Adapting ClickHouse® to Use Apache Iceberg Storage

Robert Hodges - Altinity CEO

8 July 2025 - Real-time data lakes meetup @ Sentry

ClickHouse is a famous real-time analytic database

Understands SQL

Runs on bare metal to cloud

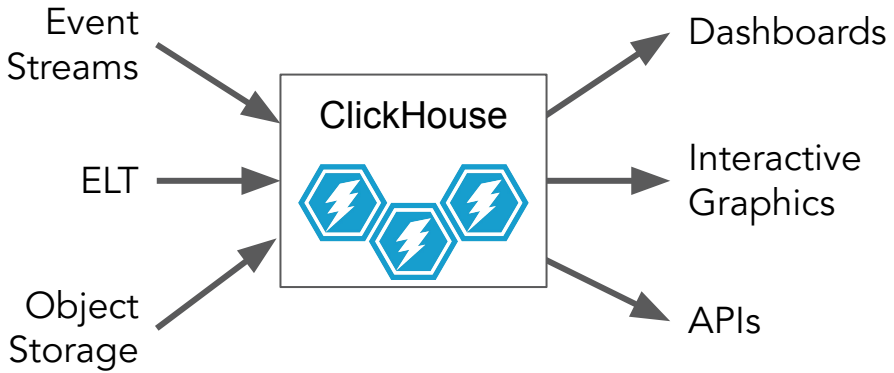
Shared nothing architecture

Stores data in columns

Parallel and vectorized execution

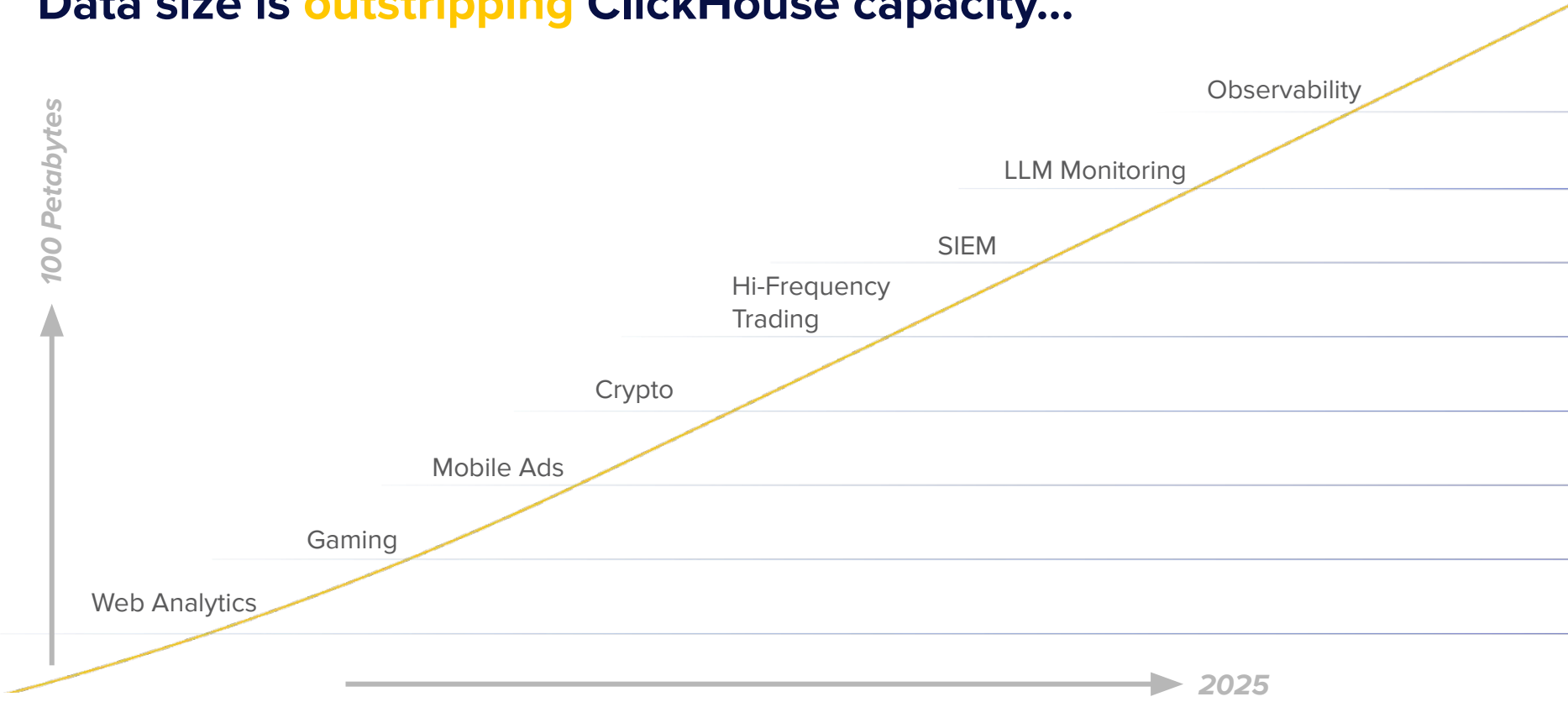
Scales to many petabytes

Is Open source (Apache 2.0)



