ClickHouse Meetup: Shanghai

Melvyn Peignon

Principal Product Manager for ClickHouse Core

E. melvyn@clickhouse.com

March 2025

ClickHouse

Introduction

ClickHouse

ClickHouse

2009	2012	2016	2021	2022
Prototype	*Production*	*Open Source*	*ClickHouse Inc.*	*ClickHouse Cloud*

The Most Popular Analytics Database on the Planet

#1
Analytics DB on DB-Engines

Over
39,000
GitHub Stars

Over
200,000
Community Members

ClickHouse

ClickHouse, Inc.

Founded

2021

by leaders from Elastic, GCP, Netflix and Salesforce

Raised

$300M+

from Tier 1 venture capital firms: Benchmark, Index, Lightspeed

Onboarded

1000+

customers since launching ClickHouse Cloud in 2022

230+

employees

Mountain View, CA
Amsterdam, NL

AWS GCP Azure
Alicloud

Strong partnership with all major cloud providers

ClickHouse

Use Cases

Real-Time Analytics

- Applications, Dashboards, APIs
- Customer facing
- Interactive
- Querying ClickHouse directly

Data Lake & Warehouse

- Business data for internal use
- Use any BI software on top (Tableau, Power BI, Superset, Metabase)
- Move between ClickHouse and Iceberg, Parquet, Delta Lake, etc.

Observability

- Logs, metrics, traces
- Built-in integration with OpenTelemetry
- Grafana as a UI
- Unify on SQL as the query language

Machine Learning & GenAI

- Feature Store
- Vector Search
- LLM Observability
- MCP Server

Nansen: Real-time analytics on the blockchain

Nansen

Customer Story

"

Nansen, a leading blockchain analytics platform, transitioned from BigQuery to ClickHouse Cloud to address performance issues and rising costs. This move significantly enhanced their data processing speed and efficiency, enabling real-time insights for their users.

"

Goldsky: Blockchain at scale in real-time

GOLDSKY

Customer Story

"

Goldsky uses ClickHouse, Redpanda, and Apache Flink to deliver real-time blockchain analytics. The architecture enables efficient, multi-tenant processing and analytics on blockchain data at scale.

"

Demo

ClickHouse

Opensee: Analytics for institutional banking

opensee

Customer Story

"

OpenSee uses ClickHouse to analyze terabytes of financial data daily, enabling real-time insights into global markets. ClickHouse's performance and scalability allow OpenSee to efficiently process large datasets, delivering valuable analytics for financial decision-making.

"

Coinhall: Powering Blockchain Data platform

Coinhall

Customer Story

"

CoinHall uses ClickHouse to power its blockchain data platform, enabling fast, scalable analytics for DeFi applications. With ClickHouse, CoinHall delivers real-time insights into blockchain activity, helping users track key metrics like transaction volumes and token movements.

"

Cognitiv: ClickHouse as a feature store

COGNITIV™

The brain behind the brands

Customer Story

"

Cognitiv uses ClickHouse to process large datasets, enhancing its machine learning models for digital advertising. By harnessing ClickHouse's speed and scalability, Cognitiv improves targeting, optimization, and decision-making for ad campaigns.

"

W&B: Scaling AI development using ClickHouse

Customer Story

"

Weights & Biases uses ClickHouse to scale its AI development, allowing for real-time analytics and improved tracking of machine learning workflows. ClickHouse helps optimize model performance and decision-making by efficiently managing large datasets.

"

LangChain: Observability for LMM

LangChain

Customer Story

"

LangChain uses ClickHouse to manage large-scale data from language model interactions, leveraging its speed and scalability to optimize data processing and querying for efficient application performance.

"

Why ClickHouse?

