DuneSQL
A query engine for Blockchain data

15 June 2023

Jonas Irgens Kylling & Miguel Filipe

Miguel Mascarenhas Filipe

- Love databases since University
- Some early experience in HPC
- At AWS, part of the DynamoDB launch team
- At Skype worked on Distributed Timeseries DB
- Working in Startups since 2015
- Principal Engineer at Dune

Jonas Irgens Kylling

- PhD in Mathematics
- Worked on a Timeseries database at Cognite
- Building data platform at Dune

Agenda

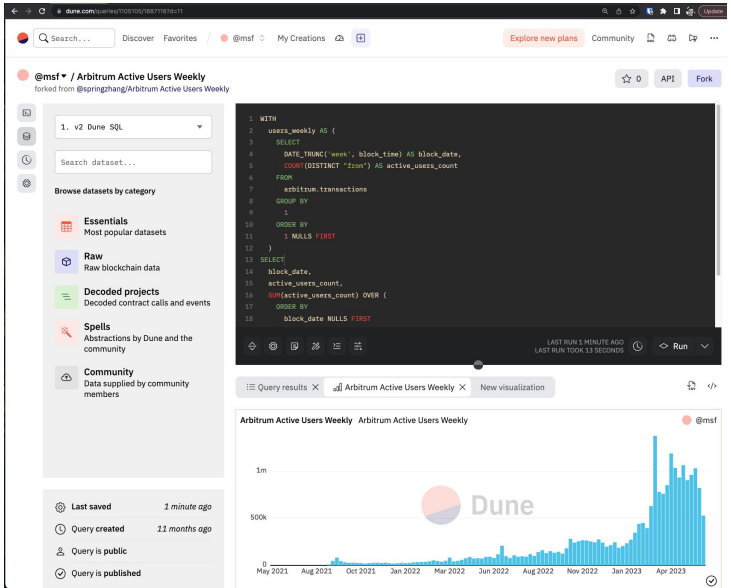
1. Intro to Dune
2. Blockchain Data Challenges
3. Query Experience Challenges
4. The Journey to Trino
5. DuneSQL (extending Trino)
6. Operating DuneSQL
7. Future ahead

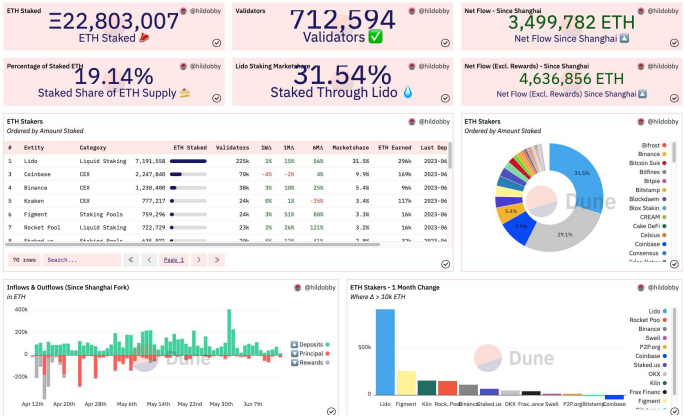
Intro to Dune

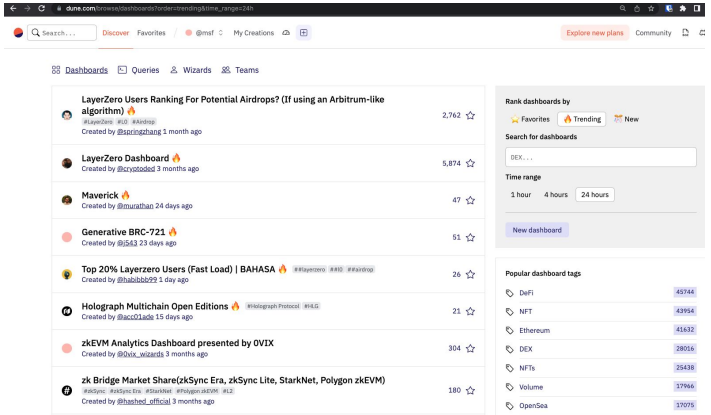
We are on a mission to make crypto data accessible

