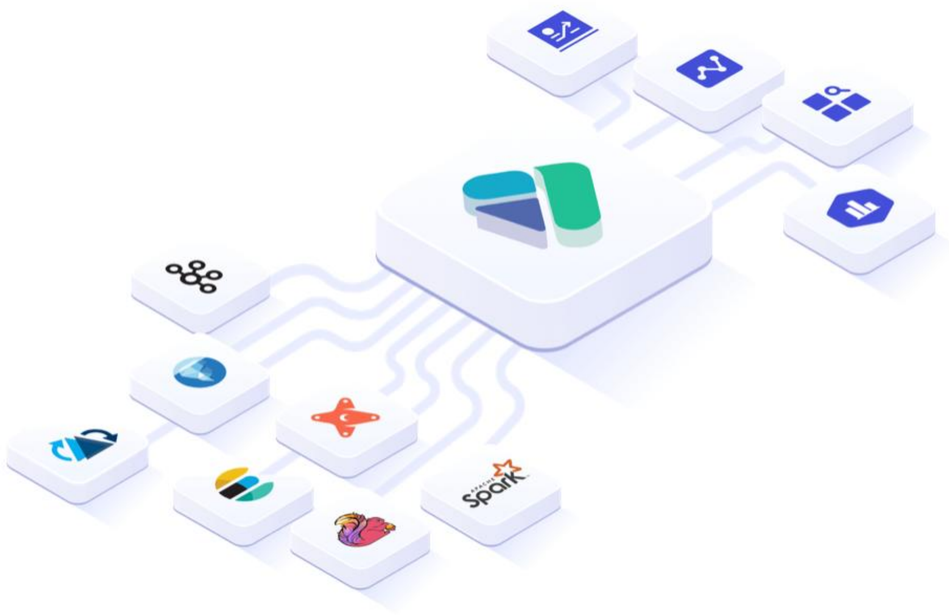
Apache Doris

A Next-Generation Real-Time Data Warehouse

- What can it do?
- Community
- Performance
- Architecture & Features



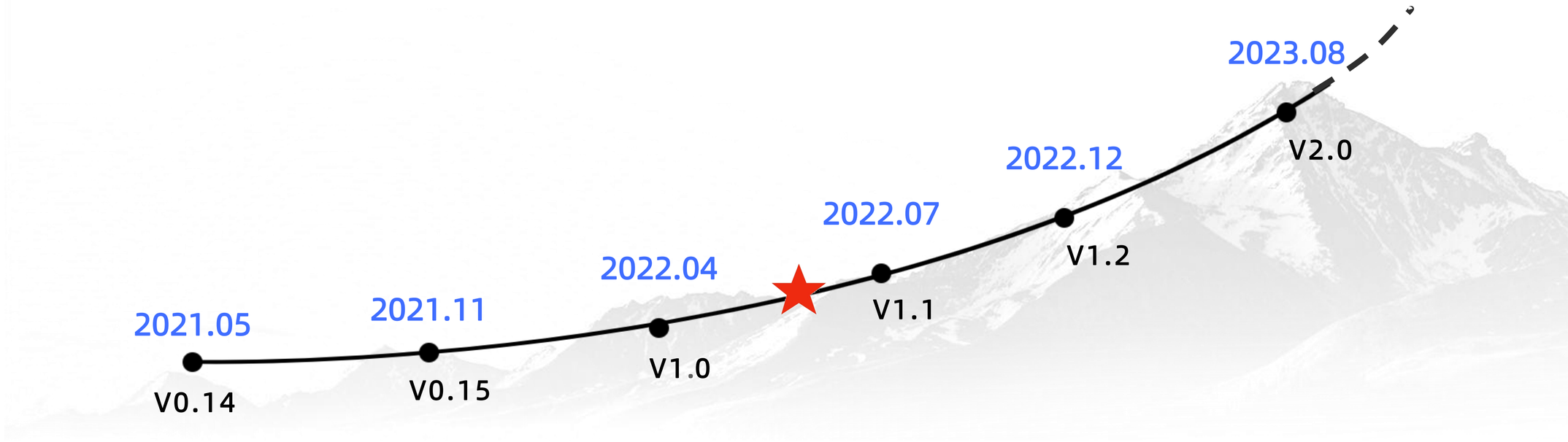
Apache Doris

DORIS

Apache Doris

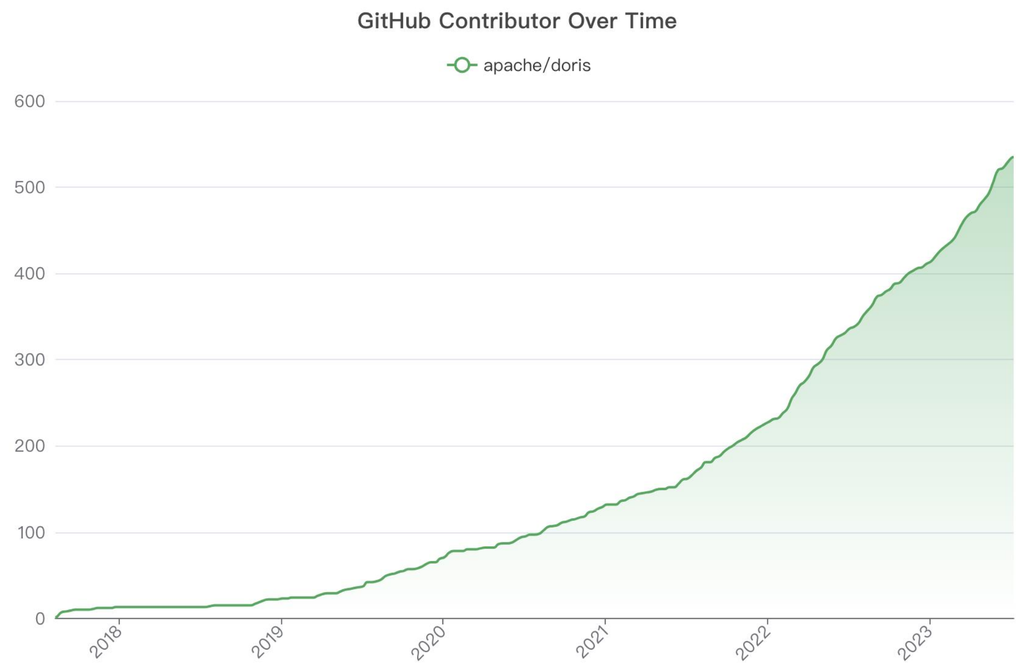
Open-Source
Real-Time
Data Warehouse

2013 Project Creation	2017 Open Source	2022 ★Graduation (ASF Top Level Project)
9400+ GitHub Stars	550+ Contributors	2500+ Enterprises