Benefits our customers report

1 **Performance**

2 **Cost**

3 **Productivity**

ClickHouse Cloud

Managed | Hybrid | Self-managed

ClickHouse

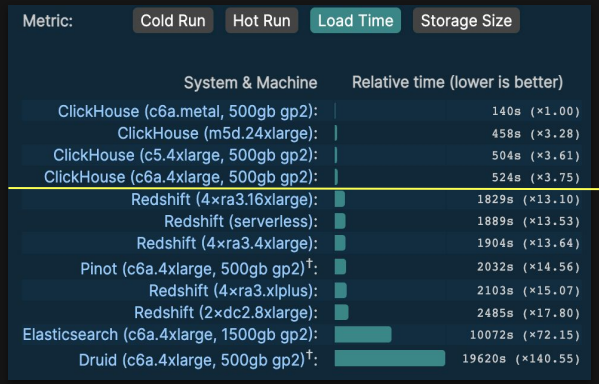
Performance

The fastest analytical database on the planet

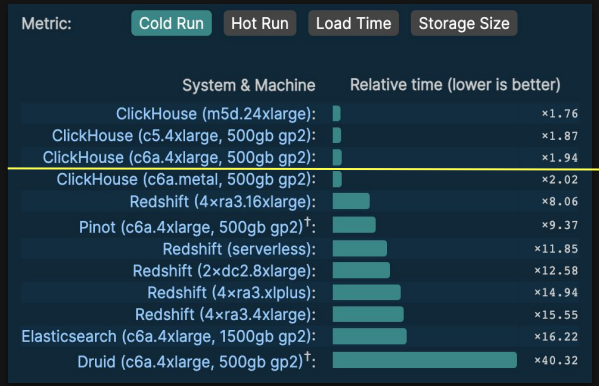
Real-time performance
Build interactive use cases for internal and external use

Concurrent queries
Very high query concurrency (1000 per node by default)

High data volumes
Read and write billions of rows - at the same time



37x faster loading data



20x faster querying data

benchmark.clickhouse.com

ClickHouse

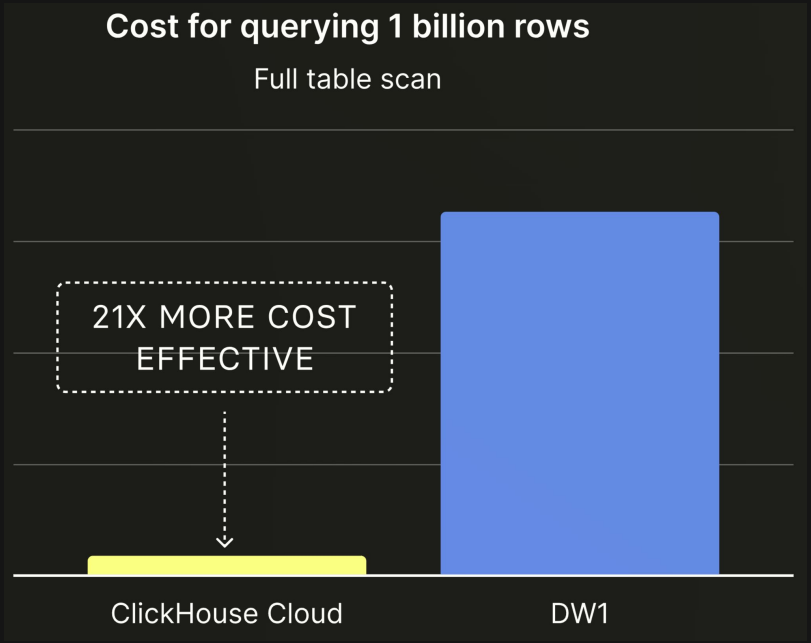
Cost

Faster queries = Lower cost

Efficiency
The faster queries finish, the less resources are needed

Serverless compute
Only the resources you need when you need them

Predictability
Pay for CPU and Memory, not per query



ClickHouse

Developer Productivity

Beyond just SQL

Open Source
Run locally and as part of CI/CD. No vendor lock-in.

Large community
Many resources available for free

Over 1500 built-in functions
Arrays, JSON, Geo, Statistics, User-defined, etc.

```
WITH
  Combinations AS (
    SELECT
      ['a', 'b'] AS letters,
      [1, 2, 3] AS numbers
  )
SELECT
  ARRAY(
    SELECT AS STRUCT
      letters[SAFE_OFFSET(index)] AS letter,
      numbers[SAFE_OFFSET(index)] AS number
    FROM Combinations
    CROSS JOIN
      UNNEST(
        GENERATE_ARRAY(
          0,
          LEAST(ARRAY_LENGTH(letters), ARRAY_LENGTH(numbers)) - 1)) AS index
    ORDER BY index
  );

/*-----------------------------*
 | pairs                       |
 +-----------------------------+
 | [{ letter: "a", number: 1 }, |
 |  { letter: "b", number: 2 }] |
 *-----------------------------*/
```