Dune, a community data platform

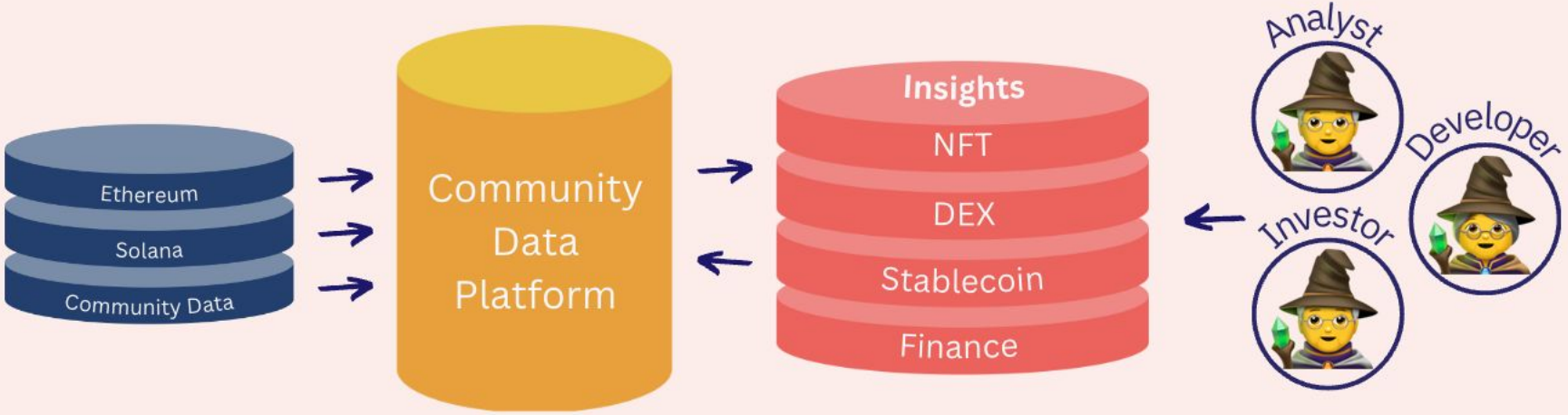
Dune.com is a platform for querying public blockchain data & building beautiful dashboards







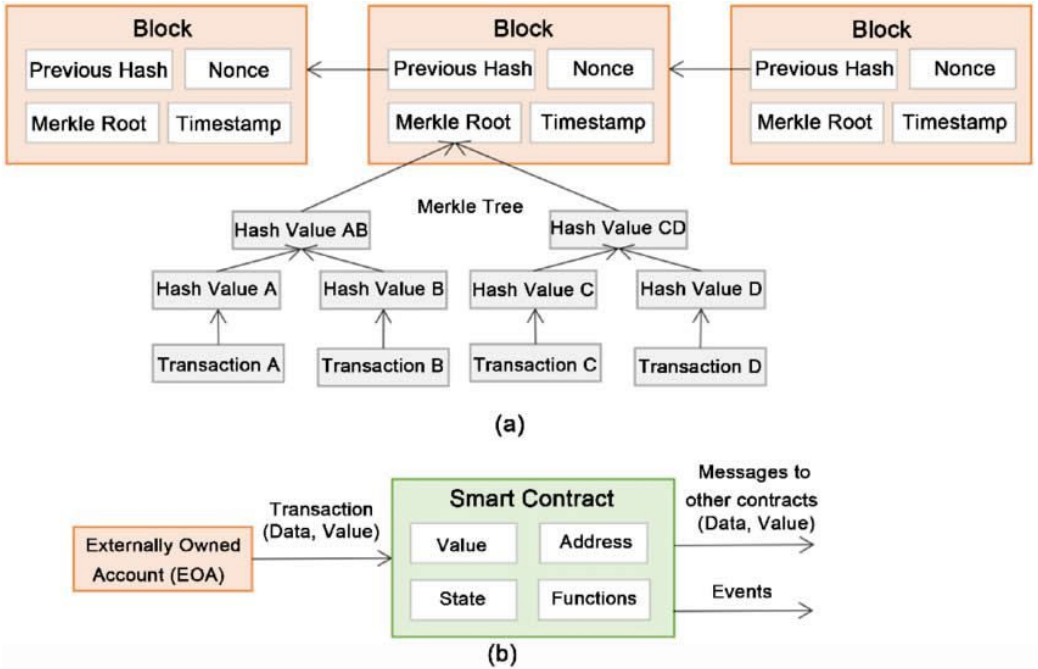
What is a Community Data Platform?

