CloudTable Service

Service Overview

Issue	01
Date	2026-01-27

HUAWEI

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Cloud Computing Technologies Co., Ltd.

Address: Huawei Cloud Data Center Jiaoxinggong Road
Qianzhong Avenue
Gui'an New District
Gui Zhou 550029
People's Republic of China

Website: https://www.huaweicloud.com/intl/en-us/

Contents

1 What Is CloudTable

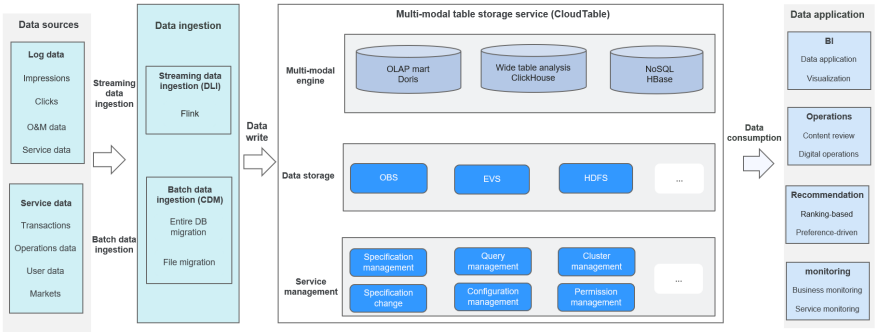
CloudTable is a fully managed data storage and analysis service based on HBase, Doris, and ClickHouse. It provides data storage and analysis capabilities ranging from gigabytes to petabytes, catering to scenarios such as online queries, data warehousing, data marts, and real-time analysis. Widely applicable across industries, CloudTable serves the Internet, Internet of Things (IoT), Internet of Vehicles (IoV), finance, government, logistics, manufacturing, and retail sectors. CloudTable provides you with dedicated clusters, which are available upon subscription. It is suitable for users with high service throughput and low latency requirements.

- CloudTable offers HBase-based fully managed NoSQL services, delivering lightning-fast random read/write capabilities within milliseconds. It is an excellent choice for storing vast amounts of structured, semi-structured, spatiotemporal, and time series data. Whether in IoT, IoV, finance, smart cities, or meteorology, CloudTable finds versatile applications. For details, see **HBase**.
- Based on Doris, CloudTable provides fully managed real-time data warehouse services and query results of mass data can be returned in subseconds. Doris supports high-concurrency point queries and high-throughput complex analysis. Doris is suitable for report analysis, ad hoc query, unified data warehouse construction, and data lake federated query acceleration. You can build applications such as user behavior analysis, A/B test platform, log retrieval and analysis, user profile analysis, and order analysis. For details, see **Doris**.
- ClickHouse is an open-source columnar database oriented to online analysis and processing. It is independent of the Hadoop big data system and features compression rate and fast query performance. In addition, ClickHouse supports SQL query and provides good query performance, especially the aggregation analysis and query performance based on large and wide tables. The query speed is one order of magnitude faster than that of other analytical databases. For details, see **ClickHouse**.

Architecture

The following figure shows the architecture of CloudTable.

Figure 1-1 Architecture



- Doris: MySQL-compatible database that is suitable for complex multi-table analysis, outperforming traditional MPP databases.
- ClickHouse: supports multidimensional aggregation analysis for large and wide tables (with 10,000 columns), subsecond-level response, and full self-service analysis.
- HBase: supports high concurrency and millisecond-level query response.

Advantages

- Wide applicability: CloudTable is compatible with multiple engines (such as HBase, Doris, and ClickHouse).
- High reliability: The architecture is highly available and the kernel is optimized to improve system stability.
- Cost-effectiveness: Cold and hot data can be segregated, utilizing different compression algorithms to reduce storage costs.
- Easy to use: Analysis clusters can be swiftly established through the console within minutes. The platform offers comprehensive cluster management, monitoring, and alarm reporting functions. With robust SQL statement support, you can focus on extracting value from your data without concerning yourself with the underlying infrastructure.

New to CloudTable

If you are new to CloudTable, consider exploring the following information:

- Basic knowledge

  The "Functions" section covers the fundamental principles and scenarios of CloudTable components, along with specific concepts and functions related to CloudTable.

- Getting started

  The "Getting Started" part provides detailed operational guidance using practical samples. You can create and utilize CloudTable clusters based on this guide.