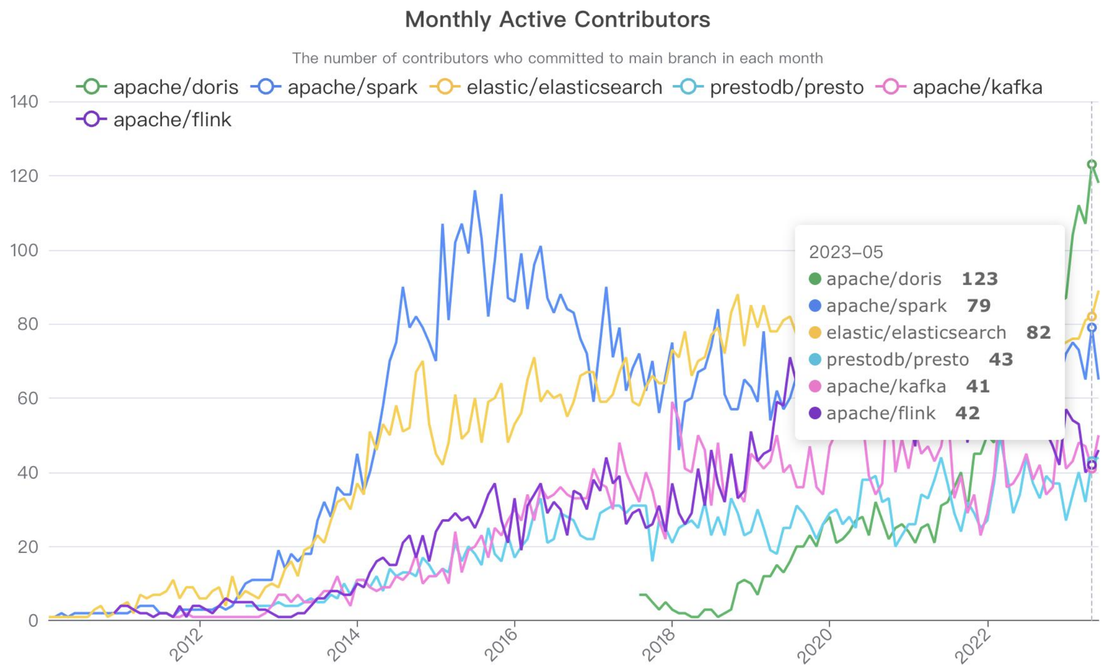
Active Community

Contributors



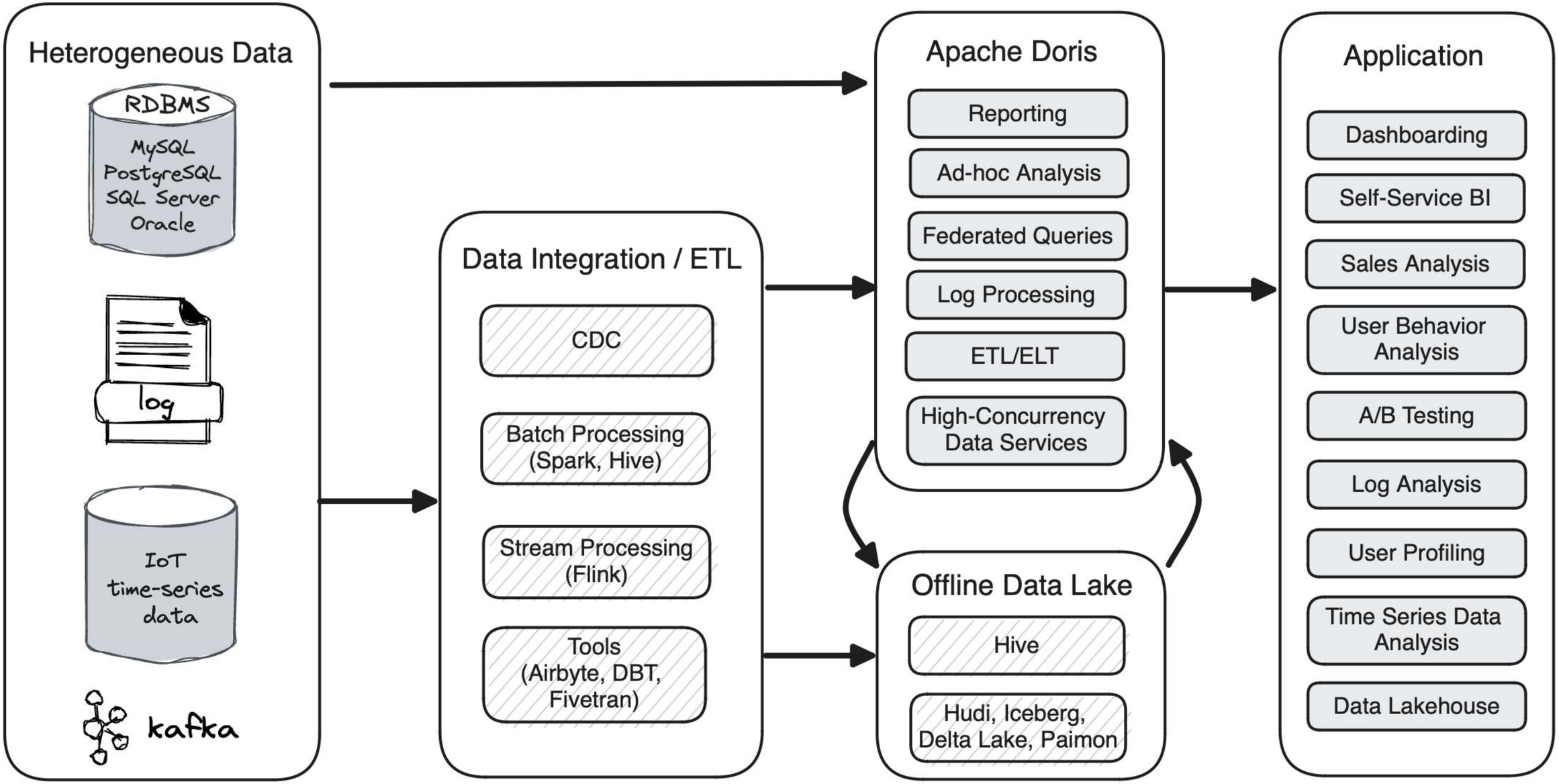
550+

Monthly Active Contributors



Top 1

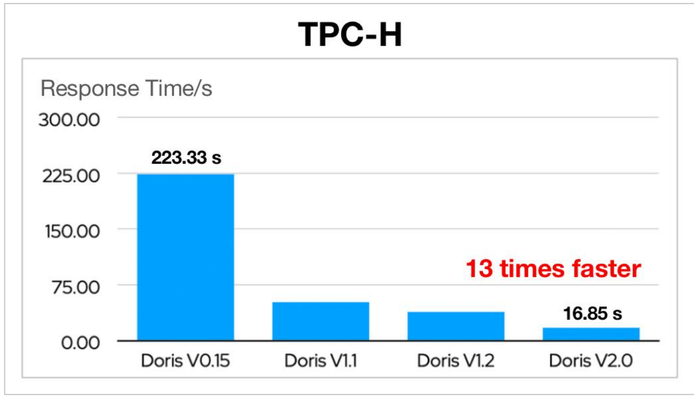
What Does Apache Doris Do?

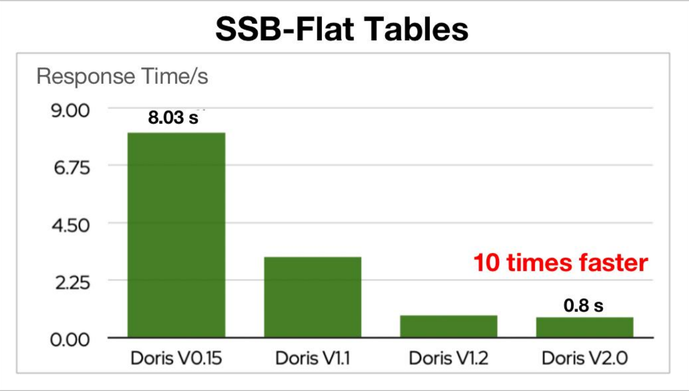


Performance

TPC-H	Doris 2.0	Presto	Greenplum	ClickHouse
Q1	1.57	13.9	30.9	2.74
Q2	0.14	7.31	3.2	108.88
Q3	0.54	12.88	7.9	248.13
Q4	0.26	9.2	8.8	8.54
Q5	1.15	17.89	7.9	235.92
Q6	0.04	4.63	3.9	0.54
Q7	0.73	21.49	11.5	OOM
Q8	0.33	29.9	8.9	OOM
Q9	3.41	60.3	37.8	OOM
Q10	1.34	14.46	7	OOM
Q11	0.1	4.7	1.9	9.71
Q12	0.12	11.17	5.2	OOM
Q13	1.77	10.57	10.4	317.63
Q14	0.14	10.93	3.7	25.38
Q15	0.26	13.46	9.5	1.41
Q16	0.25	3.1	2.3	28.88
Q17	0.12	20.74	71.2	42.32
Q18	2.38	36.28	26	OOM
Q19	0.21	11.34	4.4	54.96
Q20	0.46	14.41	11.7	90.86
Q21	1.06	45.71	16.4	OOM
Q22	0.47	3.58	7.3	10.3
Sum (s)	16.85	377.95	297.8	1186.2

SSB-Flat	Doris 2.0	Presto	ClickHouse
Q1.1	0.02	5.43	0.06
Q1.2	0.01	3.55	0.02
Q1.3	0.03	3.58	0.17
Q2.1	0.08	6.66	0.25
Q2.2	0.08	3.74	0.25
Q2.3	0.06	2.97	0.22
Q3.1	0.16	9.29	0.43
Q3.2	0.07	8.56	0.34
Q3.3	0.06	5.39	0.33
Q3.4	0.01	5.04	0.03
Q4.1	0.14	9.26	0.45
Q4.2	0.05	9.81	0.17
Q4.3	0.03	6.54	0.13
Sum (s)	0.8	79.82	2.85





Test Environment

3 × 16 Core, 64GB cloud virtual machines, SF100

What's Behind the Speed?

DORIS

Cost-Based Optimizer

- Cost-based join reorder, pushdown, choice of RF
- Short circuit plan for high-concurrency queries

PipeLine Execution

- Data-driven, no blocking of threads, fine-grained concurrency
- Self-adjusted parallelism level

Materialized Views

- Consistent single-table materialized views, support general aggregation functions
- Multi-table materialized views (V2.1)

Full Vectorization

- Reduce virtual function calls and cache miss
- Efficient use of SIMD instructions, supports X86 and ARM

Indexes

- Bloom Filter, Min / Max / Sum
- Prefix Sorted Index
- Inverted Index

Smart Caching

- Caching of query results, data, metadata, and intermediate data
- Caching of internal and external tables

Massively Parallel Processing (MPP) Architecture

- Parallelism within and between nodes to give full play to machines and cores
- Supports distributed join of large tables and operator materialization

Columnar Storage & Hybrid Storage

- Columnar storage for efficient encoding, compression, and data sharding
- Row and columnar hybrid storage for flat tables to reduce IOPS amplification