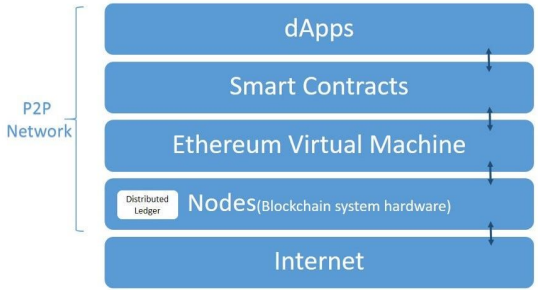


Serverless, open access, community wide collaboration

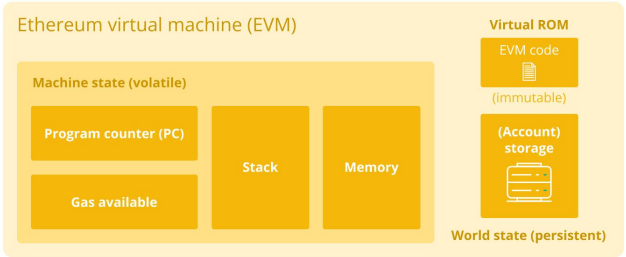
Blockchain Data Challenges

It’s a distributed Virtual Machine





Schematics of an Ethereum Virtual Machine



cointelegraph.com

Blockchain Data Challenges (ingestion)

1. Process and Ingest Raw Data
    - Expose “*raw tables*” transactions, events, logs
2. Deserialize & Decode Function Calls & Arguments
    - Expose *Decoded* smart-contract *tables & views*
    - *Almost 1Million views*
3. Allow community to build abstractions on top
    - Tables & Views
    - Queries & Dashboards

Query Experience Challenges

- 10 000s of queries executed per day
- 10 000s of queries saved & re-used
- Almost a million Tables & Views

- Very heterogeneous SQL queries
- Bimodal query workloads (Interactive vs Batch)
- Many extremely complex queries (>5000 LOC)

The journey to Trino

- PostgreSQL
  - Sharded per Blockchain
  - Vertical scaling. Bottlenecked on storage size and IOPS
- (Data lake) SQL-on-Spark/Databricks
  - Horizontal scaling & unlimited Blockchains
  - Time-to-Market
  - Not interactive enough, Bad 5th, 10th percentile latency
  - Tied to the vendor (cost, bugs, roadmap)
- Other options evaluated:
  - Self Hosted Spark, Presto & Trino
  - Apache Calcite + Trino

DuneSQL

DuneSQL

- A fork of Trino and custom plugins
  - Support for Spark views
  - Improved UX for blockchain data
  - New datatypes: (U)INT256
  - Features to support migrating from Spark
- Control over the *Database*
  - Data layout and data types
  - ETL and schema design
  - Query Experience

Binary data & wallet addresses

- Hex strings (Ethereum, etc.): `0x1234abcdef`
- We store all data as VARBINARY

- Display data as hex strings
- Varbinary literals as hex strings

```
1  select block_time, value, "from"
2  from
3    ethereum.transactions
4  where
5    to = 0xd8dA6BF26964aF9D7eEd9e03E53415D37aA96045
```

Query results

block_time	value	from
2015-10-24 08:49	0	0x1db3439a222c519ab44bb1144fc28167b
2015-09-28 08:24	250000000000000000	0x1db3439a222c519ab44bb1144fc28167b
2015-10-04 03:14	2	0x1db3439a222c519ab44bb1144fc28167b
2015-10-13 06:54	1000000000000000000	0x1db3439a222c519ab44bb1144fc28167b

INT256

- Cryptography (Keccak-256 & others)
- Fixed point arithmetic
- EVM native word size is 256 bits