Cloud Data Warehouse

```
WITH Combinations AS
    (
        SELECT
            ['a', 'b'] AS letters,
            [1, 2, 3] AS numbers
    )
SELECT arrayZip(letters, arrayResize(numbers, length(letters))) AS pairs
FROM Combinations;
   ┌─pairs─────────────┐
1. │ [('a',1),('b',2)] │
   └───────────────────┘
```

ClickHouse

ClickHouse

ClickHouse Cloud and Alibaba Cloud Enterprise Edition

The easiest way to use ClickHouse

Deployment options
Fully managed, hybrid and self-managed

Choose your cloud provider and region
Many regions available in AWS, GCP, Azure (not available in China Mainland) and Alicloud

Highly Available
Three availability zones by default

Configurable backups
Choose the frequency and retention period

Elastic Deployments
Scale up and down, Scale out and in without friction

Flexible billing
PAYG or committed spend.

** Access all documents at trust.clickhouse.com*

Roadmap for 2025

ClickHouse

Focus Area for 2025

JOINs

Expand supports for JOINs.

- Focus on Performance
- Better default experience
- Better resource usage

Distributed Query

Distribution of queries designed for cloud workload

- Leverage replicas
- Expanded support for parallel replica
- Better default experience

Data Lakes

Focus on Iceberg and Delta Lake support

- Catalog integrations
- Best-in class support for Iceberg and Delta
- Expand Data Lake operational workload

Mutable Data

Improved experience with mutable data

- Lightweight operations
- Improved mutations internals
- RMT Improvements

Inverted Index

Better experience for observability workload

- Additional index type
- Index aimed at improving search experience for logging workload