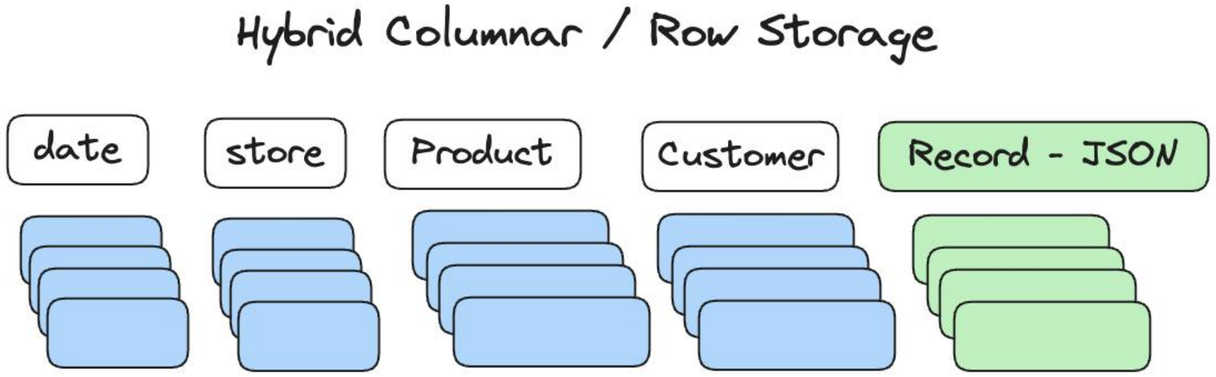
Performance

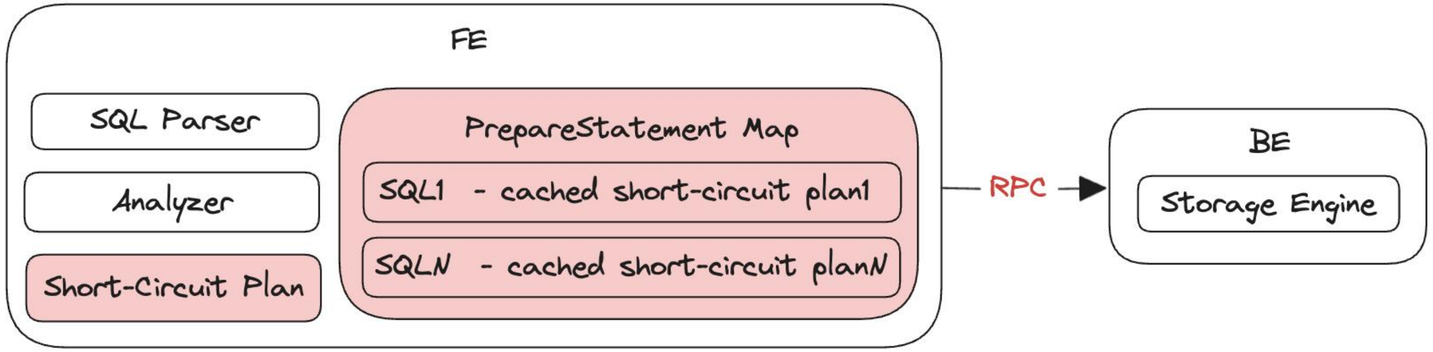
High-Concurrency Data Services

- Use cases
  - Customer-facing: e-commerce delivery checking, etc.
  - Machine-facing: real-time risk control, IoT, etc.

- Eg. select * from user_table where id = xxx
  - For flat tables, columnar storage brings IOPS amplification
  - Unnecessary to use query optimizer for simple queries
  - SQL parsing in frontend, causing bottlenecks

- Solutions
  - Introduce hybrid storage
  - Short circuit plan
  - Introduce Prepared Statement





Data Ingestion

DORIS

- Real-time, high throughput, easy-to-use, read-write separation

- Real-time ingestion (done in 1~5s)
  - **Stream Load**: Load local files to Doris via HTTP; linearly scalable throughput; 10 million rows per second
  - **Flink Doris Connector**: Built-in Flink CDC; Load data from OLTP databases
  - **Routine Load**: Subscribe data from Apache Kafka
  - **Insert into values**: via JDBC
- Batch writing
  - **Spark Load**: Leverage Spark resources to pre-process data from HDFS and object storage
  - **Broker Load**: No need to deploy Broker; supports HDFS and S3 protocol
  - Data integration for data lakehouse: **insert into** <internal table> **select from** <external table>
    - ✓ Storage systems (S3, HDFS, local files)
    - ✓ Data lakes (Iceberg， Hudi， Hive)
    - ✓ Databases (MySQL, Oracle, Elasticsearch, etc.)
- Ecosystem: Spark Connector, Airbyte, Dataworks, DataX, **X2Doris, Parallel Load**
- Read-write separation (V2.1)

Data Update

DORIS

Use cases & requirements

- E-commerce order analysis, user profile update, data deletion, data overwrite

Problems with common data updating mechanisms

- Hive: Update on a partition level
- Iceberg, DeltaLake, Hudi: Merge on Read or Copy on Write, suitable for low-frequency batch updating but not high-frequency real-time updating
- Elasticsearch, OLTP system, HTAP systems: inadequate analytic capabilities and high costs

Common data update types

- Update an entire row (Upsert)
- Partial column update
- Conditional deletion
- Conditional update
- Rewrite a table or partition (insert overwrite)

Data Update

Primary Key Tables (Unique Key tables)

- Merge on Read: Suitable for low-frequency batch updates
- Merge on Write:
  - ✓ **Suitable for real-time writing**
  - ✓ Light merge upon writing, **enabling 5~10 times faster query performance than Merge on Read**
- **Supports most updates: Upsert, partial column update, conditional update/deletion, partition overwrite, etc.**

UPSERT is supported is all data ingestion methods. In concurrent updates, the updating order is decided by that or transaction commits or the Sequence column

Partial column update is supported Stream Load, Update, Insert into values （v2.0.1）

```
UPDATE test SET v1 = v1 + 1 WHERE k1=1

UPDATE t1
    SET t1.c1 = t2.c1, t1.c3 = t2.c3 * 100
    FROM t2 INNER JOIN t3 ON t2.id = t3.id
    WHERE t1.id = t2.id;
```

```
DELETE FROM my_table PARTITIONS (p1, p2)
    WHERE k1 >= 3 AND k2 = "abc";

DELETE FROM t1
    USING t2 INNER JOIN t3 ON t2.id = t3.id
    WHERE t1.id = t2.id;
```

