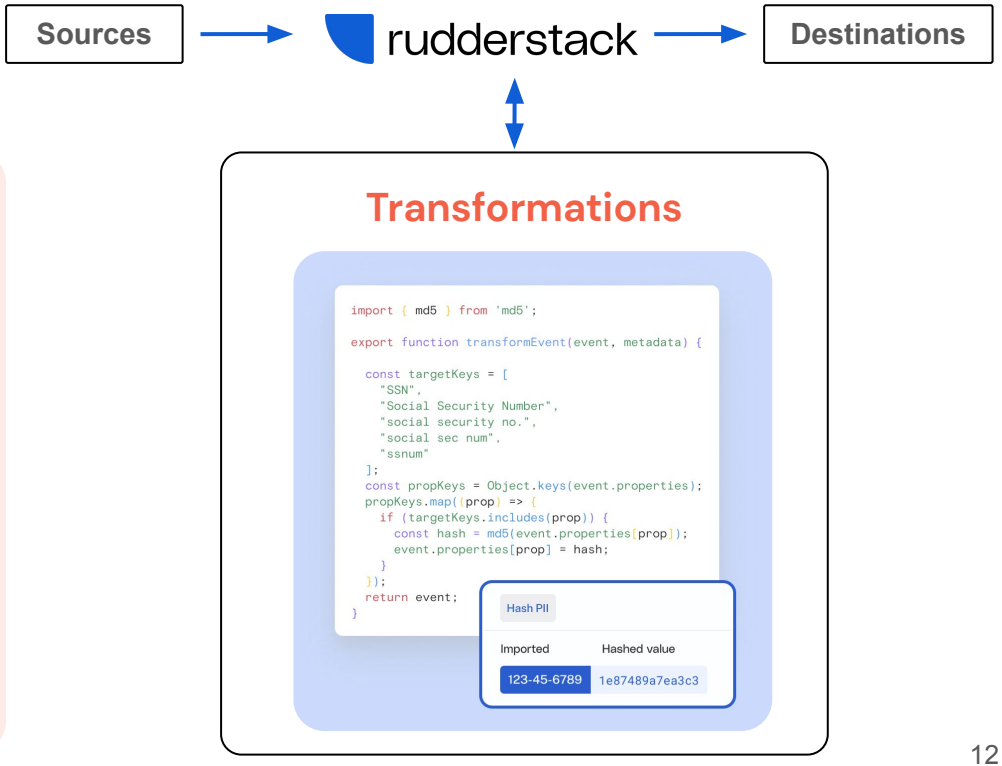


What is RudderStack Transformations?

Transformations lets users customize event data in real-time using JavaScript or Python.

Benefits:

- Guarantee high quality data
- Secure and build data trust
- Quickly adapt to new business needs



How Rec Room uses Transformations

Filtering events by destination

Cleaning and enriching event data before it lands in the destination

Testing Transformations in dev environments before shipping to production

Tips for RudderStack Transformations

Treat RudderStack Transformations like any other code

Define functions that are used repeatedly to keep code DRY

Version Control and PR process

Automate testing using unit tests via Github Actions

Simplifying Databrick's Data Processing

RudderStack's clean lakehouse data model enables painless data processing

identities Table	
user_id:_	varchar
anonymous_id:_	varchar
email:_	varchar
first_name:_	varchar
last_name:_	varchar
context_<prop>:_	varchar
timestamp:_	timestamp

users Table	
user_id:_	varchar
email:_	varchar
first_name:_	varchar
last_name:_	varchar
context_<prop>:_	varchar
timestamp:_	timestamp

tracks Table	
user_id:_	varchar
anonymous_id:_	varchar
event:_	varchar
context_<prop>:_	varchar
timestamp:_	timestamp

Simplifying Databricks' Data Processing

Using a medallion data architecture to power multiple use-cases

Bronze
Raw ingestion

- Copied from blob storage and inserted hourly
- No table stats collection

Silver
Filtered, cleaned, augmented

- Merge w/ 2-day lookback window to process **late arriving data**
- Partition by **server received date**
- Table stats collected Daily

Gold
Business-facing

- Append by **server received date**
- Partition by **event date**
- Table stats collected Weekly

Configure tblproperties sooner than later

- Configure appropriate **tblproperties** during the pipeline building process

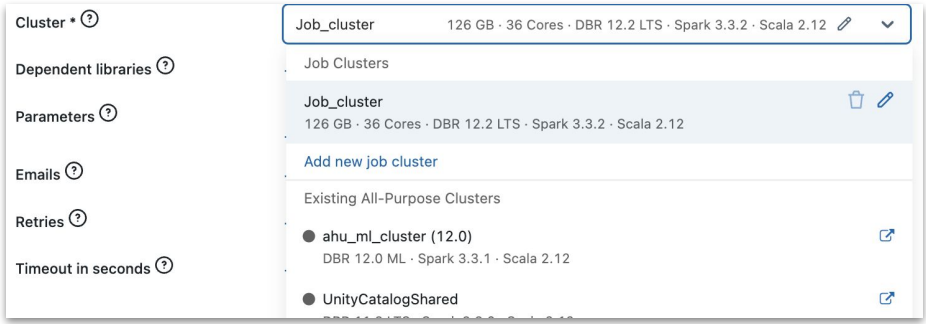
- Structure table columns to optimize **z-ordering** and take advantage of **data skipping** by data type: keys/numericals on the left and strings on the right

```
ALTER TABLE example_table
SET TBLPROPERTIES (
  'delta.columnMapping.mode' = 'name',
  'delta.minReaderVersion' = '2',
  'delta.minWriterVersion' = '5',
  'delta.dataSkippingNumIndexedCols' = 40
  );
```