We support reading and writing (U)INT256 to Delta tables

```
create table
  usdc_monthly_swaps as
select
  date_trunc('month', call_block_time) as month,
  sum(output_amount0) / 1000000 as usd_swapped,
  typeof(sum(output_amount0)) as type
from
  uniswap_v3_ethereum.Pair_call_swap
```

```
select
  date_trunc('month', call_block_time) as month,
  sum(output_amount0) / 1000000 as usd_swapped,
  typeof(sum(output_amount0)) as type
from
  uniswap_v3_ethereum.Pair_call_swap
where
  contract_address = 0x88e6A0c2dDD26FEEb64F039a2c41296FcB3f5640
    -- USDC Uniswap V3 contract
  and call_success = true
group by 1
```

Query results

month	usd_swapped	type
2023-06-01 00:00	-94145547	int256
2023-05-01 00:00	-37478365	int256
2023-04-01 00:00	8943580	int256
2023-03-01 00:00	-3936347	int256
2023-02-01 00:00	-11676900	int256
2023-01-01 00:00	83479107	int256
2022-12-01 00:00	-85132943	int256
2022-11-01 00:00	-106078255	int256
2022-10-01 00:00	134775	int256
2022-09-01 00:00	-17736117	int256

Building for collaboration

- Every* query is a view
- Very popular feature (used in ~30% of new queries)

```
1  with my_address as (select address from labels.ens where name = 'kylling.eth')
2  select * from ethereum.transactions, my_address
3  where to = my_address.address or "from" = my_address.address
```

```
1  select count(*) from query_62591
```

Query results New Query @kylling

_col0
12

Delta Lake on Trino

- The Trino Delta Lake connector is great
- Obstacles when moving from Spark
  - Cannot read Spark views
  - Missing features
    - CREATE OR REPLACE TABLE
    - Custom table properties
    - Generated columns

Delta Lake on Trino

- CREATE OR REPLACE TABLE
- Custom table properties (#17592, #17595)

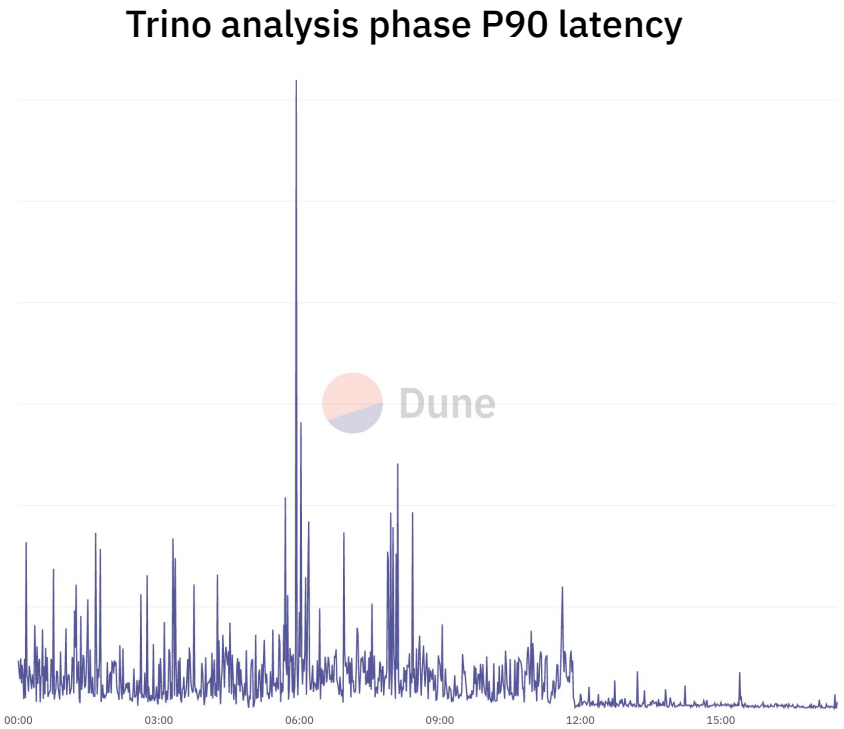
```
CREATE OR REPLACE TABLE nft_trades_volume
WITH (
  extra_properties = MAP(ARRAY['dune.category'], ARRAY['rollup'])
) AS
SELECT project, collection, COUNT(*), SUM(amount_usd) FROM nft.trades;

ALTER TABLE nft_trades_volume
  SET PROPERTIES -- upsert
  extra_properties = MAP(ARRAY['dune.category'], ARRAY['rollup'])
```