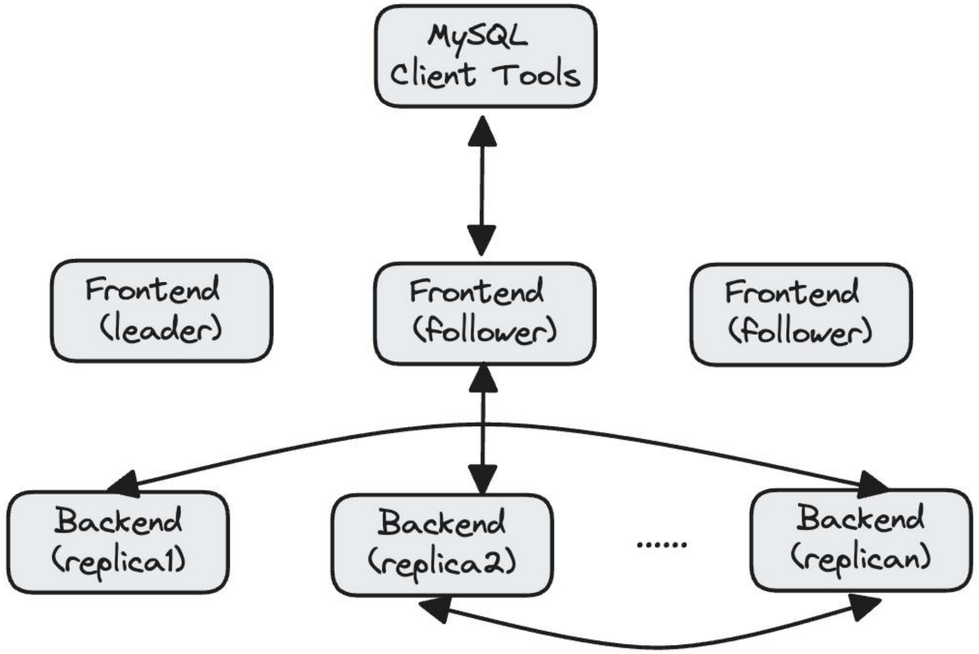
Non-Primary Key Tables (Duplicate tables, Aggregate tables)

- Aggregate tables implements partial column udpate by replace_if_not_null
- All table models supports data deletion based on specified predicates and some date expressions (curdate(), etc.)

```
DELETE FROM my_table PARTITION p1 WHERE k1 = 3;
DELETE FROM my_table PARTITIONS (p1, p2) WHERE k1 >= 3 AND k2 = "abc" AND t = curdate();
```

Service Availability & Data Reliability

- Frontend nodes: metadata management; multi-replica
- Backend nodes: query execution, multi-replica, auto-balancing and restoration
- Client can be linked to frontend nodes via workload balancing
- In Backend downtime, queries will be retried
- Rolling upgrading/scaling



Data Back & Recovery

Supports data backup and restoreation at the level of table/partition (data backup to HDFS or object storage

```
BACKUP SNAPSHOT example_db.snapshot_label2
TO example_repo
ON (
  example_tbl PARTITION (p1,p2),
  example_tbl2 );
```

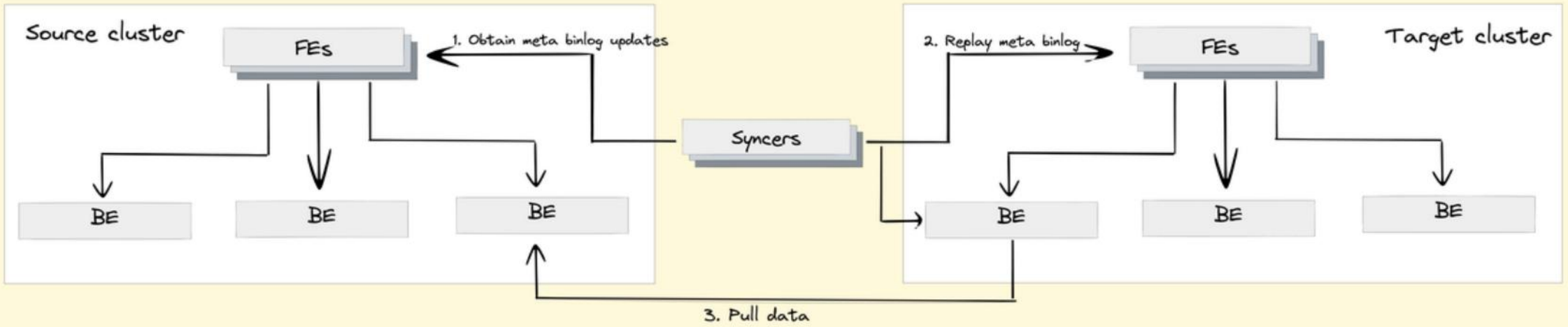
```
RESTORE SNAPSHOT example_db1.`snapshot_1`
FROM `example_repo`
ON ( `backup_tbl` )
PROPERTIES (
  "backup_timestamp"="2022-04-08-15-52-29");
```

Service Availability & Data Reliability

Cross Cluster Replication

- **Disaster recovery**: for quick restoration of data and services
- **Read-write separation**: for higher performance and stability
- **Isolated upgrade of cluster**: pre-creation of backup clusters to prevent interruptions by incompatibility or bugs

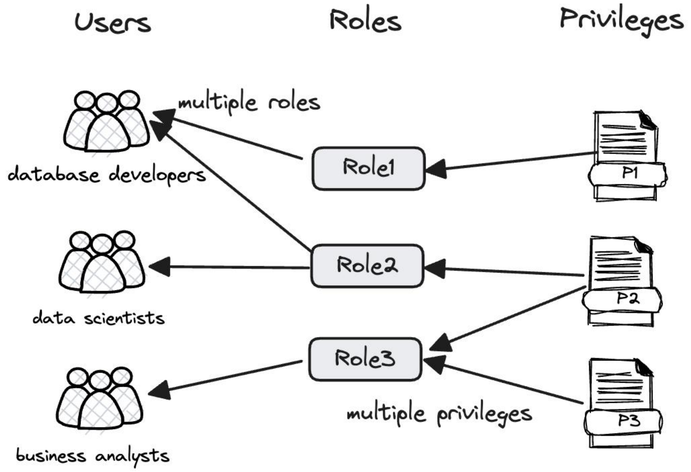
- **Performance**: minute-level latency, reaching the upper speed limit of your NIC and disks