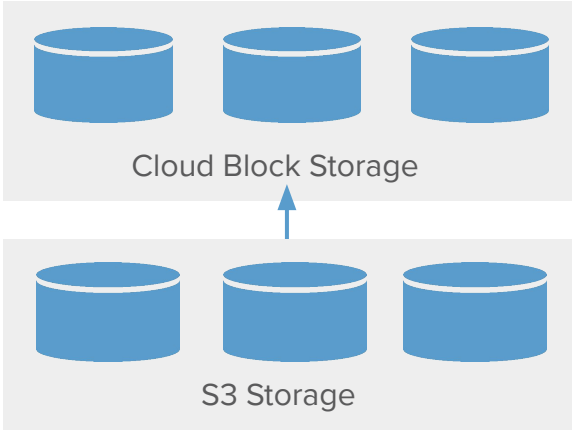
It's a popular engine for real-time analytics

Data size is outstripping ClickHouse capacity…

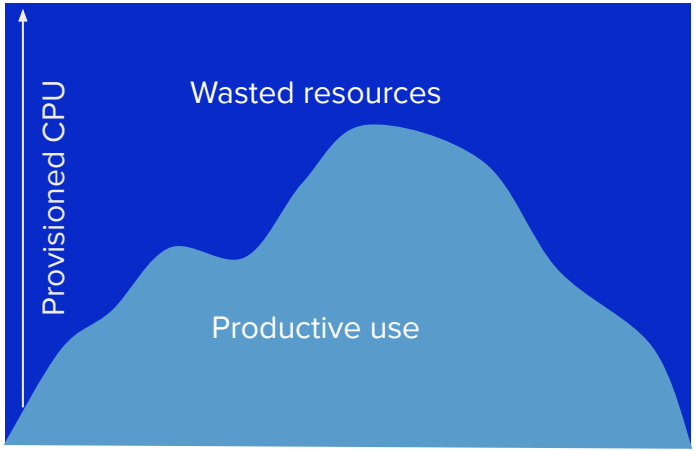


...Leading to pressure on storage and compute cost

Block storage with replication is 10x more expensive



Overprovisioning wastes compute



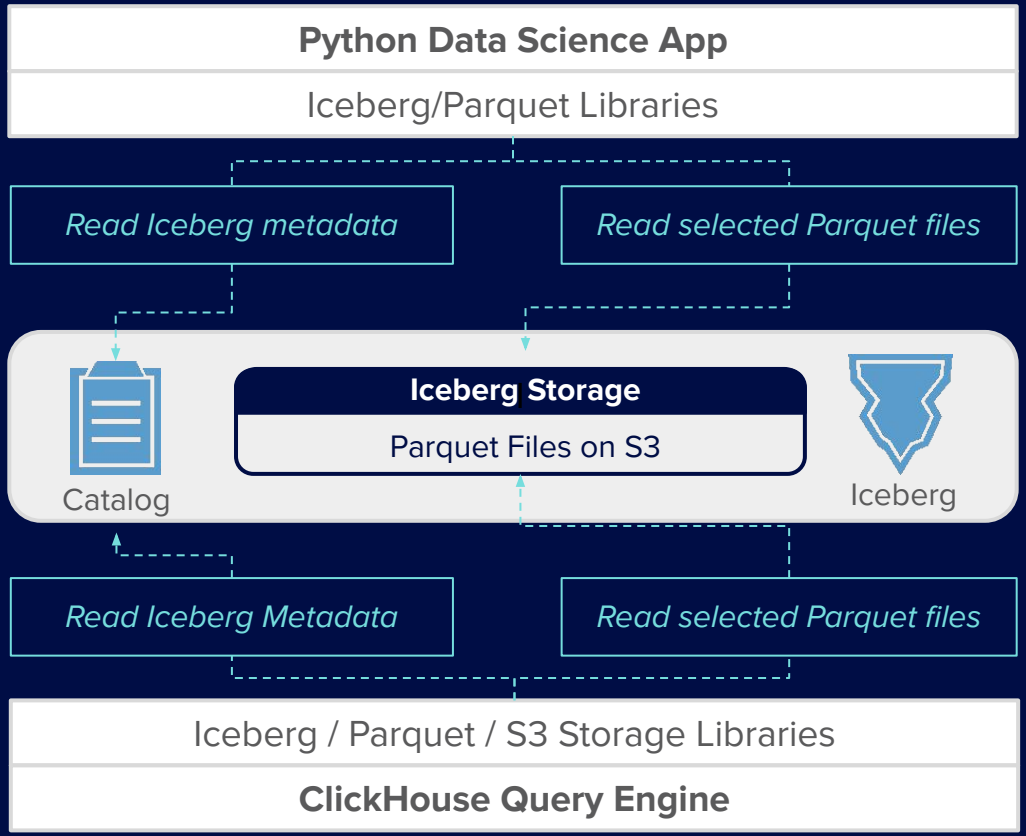
Goals for Project Antalya

Build the next generation architecture for real-time analytics

- Reduce storage costs
- Introduce separation of compute and storage
- Single copy of data
- Preserve ClickHouse compatibility
- Keep system building blocks 100% open source