Choose the right compute

Choose the appropriate compute based on **frequency** and **processing time (including cluster startup)**

- For **Scheduled automated jobs** check if **Job** vs. **All-Purpose Cluster** are appropriate (3x cost difference)
- For **SQL-only job** , try **Serverless SQL Warehouses**

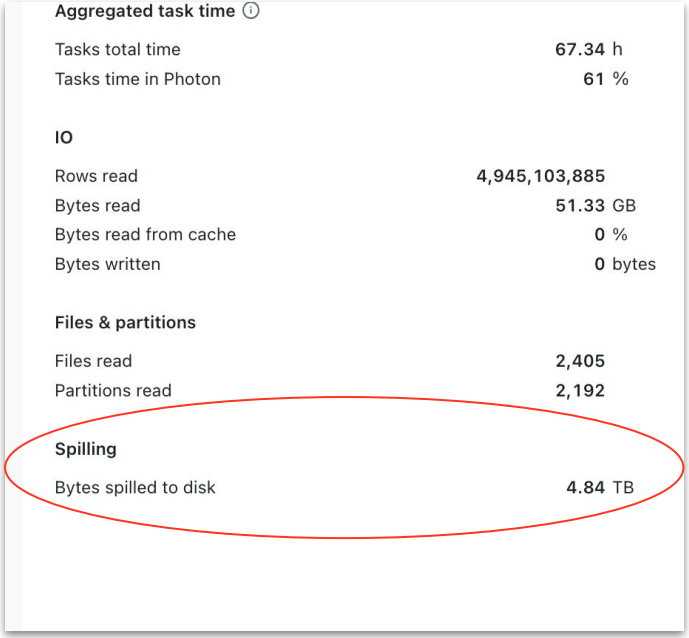


Optimize compute cost over time

Map workloads to compute that optimizes **cost over time**

- Selecting the smallest compute is not always cheapest (and vice versa)

- Find clues workloads might not be compatible with compute such as if **bytes are spilling to disk**



Choosing a partition

Query the **DeltaLog** to understand if a table is partitioned appropriately

- Table Size **>1TB**
- Size of each partition is **1GB+**
- Field is low cardinality and will be used for filtering and merge operations

```
val partitions = meta
  .select(explode($"partitionValues"))
  .select($"key")
  .distinct
  .map(f=>f.getString(0))
  .collect()
  .toList

val partitionsClause = partitions.map(x => col(s"partitionValues.${x}"))

display(
  meta.groupBy(partitionsClause:_*)
    .agg(sum("size").as("size"))
    .withColumn("size_in_mb", $"size"/lit(bytes_to_mb_factor))
    .withColumn("size_in_gb", $"size"/lit(bytes_to_gb_factor))
)
```

▸ (5) Spark Jobs

Table +

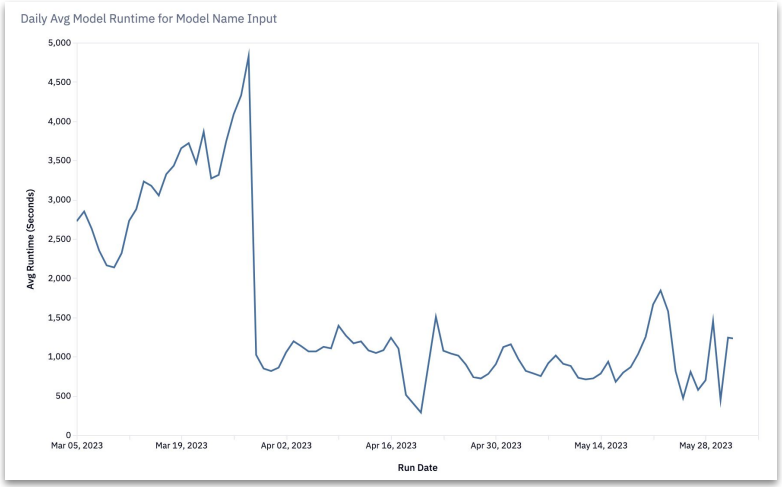
	created_date_utc	size	size_in_mb	size_in_gb
1	2023-06-04	23224942825	22149.031472206116	21.629913547076285
2	2023-06-03	23141441690	22069.398584365845	21.55214705504477
3	2023-06-02	20062565059	19133.15301799774	18.68471974413842
4	2023-06-01	16708152819	15934.136218070984	15.560679900459945
5	2023-05-31	14776246804	14091.727069854736	13.761452216655016
6	2023-05-30	14721323634	14039.348253250122	13.710301028564572
7	2023-05-29	18027145069	17192.025250434875	16.789087158627808