Delta Lake on Trino

Delta log limitations

- Cannot have multiple writers from different query engines
- Performance problems for streaming tables with large amounts of metadata (#17408, #17516)

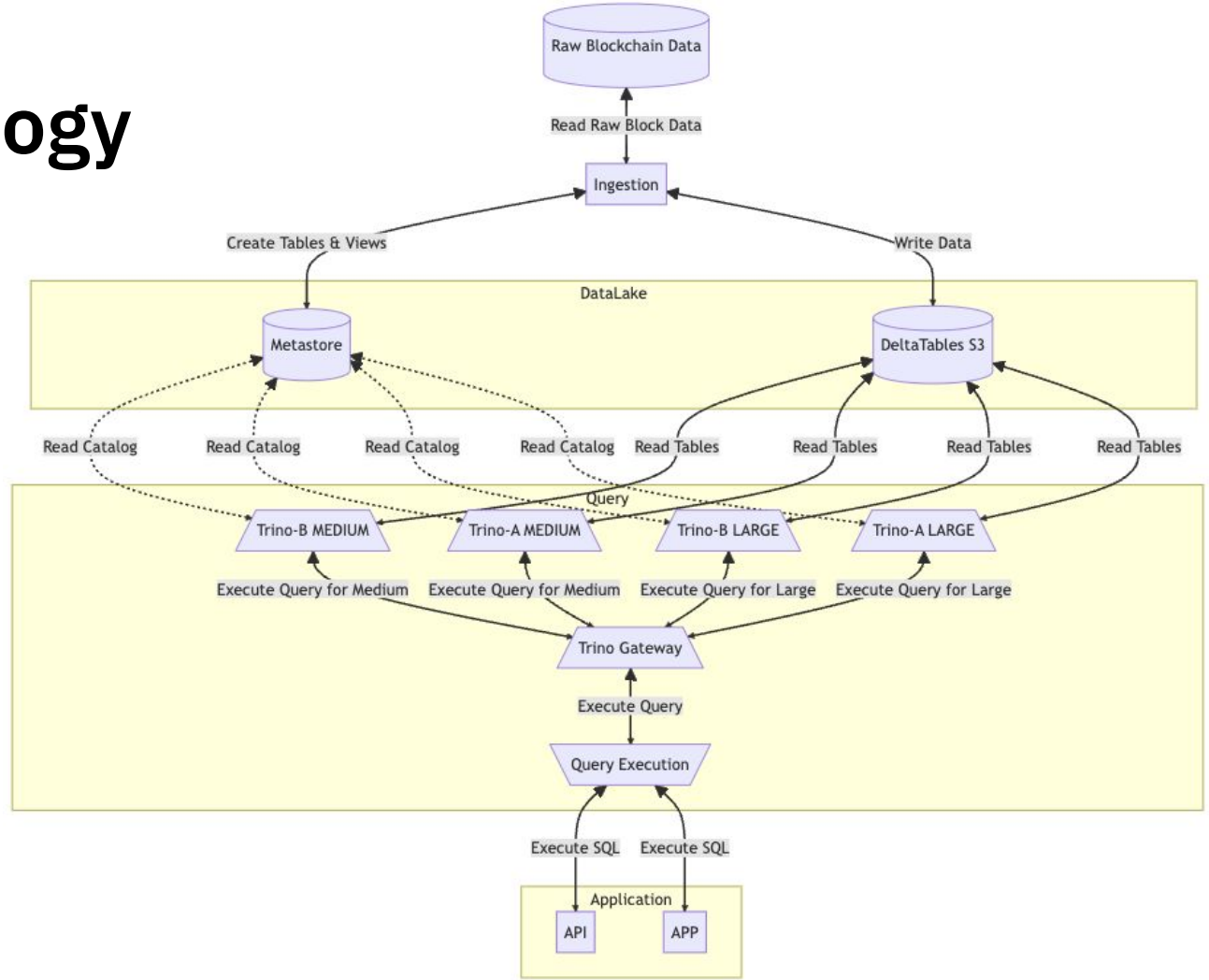


Operating DuneSQL

Operating DuneSQL

- Handle 10 000s of queries per day
  - Different priority classes & performance tiers
  - Track performance & query error rates
- Provide ~*predictable*~ performance per tier
- Capacity planning & Fleet Management
  - >4000 cpus/hour
  - >100B S3 req/month
  - >10 clusters