Our inspiration:
Apache Iceberg

- There's no database!
- Use catalog for table table metadata
- Table files are on S3
- Many apps can read same data
- 100% open source



What we're doing: Workstreams at multiple levels to deliver shared, low-cost data on Iceberg with real-time response

Project Antalya

BUILDS - CONTAINERS - CLOUD NATIVE BLUEPRINTS

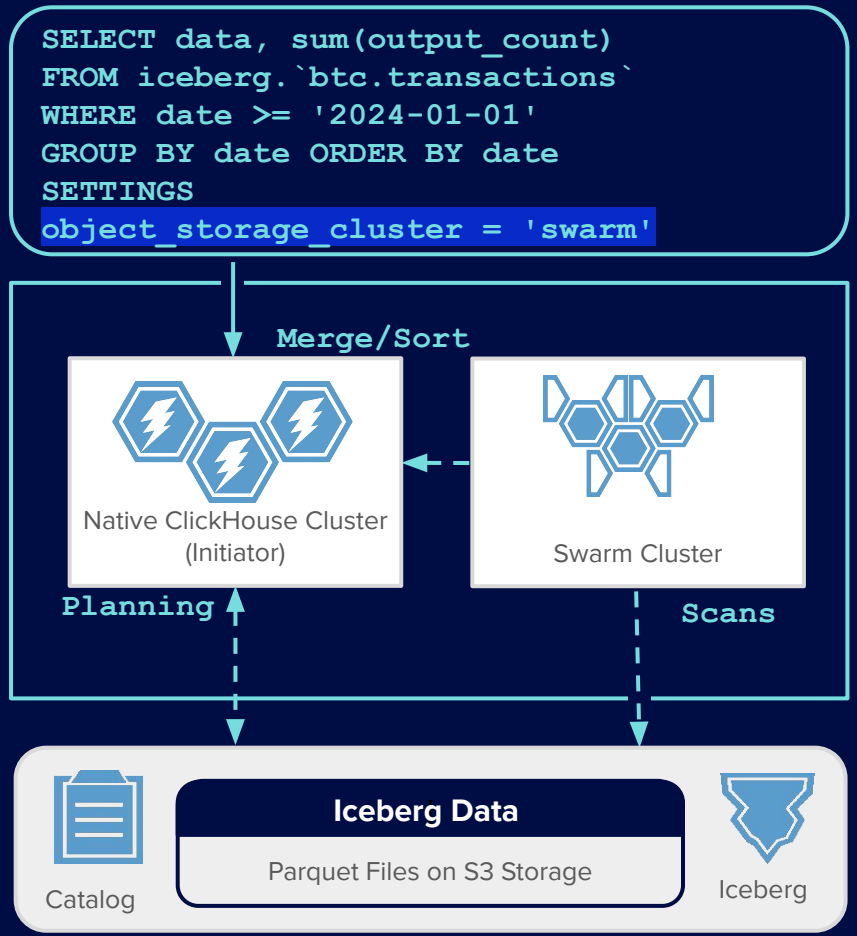
Make I/O on Parquet very fast
Bloom filters, PREWHERE on Parquet columns, metadata caching, S3 file system caching

Shared storage & scalable compute
Stateless compute swarms, caching, Parquet export, tiered storage to Iceberg

Run in any cloud or on-prem
Kubernetes blueprints for auto-scaling, cache deployment, Iceberg catalog

Project Antalya swarm cluster queries

- Initiator plans the query
- Swarm Cluster nodes scan shared files
- Initiator merges and sorts responses