Multi-Tenant Management

Authentication

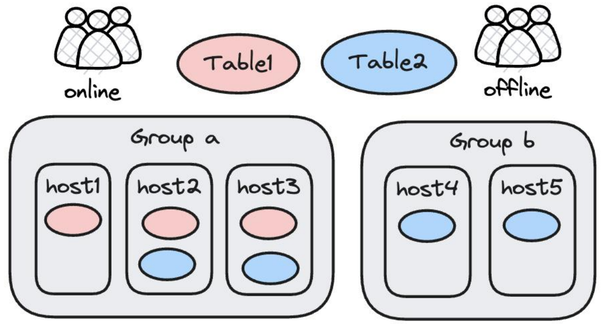
- Role-Based Access Control



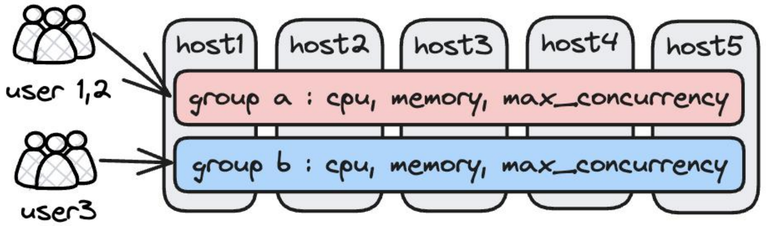
- Granular Privilege Control
  - Privileges on db/tables
  - Privileges on rows
  - Privileges on columns
- Connect to your LDAP for authentication
- Supports SSL/TLS encryption

Resource Isolation

- **Resource Group**: replicas in groups, users bound to groups



- WorkLoad Group (V2.0): soft limit (resource%)



- **Block Rule**: Block certain queries that involve too many tablets/partitions
- **Memory limit on single query (Memory Tracker)**
- **Storage-Compute Separation**: integrate with multi-cluster mechanism (V2.1)
- **Separate data ingestion, compaction threads from queries (V2.1)**

Easy to Use

- **Compatible with MySQL protocol, easy-to-learn, compatible with most BI tools**

- **Light Schema Change**
  - Add or delete fields (within milliseconds)
  - Add or delete indexes
  - Modify column types
- Light Schema Change + Flink-Doris-Connector = Synchronizing DDL from upstream tables within milliseconds

Semi-Structured Data Analysis

DORIS

Use Cases & Requirements

- Log analysis, observability analysis, time series data analysis, security analysis
- Schema-free support, low cost, text analysis, multi-dimensional analysis, full-text search

Text Analysis (Fuzzy Queries- “LIKE”)

- NGram Bloomfilter, 200%↑
- “LIKE” pushdown to storage layer, 200%~300%↑
- Self-adaptive “LIKE”
- The Hyperscan regex matching library, 200%~1000%↑
- Volnitsky algorithm for sub-string matching, 150%~300%↑

```
str LIKE 'abc'          equals(str, 'abc')
str LIKE 'abc%'         starts_with(str, 'abc')
str LIKE '%abc          ends_with(str, 'abc')
str LIKE '%abc%'        sub_string(str, 'abc')
str LIKE '%a%b%c'       regex(str, '.*a.*b.*c')
```

Inverted Index

- Supports tokenization for full-text search

Compound Data Types

- Array, Map, Json; Variant (2.1), different data types in one field

Storage & Writing

- Columnar storage, ZSTD compression, tiered storage for hot and cold data
- Less forward indexing, time-series compaction, single-copy ingestion

Data Lakehouse Capabilities

Scenarios

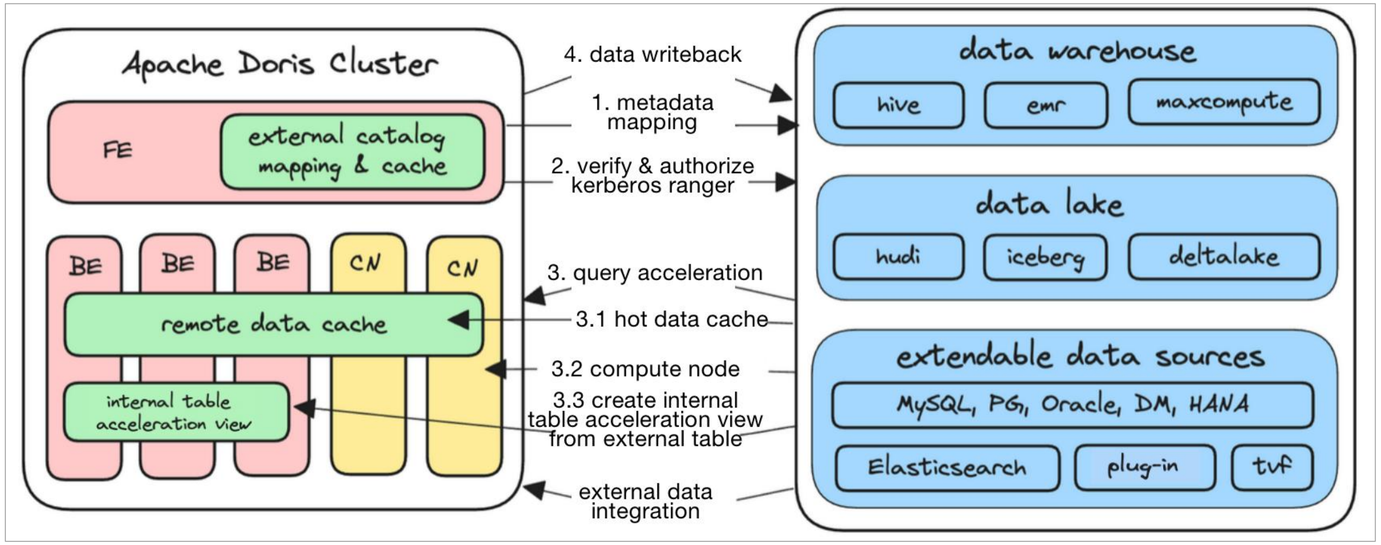
- Faster queries in lakehouse
- Data integration
- Unified query gateway
- ETL/ELT acceleration