Prune Partitions to Optimize Merge

Rewrite the least number of files

- Filter on the partition during merge

```
MERGE INTO sources.rudderstack_db.source_rudderstack_db__events AS d USING
(SELECT * FROM source_rudderstack_db__events__dbt_tmp
) AS s
  ON d.message_id = s.message_id
  AND d.server_received_date_utc BETWEEN '2023-06-02' AND '2023-06-04'
WHEN MATCHED THEN UPDATE SET *
WHEN NOT MATCHED THEN INSERT *
```



3x time difference after partition pruning

Maximize Photon Runtime

If you have Photon enabled, **maximize the % of tasks using Photon**

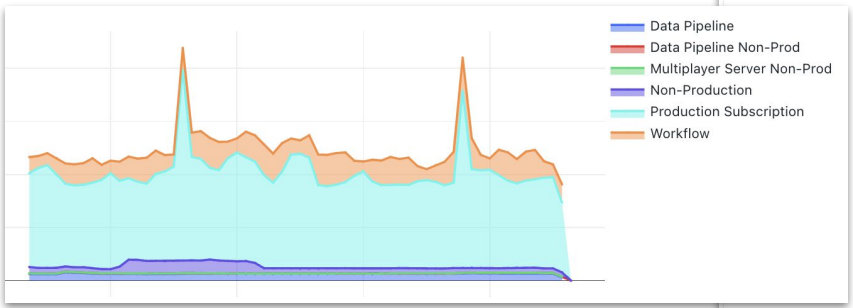
- Query performance can be dramatically faster on Photon (2x+)
- Check if queries are using **Photon-compatible functions**

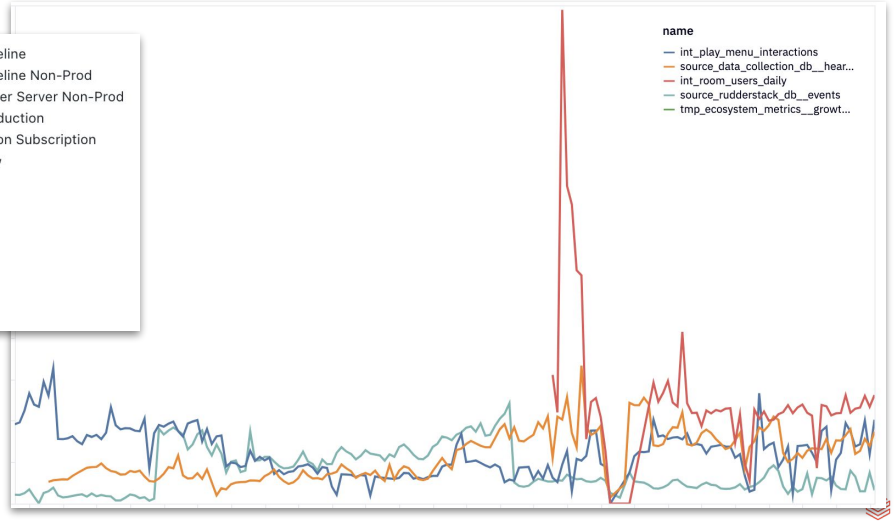
```
select
    user_id
    , anonymous_id
from
    core.events.unified_events
where
    created_date_utc >= DATE_SUB(current_date, 100)
qualify row_number() over (partition by user_id order by
created_at_utc desc) = 1
limit 1000


select
    user_id
    , max_by(anonymous_id, created_at_utc) as anonymous_id
from
    core.events.unified_events
where
    created_date_utc >= date_sub(current_date, 100)
group by 1
limit 1000
```

Measure and Meet

- Analyze job metadata and create dashboards to measure pipeline processing time. **Translate time into dollars**
- Create team rituals to review and understand drivers behind trends





RudderStack + Databricks: Unlocking value benefits for Rec Room

Clean, high-quality data

- With RudderStack, we can **collect data once and unify** it across our entire stack
- Rudderstack is the **backbone of A/B testing the user experience**, providing us product insights as well as additional data on user preferences

Seamless integrations

- Engineering teams are **spending more time on product features** and less time troubleshooting integrations
- Transformations enable the team to **quickly make changes** to existing events

Unlocking business outcomes

- A/B testing (**150+ experiments** executed through Statsig as of Q2 2023)
- **Clean data** for product analytics in Amplitude
- **Recommendation models** to improve Discovery

What's Next for RecRoom?

- Enable even more **personalized** recommendations for our players
- Share new metrics and datasets with our **Creator Community**
- Create tools to **automate the optimization decisions** needed to scale the lakehouse

What's Next for RudderStack?

- Build **unified 360 view** of a customer on top of the lakehouse to power more impactful use-cases
- Bolster **data governance** offering for collection of high quality first-party data
- Provide **more out-of-the box integrations** for our customers
- Unlock **real-time use cases** with near real-time sync to Databricks

Thank you for Attending!

Albert Hu

ahu@recroom.com

Lewis Mbae

lewis@rudderstrack.com