Topology


Raw Blockchain Data
Read Raw Block Data
Ingestion
Create Tables & Views
Write Data
DataLake
Metastore
DeltaTables S3
Read Catalog
Read Catalog
Read Catalog
Read Catalog
Read Tables
Read Tables
Read Tables
Read Tables
Query
Trino-B MEDIUM
Trino-A MEDIUM
Trino-B LARGE
Trino-A LARGE
Execute Query for Medium
Execute Query for Medium
Execute Query for Large
Execute Query for Large
Trino Gateway
Execute Query
Query Execution
Execute SQL
Execute SQL
Application
API
APP

Scheduling & Load Balancing

- Query Execution Service
  - Queueing , Scheduling & Routing queries
  - Segregation of performance tiers
  - Controlling concurrency & retry logic
  - Performance tracking, Observability
- Trino-Gateway (fork of Lyft’s presto-gateway)
  - Routing Groups & Load Balancing

Clusters and *Clustersets*

- Fixed size clusters for:
  - predictable capacity & performance
  - reduce blast-radius & noisy-neighbor
- Multiple *Cluster-Sets*
- Trino Kubernetes operators to automate operations

Trino k8s operators

- Cluster = k8s deployment
- Load Balancer Integration
- Cluster registration
- Cluster Health checks
- Rolling deployments
- Grouped by profile

```
apiVersion: query.dune.com/v1beta1
kind: TrinoClusterSet
metadata:
  name: community
  namespace: query-engine
spec:
  replicas: 10
  gateway:
    endpoint: http://trino-gateway.query-engine.svc.cl
    routingGroup: community
  gatewayHealthcheckWaitSeconds: 60
  clusterSpec:
    accessControlConfigMapName: access-control-u1mtg
    catalogsSecretName: catalogs-
    configProperties: |
      query.max-memory-per-node=5632000MB
    coordinator:
      instanceType: hpc6a.48xlarge
    extraJvmOpts: |
      -XX:+UnlockDiagnosticVMOptions
    image: 1234.dkr.ecr.us-west-1.amazonaws.com/dune-t
    nodegroup: trino
    serviceAccountName: trino
    spillToDisk:
      enabled: true
      sizeInGb: 50000
      storageClassName: spilltodisk
    workers:
      count: 5
      instanceType: hpc6a.48xlarge
```

Trino k8s operators

- Autoscaling of clusters
- Fleet management
  -

Shout out to our great colleagues:
Belén, Florent and James.

```
apiVersion: query.dune.com/v1beta1
kind: TrinoClusterSet
metadata:
  name: community
  namespace: query-engine
spec:
  replicas: 10
  gateway:
    endpoint: http://trino-gateway.query-engine.svc.cl
    routingGroup: community
  gatewayHealthcheckWaitSeconds: 60
  clusterSpec:
    accessControlConfigMapName: access-control-u1mtg
    catalogsSecretName: catalogs-
    configProperties: |
      query.max-memory-per-node=5632000MB
    coordinator:
      instanceType: hpc6a.48xlarge
    extraJvmOpts: |
      -XX:+UnlockDiagnosticVMOptions
    image: 1234.dkr.ecr.us-west-1.amazonaws.com/dune-t
    nodegroup: trino
    serviceAccountName: trino
    spillToDisk:
      enabled: true
      sizeInGb: 50000
      storageClassName: spilltodisk
    workers:
      count: 5
      instanceType: hpc6a.48xlarge
```

Future plans

- Caching S3 requests
- Materialized views
- Frontend integration
- Improved data layout/Secondary indices
- Sandboxed user defined functions
- Incremental view maintenance
- Trino-DBT and ETL

We’re hiring!

- Caching S3 requests
- Materialized views
- Frontend integration
- Improved data layout/Secondary indices
- Sandboxed user defined functions
- Incremental view maintenance
- Trino-DBT and ETL

Contact details

Jonas
jonas@dune.com

Miguel
miguel@dune.com
https://twitter.com/m3thos

Belén
belen@dune.com

Florent
florent@dune.com

James
james@dune.com

Thank you!