vs CLICKHOUSE | vs ELASTICSEARCH | vs TRINO / PRESTO

01 WHEN TO CHOOSE APACHE DORIS

Doris and ClickHouse are both leading real-time analytical databases. Doris pulls ahead on the workloads modern data teams run every day.

↑ 2-10×

FASTER JOINS

Distributed multi-table joins, no wide-table workarounds.

MPP architecture with a cost-based optimizer that automatically picks Broadcast, Shuffle, or Colocate joins. Doris completes the full TPC-DS suite; ClickHouse fails about 50% of the queries.

↓ 70%

LOWER COST

Storage-compute separation, open-source.

Elastic compute scales without rebalancing. The cost model ships with open-source 3.0+, with no commercial-cloud lock-in.

1000+

CONCURRENT QUERIES

Real-time UPSERTs without sacrificing query speed.

Merge-on-Write keeps reads stable under high-frequency updates. Full ACID, MySQL-compatible, with thousands of concurrent queries instead of fewer than 100.



02 FEATURED CASES

Real production migrations at internet-scale companies running the same SQL workloads, with very different outcomes.

CASE 01 ★ FEATURED

K Kuaishou

SHORT-FORM VIDEO · 600M+ DAU

> After replacing ClickHouse with Apache Doris, Kuaishou successfully upgraded to a unified lakehouse architecture, achieving unified storage and a simplified data pipeline.

- ✓ **Direct lakehouse queries**: no ingestion, shorter pipeline
- ✓ **Materialized view rewriting** for query acceleration across scenarios
- ✓ Flexible **data governance** via automatic materialization

03 CORE DIFFERENCES

Side-by-side, every dimension that matters.

DIMENSION	Apache Doris RECOMMENDED	ClickHouse
01 System Architecture	MPP distributed architecture MySQL-protocol compatible, standard SQL CBO automatic optimization	Scatter-Gather architecture SQL-like syntax, non-standard Requires manual tuning
02 Join Query Performance	2–10× faster, cross-node distributed CBO chooses Join strategy Full TPC-DS suite passes	Subqueries + wide-table modeling No CBO, manual tuning ~50% TPC-DS queries fail

		Efficient memory, avoids OOM	Frequent OOM on large queries
03	Real-time Updates	Merge-on-Write engine Strong consistency, immediate visibility High-throughput UPSERT, no degradation	ReplacingMergeTree, eventual consistency FINAL causes 2–10× slowdown High-frequency updates → merge overhead
04	Transaction Support	Full ACID transactions Atomic batch ingestion	No transaction support Partial data may become visible
05	Query Concurrency	Thousands of concurrent queries, 10×+ Efficient memory management	Usually below 100 concurrent Memory-intensive workloads destabilize cluster
06	Data API & Lakehouse	Arrow-Flight high-throughput protocol Hive / Hudi / Iceberg / Parquet Auto-scaling + multi-replica balancing	JDBC only Limited lakehouse integration Scaling requires manual rebalancing
07	Storage-Compute Separation	Open-source 3.0+ Elastic compute, no rebalance on scale Up to 70% cost reduction	Commercial Cloud edition only Tightly coupled, scaling requires rebalance Over-provisioning required for peaks
08	Open-Source License	Apache Foundation, community-maintained	Controlled by a commercial company

Copyright © 2026 The Apache Software Foundation,Licensed under the Apache License, Version 2.0. Apache, Doris, Apache Doris, the Apache feather logo and the Apache Doris logo are trademarks of The Apache Software Foundation.