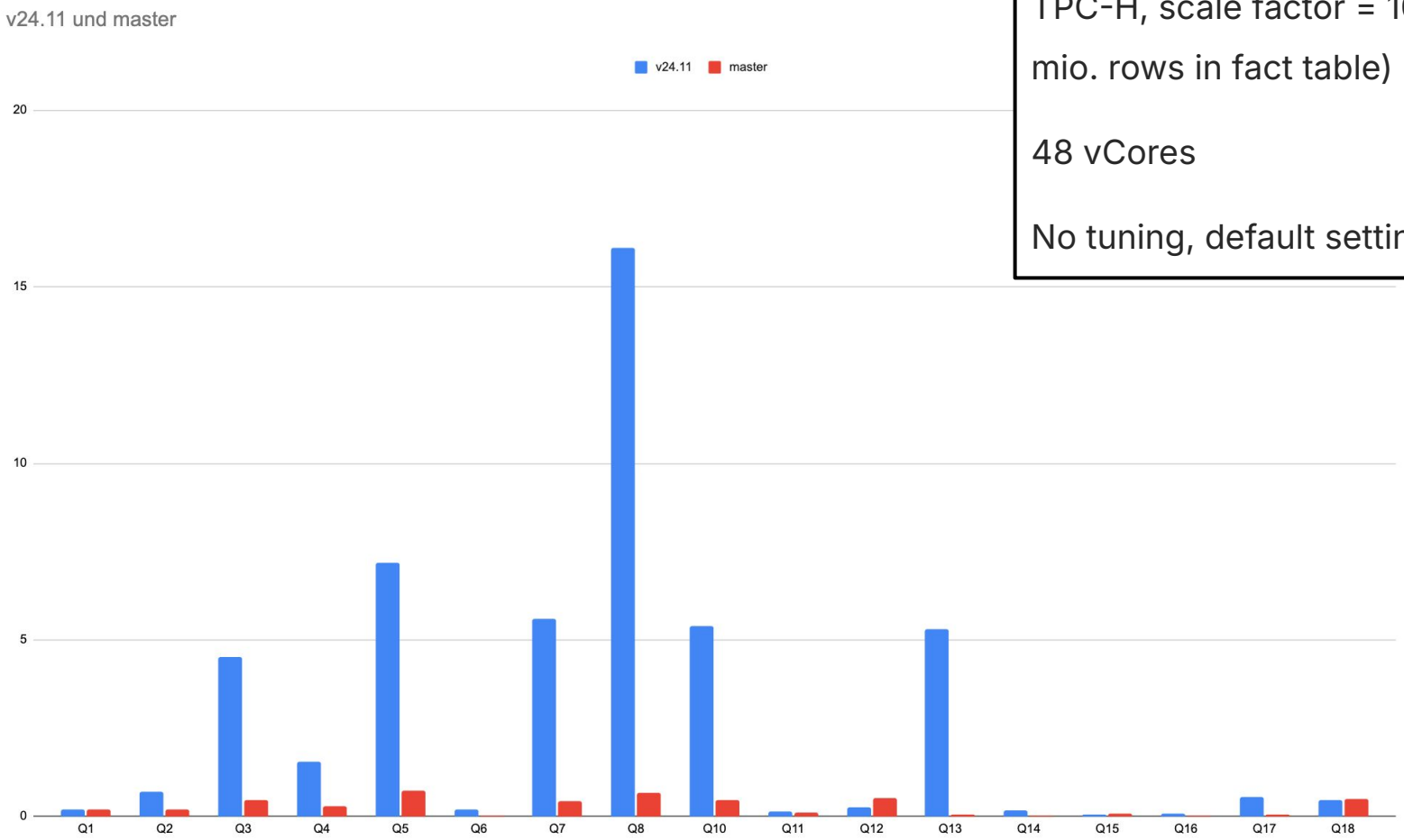
Better JOIN support

JOINs

Expand supports for JOINs.

- Focus on Performance
- Better default experience
- Better resource usage

- Hash JOIN by default
- Improved Push-down
- Statistics based JOIN reordering
- Correlated subquery
- Parallelize sort merge join
- Min-max index by default
- Partition Key support for JOIN



TPC-H, scale factor = 10 (60 mio. rows in fact table)

48 vCores

No tuning, default settings

Distributed Query Execution

Distributed Query Execution

Distribution of queries designed for cloud workload

- Leverage replicas
- Expanded support for parallel replicas
- Better default experience

- Parallel replicas GA
- Distributed Query Planning
- Auto PR:
  - Disable and enable PR based on meta information
- Distributed INSERT … SELECT