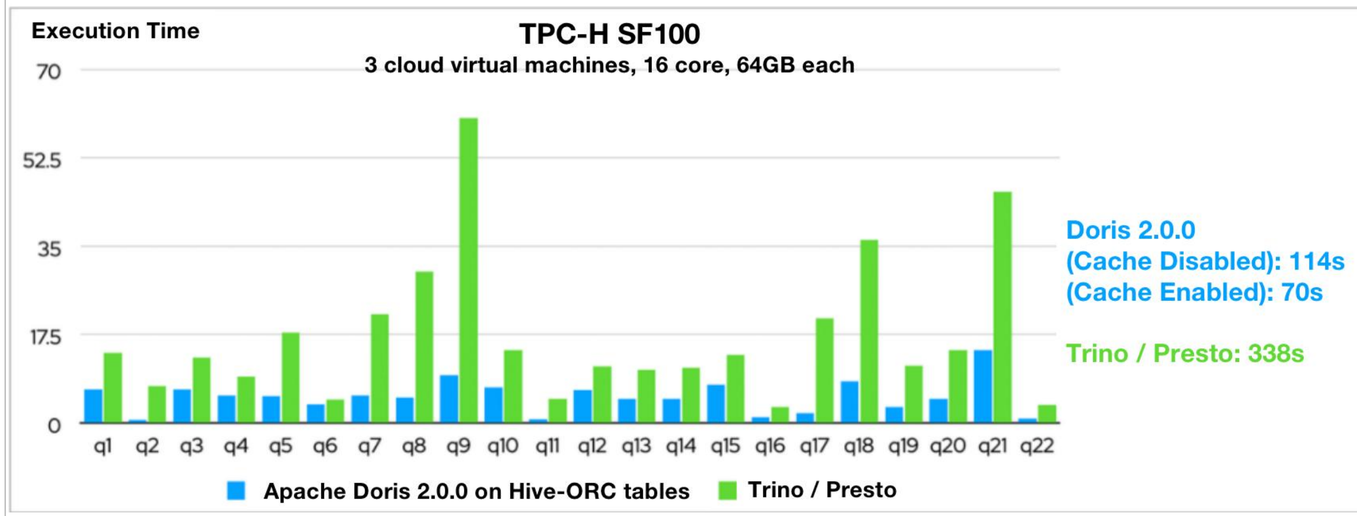
Integration

- Metadata mapping, caching & auto-refreshing
- Supports Hive Meta Store & open formats
- Supports Elasticsearch, relational database, plugins
- Supports Kerberos, Apache Ranger

Query Acceleration

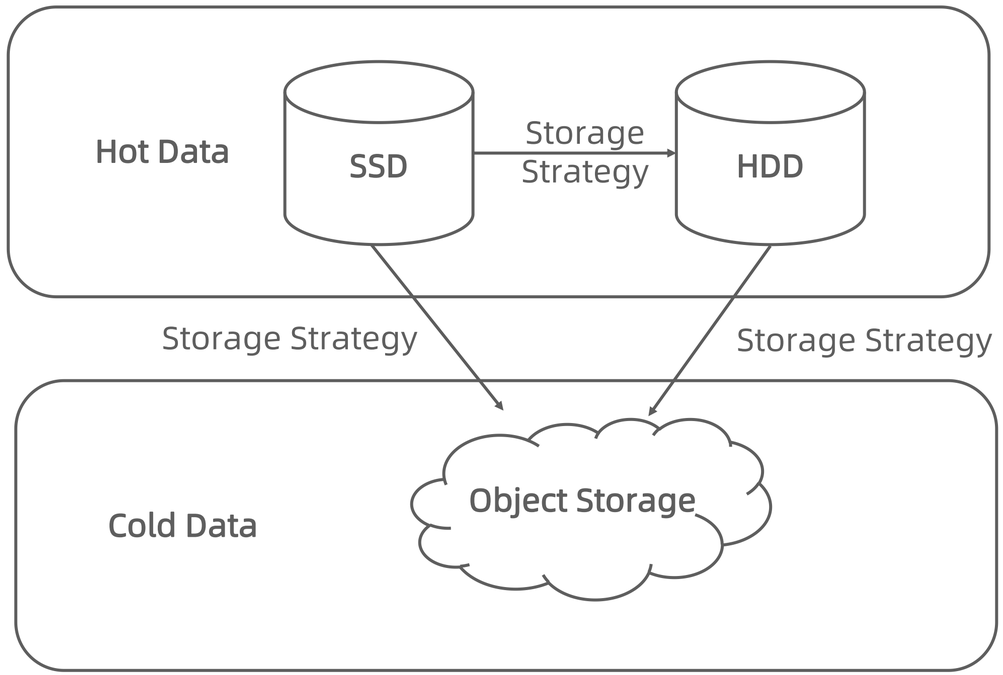
- Accelerated by the Doris engine
- Hot data cached locally
- Compute nodes
- Write query results into Doris to form views