- Advanced features and operation guides

  If you are an existing CloudTable cluster user, refer to the *User Guide* to create clusters, configure parameters, and monitor alarms.

If you are a developer, you can develop, run, and debug your applications according to the Application Development Process and sample project provided by CloudTable.

2 Advantages

Simple, Versatile, and Easy to Use

- Wide applicability: CloudTable is compatible with multiple engines (such as HBase, Doris, and ClickHouse).
- Diverse specifications: A comprehensive range of specifications are available to accommodate the unique demands of different application scenarios.
- Simplicity of use: CloudTable is a fully managed service that allows for the construction of analysis clusters in minutes via the console, freeing you from infrastructure configurations.
- Auto scaling: You can quickly change specifications, horizontally expand capacity, and expand disk capacity directly through the console.

High Reliability and Low Costs

- Multiple data copies: The multi-copy mechanism ensures high reliability.
- Low costs: CloudTable supports cold-hot separation and diverse compression algorithms, reducing storage costs.

Strong O&M

- Monitoring: CloudTable supports a comprehensive suite of monitoring metrics designed to precisely measure cluster status.
- Alarm: Featuring a robust multi-dimensional health check, it ensures that timely alerts are sent upon the detection of any suboptimal health conditions.

Superior Performance

- HBase: Offers petabyte-scale key-value data storage, with write throughput of hundreds of millions data records, and data query response in milliseconds.
- Doris: Delivers unparalleled real-time performance, with high-concurrency data access and querying capabilities, surpassing traditional MPP databases.

3 Application Scenarios

Migration of Big Data Services and Platforms to the Cloud

- Scenario: You can smoothly migrate customers' on-premises/cloud big data platforms to CloudTable to quickly build their data service systems for rapid service growth.
- Advantages
  - Ease of use: Out-of-the-box availability frees you from dedicated deployment, reduces the O&M workload, and enables you to focus on service delivery, and facilitates big data application analysis.
  - On-demand resource scalability: You can flexibly choose and scale compute and storage resources, adjust cluster nodes, specifications, and storage on demand.
  - Open source compatibility: CloudTable is fully compatible with open source APIs, ensuring seamless service integration and zero code change.
  - Superior performance: CloudTable experiences significantly faster real-time data queries compared to traditional data warehouses.
  - Security and reliability: CloudTable enjoys independent cluster deployment, VPC network isolation, and encrypted data channels for top-tier security and reliability.

Storage and Query of Message Logs

- Scenario: CloudTable HBase is designed for the rapid querying of message and log data, efficiently delivering results to applications. Structured and semi-structured key-value data can be stored and queried, including messages, reports, recommendation data, risk control data, logs, and orders.
- Advantages
  - Mass storage: CloudTable offers robust support for both offline and online storage, accommodating vast amounts of key-value data with a scalable storage framework.
  - High-performance read/write: CloudTable delivers tens of millions in write throughput and millisecond-level response time for queries, ideal for powering online applications and generating instantaneous report displays.

  - Vibrant ecosystem: Based on a wide variety of components of the Hadoop ecosystem, CloudTable can enhance its capabilities through integration with cloud products.

Real-time Report Query and Analysis

- Scenario: CloudTable Doris efficiently aggregates data from source service systems via real-time and batch processing. This enables downstream applications to conduct comprehensive multi-dimensional analyses, yielding report results in subseconds.
- Advantages
  - Real-time data write: Capable of collecting data from multiple data sources, CloudTable excels in writing millions of data lines per second, ensuring a continuous flow of real-time information.
  - Subsecond-level query response: CloudTable distributes queries across different buckets for processing. It uses point query indexes (primary keys and inverted indexes) to reduce the amount of data to be read and to provide concurrent query. It uses materialized views and pre-aggregated results to provide sub-second aggregation and statistics analysis.
  - High stability and availability: With a resilient infrastructure of multi-copy and multi-node cluster deployment, CloudTable offers flexible capacity scaling and automated load balancing to maintain optimal performance.

4 HBase

4.1 HBase Basic Principles

HBase Overview

HBase is a column-oriented distributed cloud storage system that features enhanced reliability, excellent performance, and elastic scalability. It applies to the storage of massive amounts of data and distributed computing. You can use HBase to build a storage system capable of storing terabytes to petabytes of data. With HBase, you can filter and analyze data with ease and get responses in milliseconds, rapidly mining data value.