Data Lakes

Data Lakes

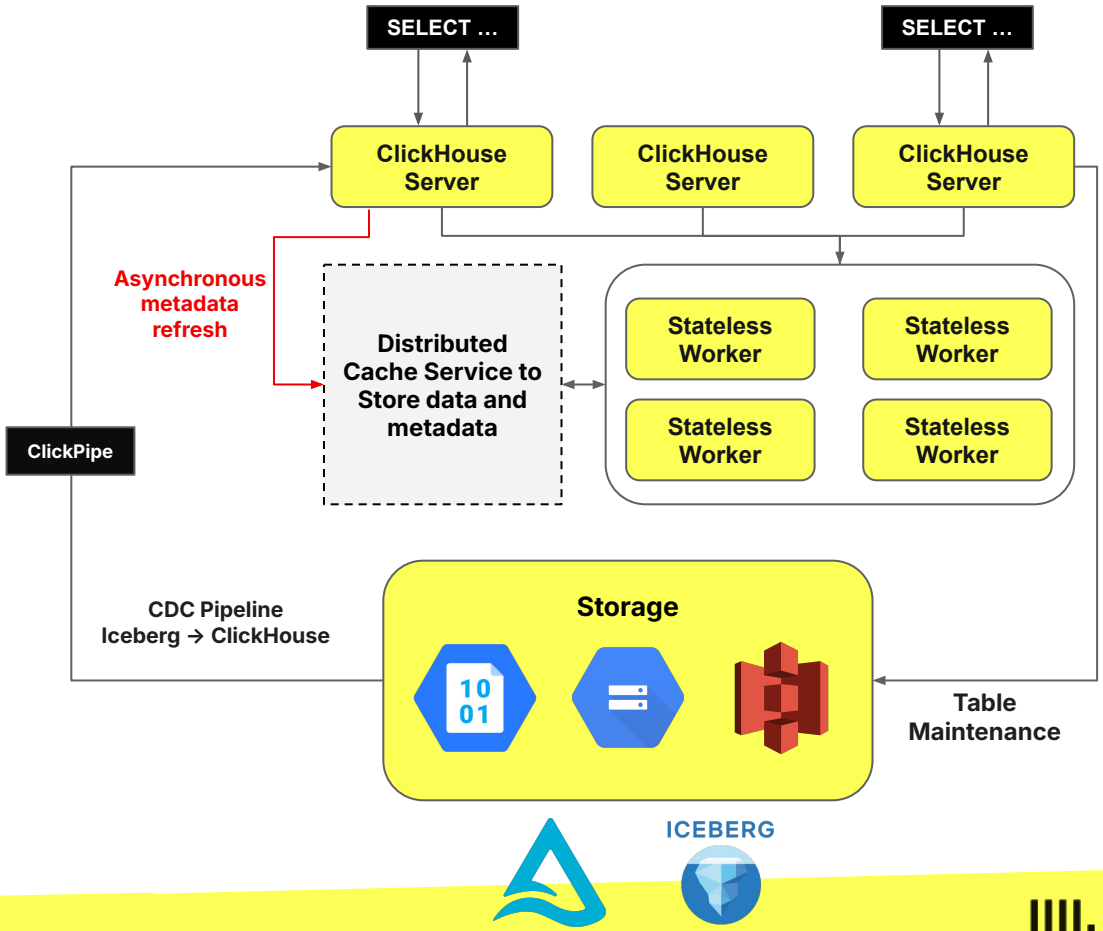
Focus on Iceberg and Delta Lake

- Catalog integrations
- Best-in class support for Iceberg and Delta
- Expand data lake operational workload

- Integration with popular catalog:
  - Glue, Polaris, Unity, …
- Compatibility with open-table format:
  - Iceberg v2/v3
  - Delta Kernel
- Additional support for operational workload:
  - Write support
  - Compaction
- Performance:
  - Caching
  - Better statistics utilization

Emphasis on the Infrastructure

- Fully Serverless:
  - Per query usage
  - Distributed cache
  - “Head nodes” for native ClickHouse materialization support
- Iceberg/Delta Lake metadata management
- Parquet maintenance (compactions)
- Metadata cached in distributed cache
- CDC pipeline Iceberg -> ClickHouse



Mutable Data

Mutable Data

Improved experience with mutable data

- Lightweight operation
- Improved mutations internals
- RMT improvements

- Add support for lightweight operations using “Patch part”:
  - Updates
  - Deletes
- Better guardrails for mutations
- Improved performance on RMT:
  - Support for additional indices
  - Prewhere with Final

Inverted Index

Inverted Index

Better experience for observability workload

- Additional Index type
- Index aimed at improving search experience for logging workload

- Removal of the existing inverted index
  - Performance improvement are non-existent
  - Doesn’t work on top of pack parts
- Target workload is for logs:
  - Simple analyzer and tokenization rule

Outside of the Top 5

Closing the Gap between Cloud and OSS:

- **Redis Table Engine**
- **Support for Protobuf and CapNproto**

Improving Cloud Experience with Core Features:

- **SMT Merge Optimizations**
- **Snapshots**
- **Dynamic Sharding**
- **Remote Database engine**

Analyzer:

- **Fixing remaining bug and improvements**
- **Migration to the new analyzer**
- **Cleanup Analyzer**

Serverless Vision:

- **Migration Shared Catalog**
- **Remove Metadata for System Database**
- **Stateless worker**
- **Distributed Cache**

R&D:

- **Support for PromQL**
- **Streaming Queries**
- **Transactions (SMT + RMT)**
- **Unique Key Constraints**

Road to GA:

- **JSON object**
- **Vector Search**

Performance Optimizations:

- **Block level hints**
- **Query cache for partial results**
- **Lazy columns**
- **Materialized CTE**
- **Automatic Low Cardinality**
- **Statistics**

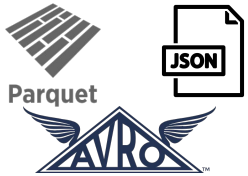
Security:

- **CMEK for Azure**
- **Secure Named Collection**
- **JWT**

Data Lake Deep Dive

ClickHouse

Extensive Support for Data Lake



And many more...