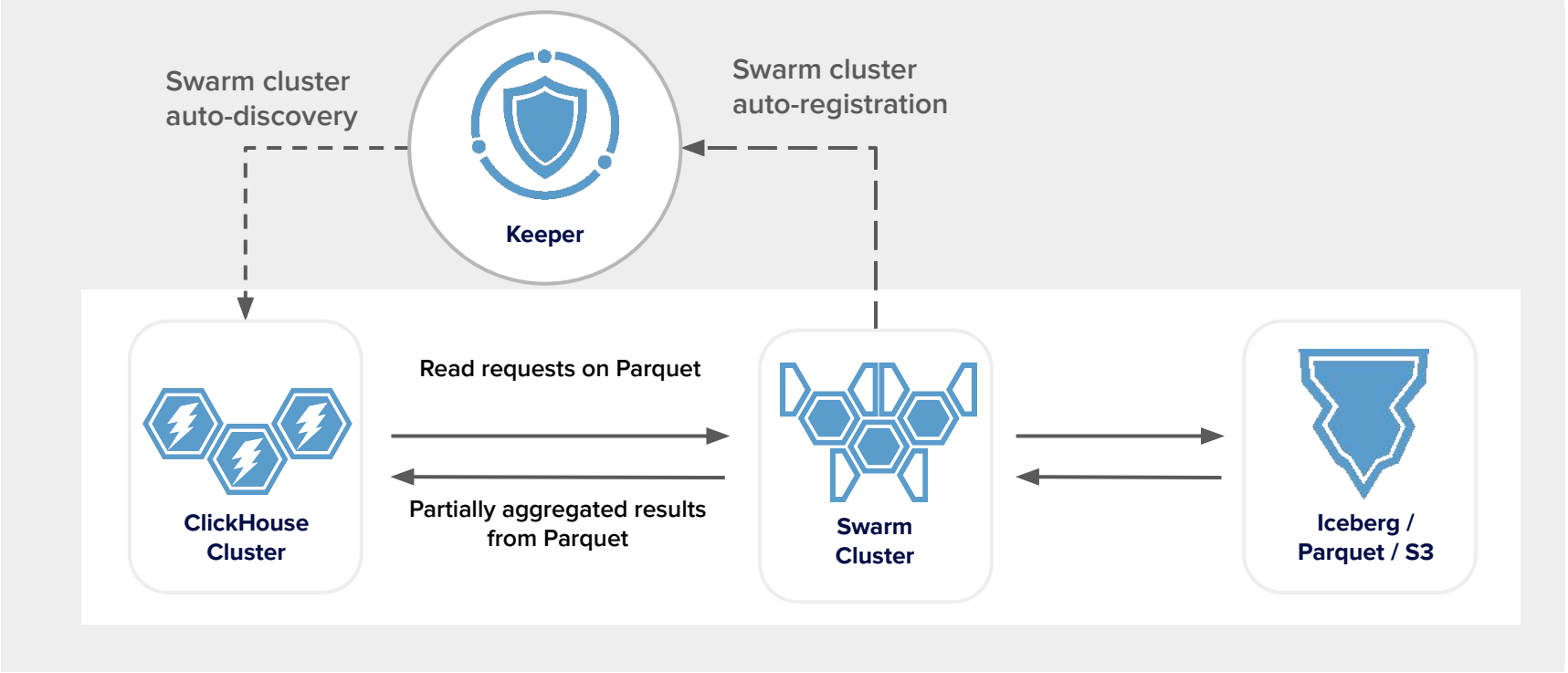


Swarm queries in action

- Swarm SELECT queries of Hive partitioned S3 buckets
- Swarm SELECT queries of Iceberg tables
- Swarm SELECT queries of Glue (needs more testing)

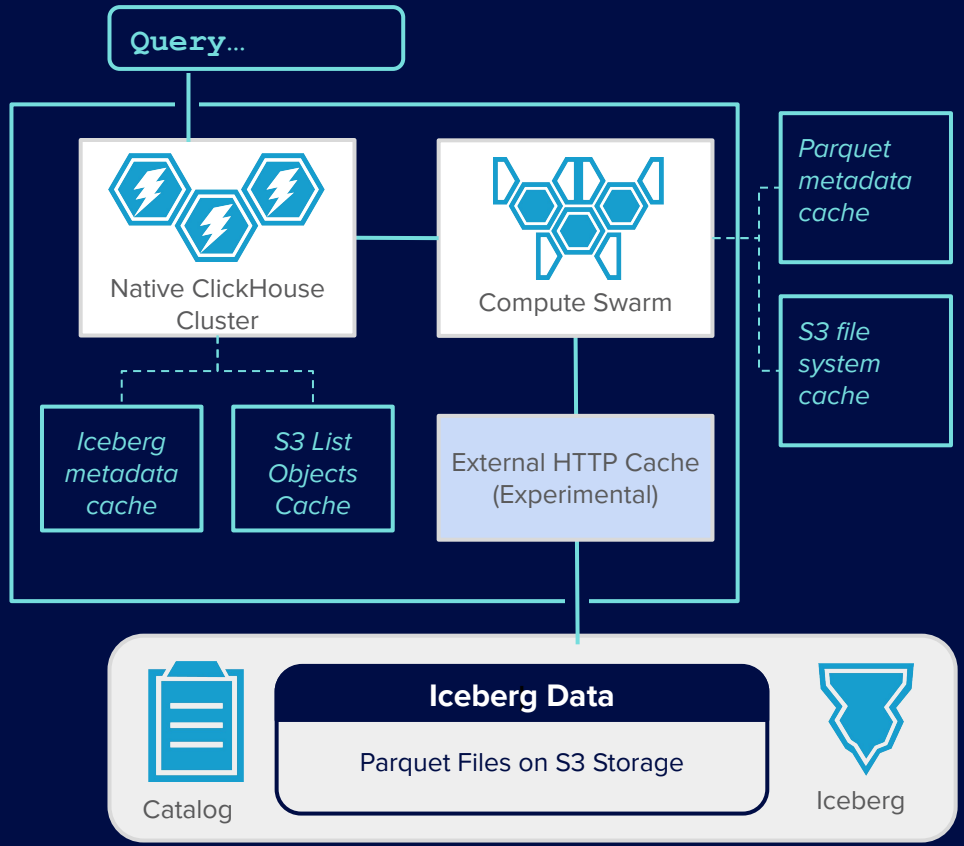
Hive / Plain S3	Iceberg
SELECT date, count() FROM s3('s3://aws-public-blockchain/v1.0/btc/transactions/date=*/*.parquet', NOSIGN) WHERE date between '2025-01-01' and '2025-01-31' GROUP BY date ORDER BY date settings object_storage_cluster='my-swarm', use_hive-partitioning=1	SELECT date, count() FROM ice."aws-public-blockchain.btc" WHERE date between '2025-01-01' and '2025-01-31' GROUP BY date ORDER BY date settings object_storage_cluster='my-swarm'