HBase applies to the following scenarios:

- Mass data storage

  HBase applies to TB- or even PB-level data storage and provides dynamic scaling capabilities so that you can adjust cluster resources to meet specific performance or capacity requirements.

- Real-time query

  The columnar and key-value storage models apply to the ad-hoc query of enterprise user details. The primary key–based low-latency point query reduces the response latency to seconds or even milliseconds, facilitating real-time data analysis.

For details about HBase architecture and principles, visit **https://hbase.apache.org/book.html**.

HBase Principles

HBase is suitable for storing PB-level structured and semi-structured data, which may experience irregular explosive growth in a short period of time. HBase is a data storage engine designed for massive datasets and provides powerful key-value query capabilities. It supports tens of millions of concurrent throughput and millisecond-level access latency, meeting the service requirements of Internet enterprises for BI reports, online monitoring, and interactive analysis.

Figure 4-1 HBase architecture

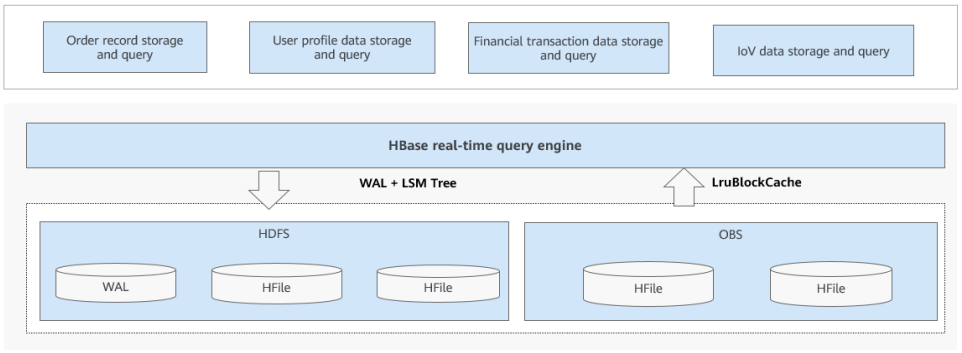


Table 4-1 Modules in the HBase architecture

Item	Description
HBase real-time query engine	HBase is a distributed, column-oriented database with high reliability and linear scalability. HBase is built on top of the Hadoop Distributed File System (HDFS) to store massive amounts of data. It provides millisecond-level point query and range scan performance through a tree structure and sharding mechanism.
WAL + LSM Tree	The Log-Structured Merge (LSM) Tree, in conjunction with Write-Ahead Log (WAL), constitutes the fundamental storage architecture enabling high-performance writes in HBase. The system design optimizes the throughput of massive data. When data is written, it is first buffered in an in-memory structure called MemStore. Once the MemStore reaches a certain size threshold, data is sequentially written to HFiles in batches. This prevents frequent random input and output of small files.
LruBlockCache	HBase utilizes the LruBlockCache as its default in-memory caching mechanism. It manages HFile data blocks (such as index blocks and data blocks) in memory. The LruBlockCache categorizes blocks into three priority levels: SINGLE, MULTI, and MEMORY. This mechanism optimizes read performance and minimizes disk I/O.
HDFS	HDFS is a distributed file storage system of HBase. It provides high reliability, high performance, column storage, scalability, and real-time reading and writing.
OBS	Object Storage Service (OBS) provides secure, reliable, high-performance, and low-cost object storage for you to store massive amounts of data.
WAL	Write-Ahead Logging (WAL) ensures user data security in the event of a RegionServer crash.

Item	Description
HFile	HFile defines the storage format of StoreFiles in a file system. HFile is the underlying implementation of StoreFile.

Advantages

- Native HBase APIs: CloudTable HBase is designed to be compatible with native HBase APIs, ensuring high availability of the architecture through the separation of compute and storage for enhanced reliability, along with in-depth kernel optimization.
- Ease of use: Secondary indexes are supported to meet non-primary key query requirements.
- Low costs: Cold and hot data can be segregated to fulfill the needs of data archiving and the storage of historical data with infrequent access, thereby minimizing storage expenses.
- Stability and Reliability: CloudTable HBase provides stable and reliable performance through hotspot diagnosis and self-healing mechanism.
- Visualized monitoring and O&M: CloudTable HBase offers visualized monitoring and user-defined alarm rules, simplifying system operation and maintenance.
- High compatibility: CloudTable HBase is compatible with native HBase APIs. The NoSQL engine supports typical interface protocols in the industry.
- SLA assurance: Stable TPS and latency are achieved.
- High availability: With HMaster HA and region transfer within seconds, read and write operations are not affected even if there are faulty disks. The dual-read mechanism helps achieve more stable service level agreement (SLA).