[s3 table function](#)
[azure table function](#)

[Iceberg Table Function](#)
[Iceberg Table Engine](#)

[Delta Table Function](#)
[Delta Table Engine](#)

[Hudi Table Function](#)
[Hudi Table Engine](#)

But Existing Limitations for Iceberg

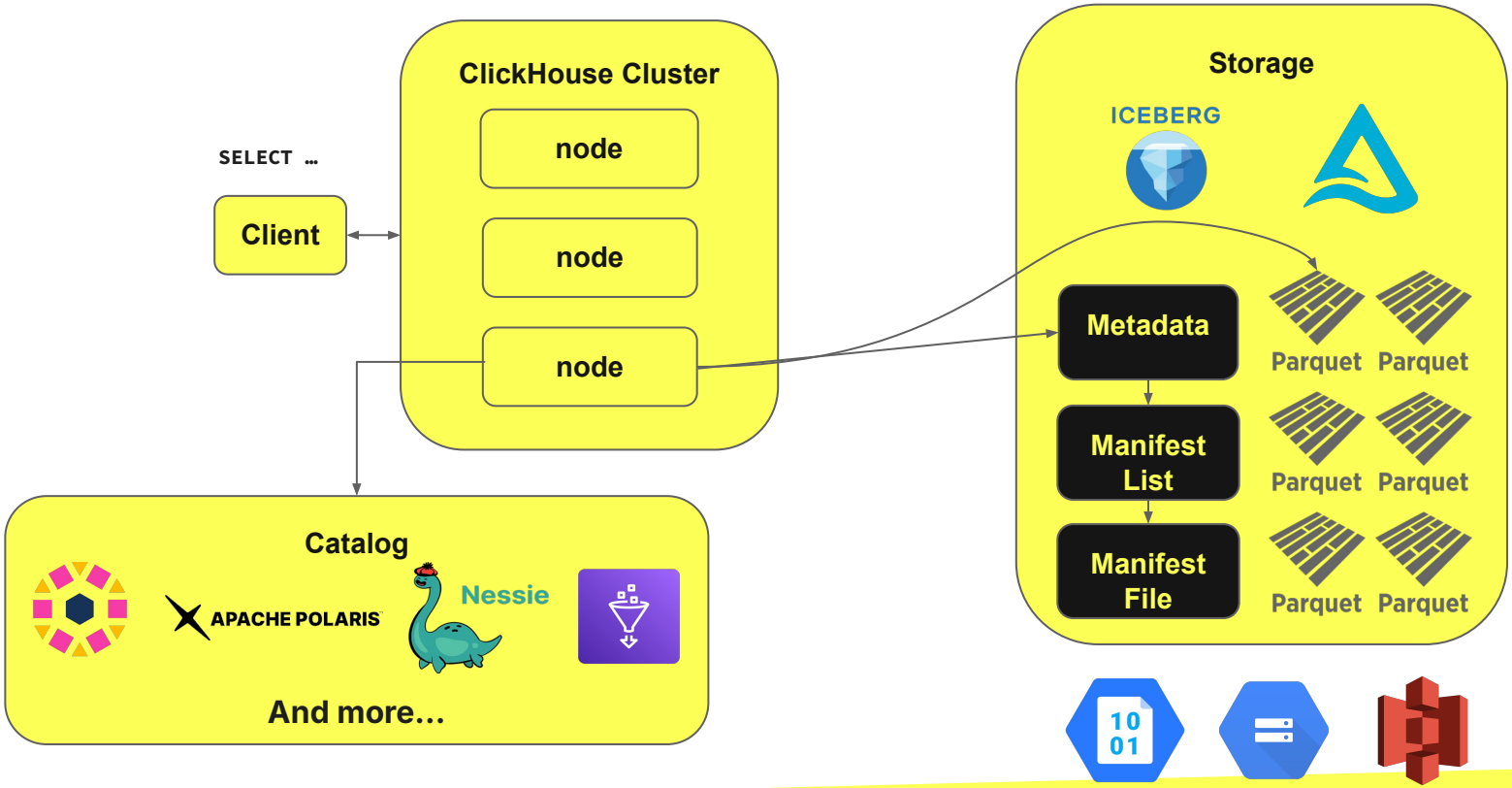
```
SELECT * FROM iceberg('iceberg_table_path')
```

Executed on a single node

Support iceberg v2 but not all of the features of Iceberg

Require path to the storage of the metadata

Adding support for Data Catalog



Demo

ClickHouse

Increasing Open Table Format Support

- Support for partitioning ✅
- Support for deleted rows ⚙️
- Support for schema evolution ✅
- Time Travel ⚙️
- Support for write

- More Native integration with Delta Lake by adding support for Delta Rust Kernel: https://docs.delta.io/latest/delta-kernel.html ✅
- Support for write

Questions?

ClickHouse