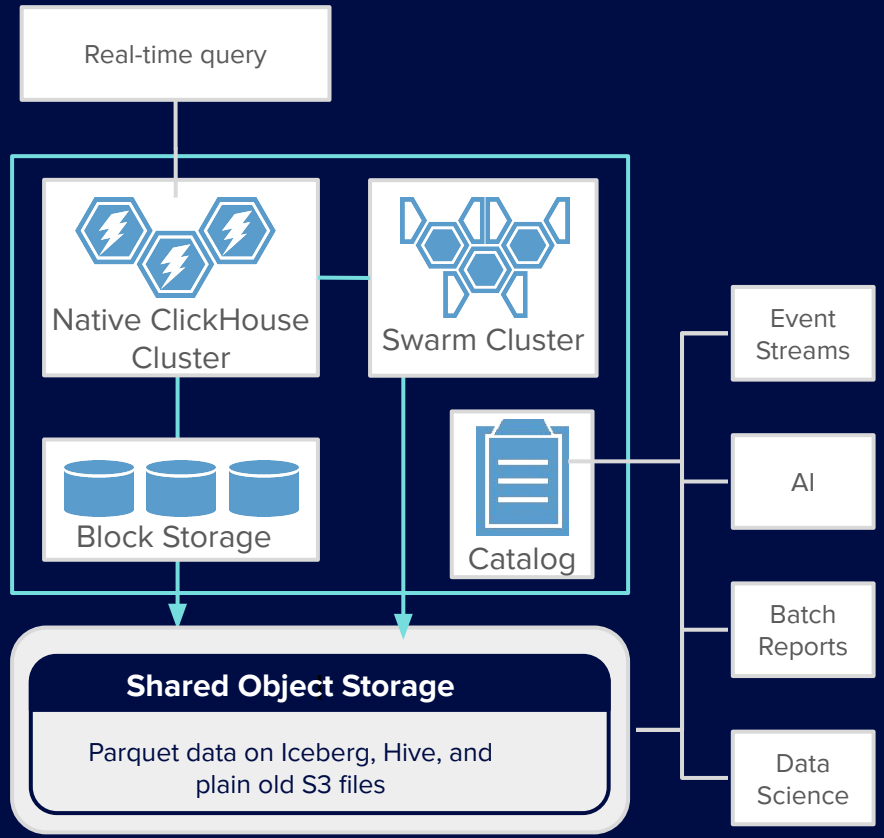
Swarm clusters: handle scans on object storage files



Add Caches to control I/O

- Skip parsing Iceberg metadata
- Skip calls to list files in object storage
- Skip Parquet files entirely
- Skip reading file metadata
- Skip fetching blocks from S3





Current architecture adds lightweight Iceberg catalog with data loading

- Single line command to start catalog (backed by etcd or database)
- Simple commands to load data from other sources
- Authentication using bearer tokens
- Adding automated data loading

Integrated Iceberg Catalog – what it can do

- ice-catalog command starts a catalog that listens on port 5000
- ice command loads Parquet data into Iceberg tables
- ClickHouse can connect as DataLakeCatalog database engine
- Works with ANY ClickHouse 25.4+ or Antalya build

Connecting to catalog from ClickHouse	Loading table data using ice utility
`CREATE DATABASE ice ENGINE = DataLakeCatalog('http://my-catalog:5000' SETTINGS auth_header = '[HIDDEN]', warehouse = 's3://antalya-23x4arm7-iceberg', catalog_type = 'rest'`	`ice insert btc.transactions -p --s3-no-sign-request --s3-region=us-east-2 s3://aws-public-blockchain/v1.0/btc /transactions/date=2025-06-*/*.parquet`