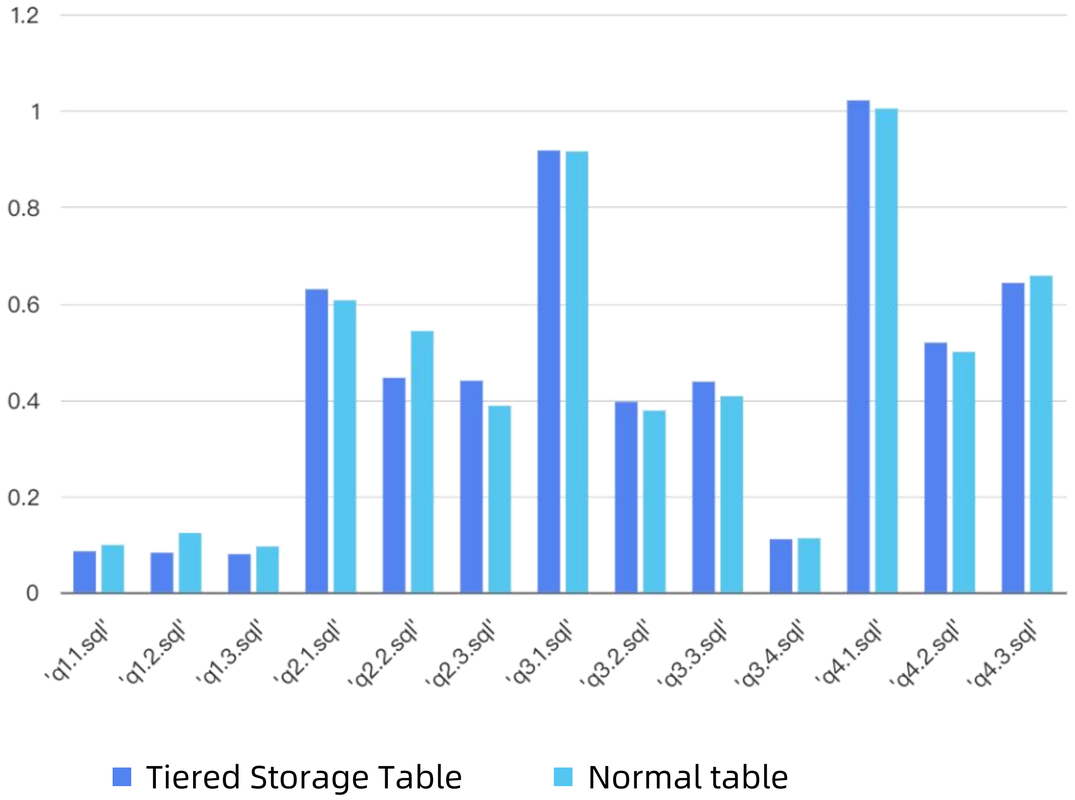


Tiered Storage

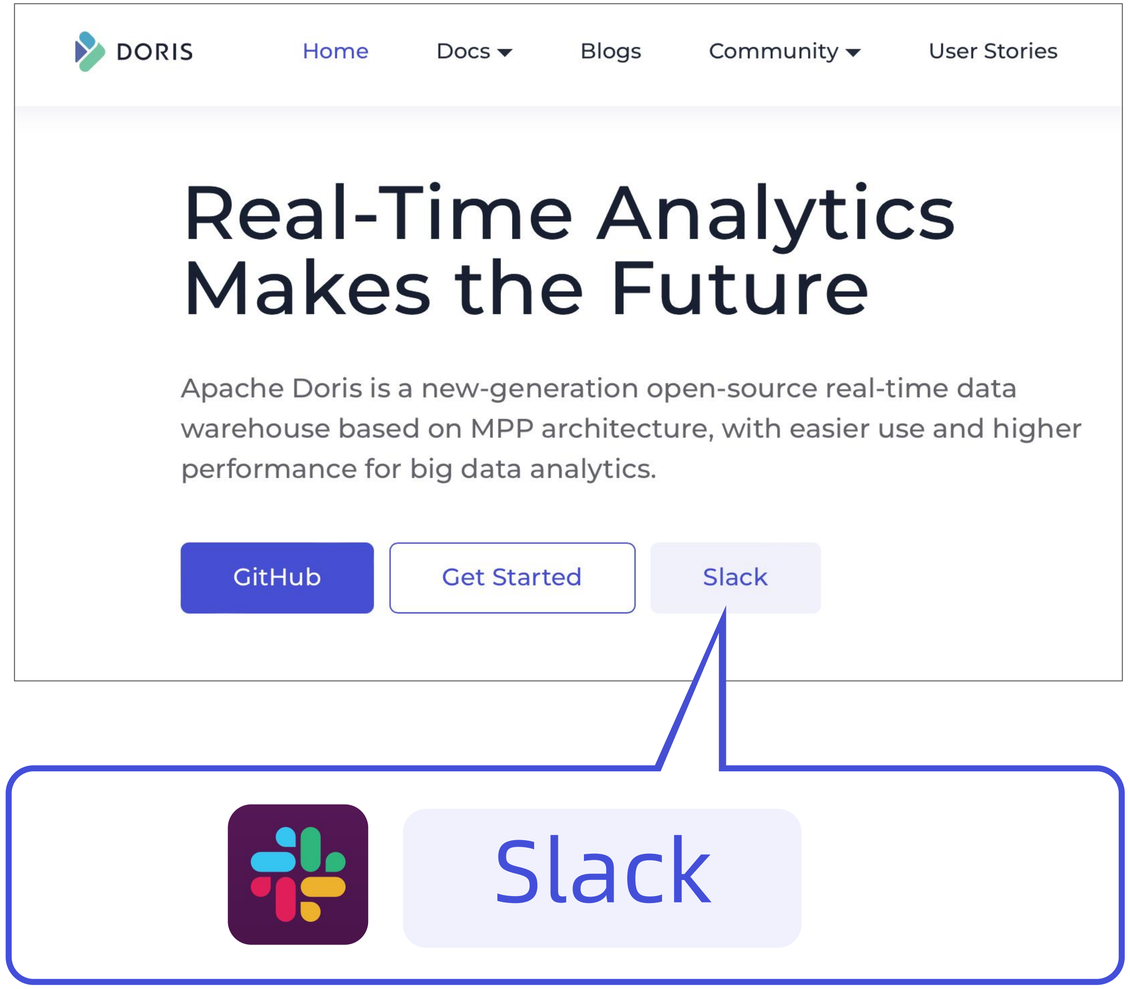
- Hot data in SSD & HDD (fast read/write but expensive)
- Cold data in object storage



Storage Cost 70% ↓



Join the Community



Website
doris.apache.org

GitHub
apache/doris

Twitter
@doris_apache

LinkedIn
Apache Doris

Email Group
dev@doris.apache.org

Medium
Apache Doris