Performance Testing

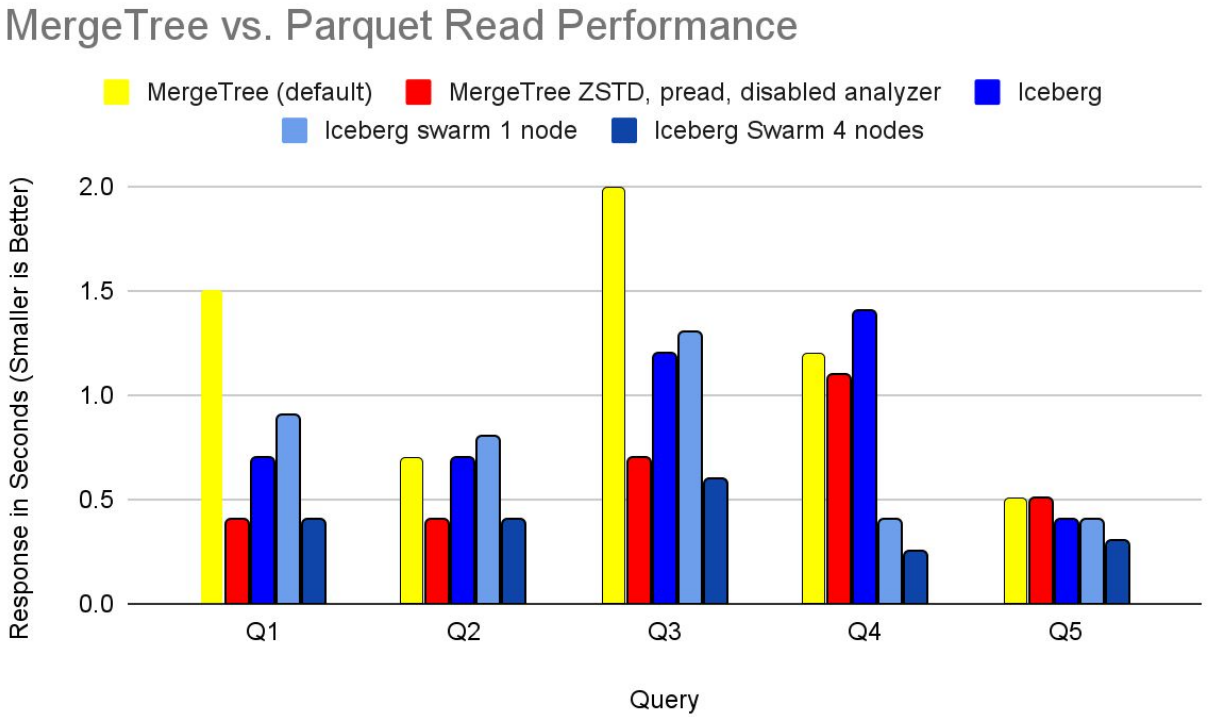
- Use nyc.taxis dataset (1.5B rows)
- MergeTree vs. Parquet
- 5 sample queries
- Caches and partition pruning enabled
- AWS c7g.8xlarge with 32vCPUs
- Run 3-5 times, take lowest response

```
Q1
SELECT
  passenger_count,
  avg(total_amount)
FROM tripdata
GROUP BY
passenger_count
```

```
Q1
SELECT
  passenger_count,
  toYear(pickup_date) AS year,
  count(*)
FROM tripdata
GROUP BY passenger_count, year
```

. . .

Performance results



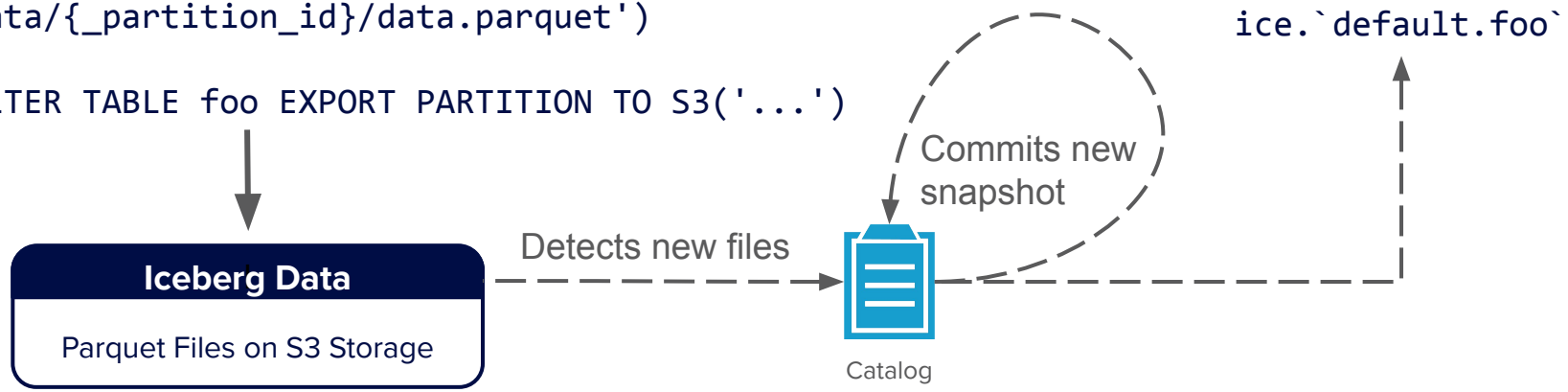
What's coming up in July/August

- Seamless downscaling of swarm while queries are running
- JOIN support within swarm queries
- Automatic snapshots of new Parquet files

```
INSERT INTO
s3('s3://<catalog_warehouse_bucket>/default/foo/
data/{_partition_id}/data.parquet')

ALTER TABLE foo EXPORT PARTITION TO S3('...')
```

```
SELECT * FROM
ice.`default.foo`
```



More things we're working on

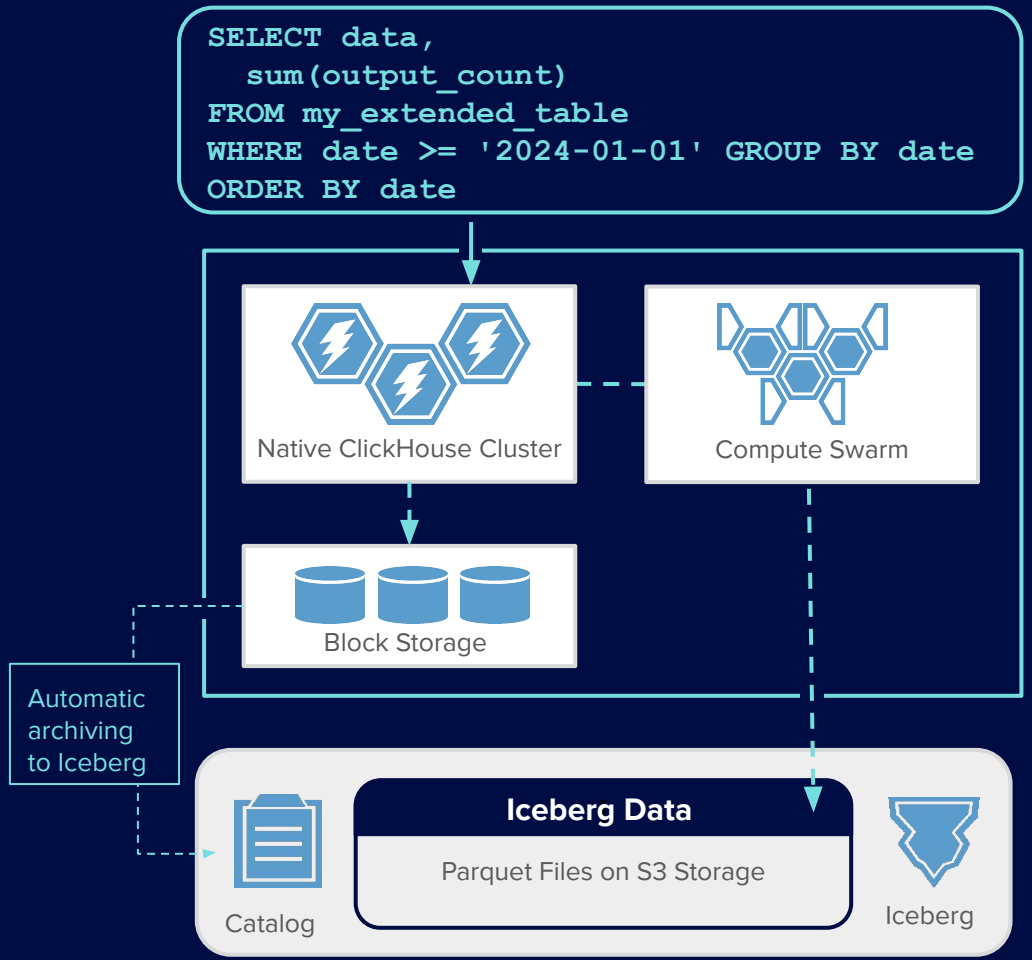
1. Documentation improvements
2. Build refresh based on ClickHouse 25.6
3. Better AWS Glue support
4. Tiered MergeTree+Iceberg tables
5. Compaction
6. Event Stream to Iceberg integration

Plus backports of useful upstream improvements

Roadmap in GitHub

Next step: Tiered storage to Iceberg

- Export from MergeTree to Parquet
- Looks like a single table
- “Watermark” tracks division between layers



How you can help?!!

- Grab the code
- Try it out
- Log issues
- Join our community
- Help us make it better

Antalya Examples Project

Project Antalya resources

- Altinity antalya-examples repo for samples and documentation

  `git@github.com:Altinity/antalya-examples.git`

- Project Antalya code in the Altinity ClickHouse repo (log issues there)

  `git@github.com:Altinity/ClickHouse.git`

- Altinity ice - Tools for Iceberg catalogs

  `https://https://github.com/Altinity/ice`

- Join the Altinity Public Slack to find out more: `https://altinity.com/slack`

Summary

- Project Antalya is extending ClickHouse to use Iceberg as table storage
- Swarm clusters, caches, Iceberg catalog work now
- Parquet reads are reaching parity with MergeTree
- Upcoming attractions:
  - Support for S3 table buckets
  - Export from MergeTree to Iceberg
  - Tiered storage
- The end result: **real-time data lakes that run anywhere**

Getting Started

Thank you!
Questions?

Contact us to learn more and join our community:

https://altinity.com

https://altinity.com/slack

https://github.com/Altinity/antalya-examples

ALTINITY

ClickHouse is a famous real-time analytic database

Understands SQL

Runs on bare metal to cloud

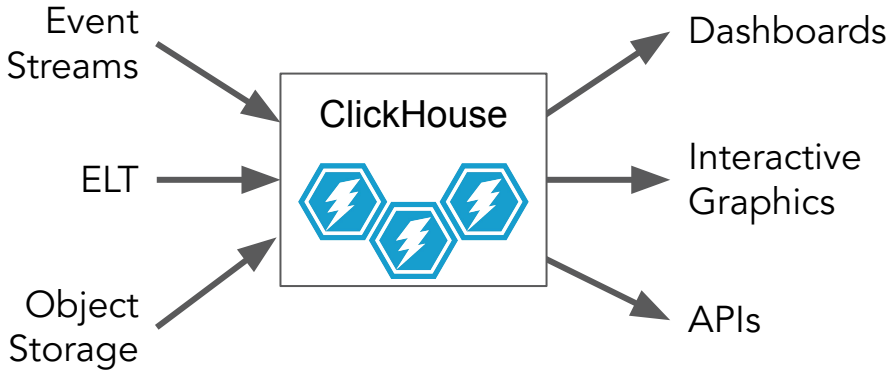
Shared nothing architecture

Stores data in columns

Parallel and vectorized execution

Scales to many petabytes

Is Open source (Apache 2.0)



It's a popular engine for real-time analytics

Icons- Transparent

Clickhouse (Native) Cluster

Director Cluster

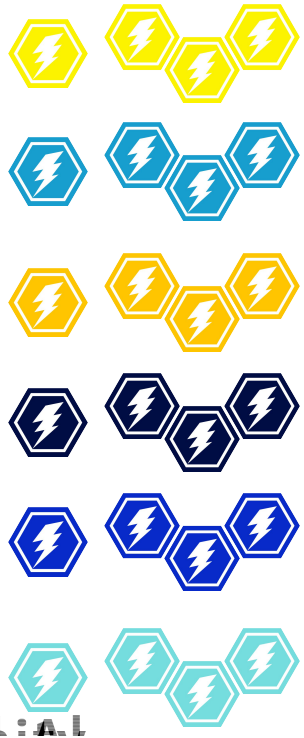
Swarm Cluster

Keeper

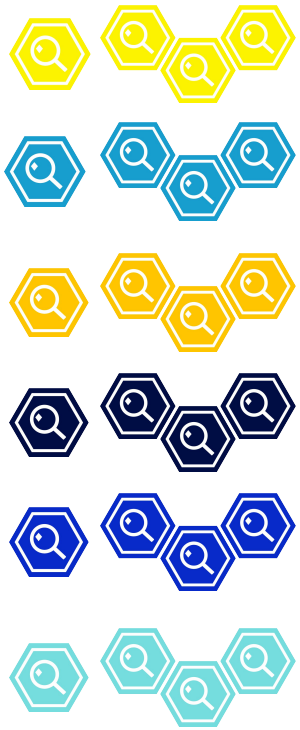
Other

Icons-Stackable on White Backgrounds

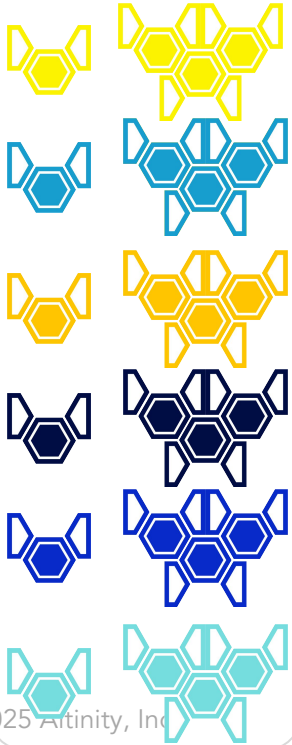
Clickhouse (Native) Cluster



Director Cluster



Swarm Cluster



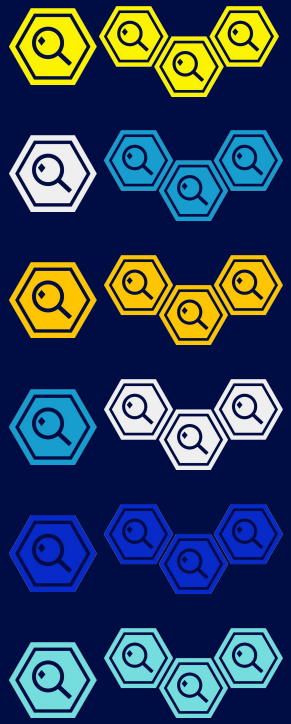
Keeper

Other

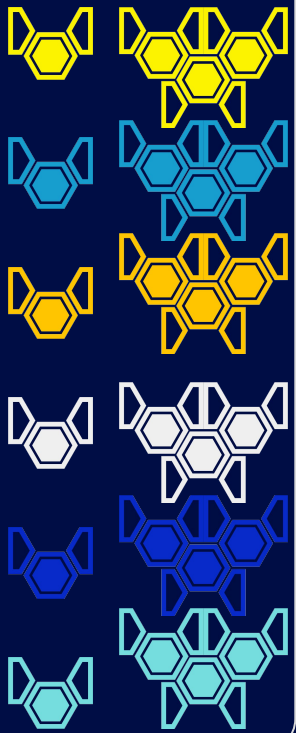
Icons-Stackable on Dark Backgrounds

Clickhouse (Native) Cluster

Director Cluster



Swarm Cluster



Keeper

Other