4.2 HBase Application Scenarios

Storage and Query of Message Logs

Application Scenarios:

Structured and semi-structured key-value data can be stored and queried, including messages, reports, recommendation data, risk control data, logs, and orders.

Advantages:

- Mass storage

  Offline and online storage of massive volumes of key-value data, and flexible capacity expansion

- High-performance read/write

  100-million-level write throughput, millisecond-level query latency for presenting online applications and reports

- Enriched ecosystem

  A large number of Hadoop ecosystem components, integrated with products

Profile Storage and Query

Application Scenarios:

Labels are used to describe characteristics of people and objects. Each person or object has a set of labels that are uncertain because data is frequently updated. This type of data is widely used in marketing decision-making, recommendation, and advertising systems.

Advantages

- Sparse matrix

  The sparse matrix model of HBase is suitable for storing unstructured data. No schema needs to be predefined for tables and no strict column definition is required among rows.

- Easy update

  You can update any rows at any time without performance loss. HBase built-in versioning mechanism is used to save multiple historical versions of data.

Storage and Queries of Mass Key-Value Data

- Data types

  Structured and semi-structured key-value data, including messages, reports, recommendation data, risk control data, logs, and orders.

- Application scenarios

  CloudTable allows high-speed write of mass online and offline key-value data and low-latency data queries. It applies to online applications or report display. CloudTable can easily be scaled to achieve HA and low-latency storage and queries of vast amounts of data.

4.3 HBase Enhanced Features

Global Indexes

- Index data stored in an independent table: Index data is synchronized to an independent index table. The index table is distributed in an independent region to decouple user data tables and enhance region stability.
- Index query link optimization: Index data is sorted by index field. Users only need to query one or more consecutive regions in the index table. User data that contains redundant columns in an index table can be closed on the index table during index query. You do not need to query the user table.

Off-Heap Cache

Caching data blocks off-heap ensures that data does not need to be copied to the heap memory during data reading. This reduces performance glitches to ensure query stability.

Tries Index

DataBlockIndex is optimized. Indexes can be converted into prefix trees to reduce the memory occupied by duplicate prefixes. In addition, LOUDS-Sparse is used to encode data into a linear structure. Memory space saved can be used to cache more data. This reduces data read I/Os and latency, improves the overall throughput. After optimization, memory consumption was reduced by 80%.

HBase Dual-Read

It is difficult to ensure 99.9% query stability in HBase storage due to reasons such as GC, network jitter, and bad disk sectors. The HBase dual-read feature is introduced to meet the requirements for random read of mass data. The HBase dual-read feature is based on the DR capability of the active and standby clusters. The probability that the two clusters generate glitches at the same time is far less than that of one cluster. The dual-cluster concurrent mode is used to ensure query stability. When a user initiates a query request, the HBase service of the two clusters is queried at the same time. If the active cluster does not return any result after a period of time (the maximum tolerable glitch time), the data of the cluster with the fastest response can be used.

HBase Cold and Hot Data Separation

Data is written based on timestamps. Hot data is typically accessed. Cold data is stored on OBS for lower storage costs. Hot data is stored on cloud disks for shorter query latency. Overall, this greatly reduces costs for storing cold data and improves hot data access performance.

Self-Healing from Hotspotting

Self-healing from HBase hotspotting is an automatic adjustment mechanism introduced to alleviate system performance deterioration caused by overload on some nodes due to uneven data access. HBase is a distributed key-value database. Regions are the smallest units of data management. Poorly designed tables or improperly planned row keys make requests directed at a few fixed regions. As a result, the service pressure is high on a single node, causing performance deterioration or even request failures. Therefore, the core capabilities below are required for self-healing from HBase hotspotting:

- Monitoring the request traffic of each node and performing aggregation analysis to quickly identify heavily loaded nodes and hotspot regions for troubleshooting.
- Automatically diagnosing based on the collected information and performing region splitting to balance the request pressure through automatic adjustment in hotspot areas.
- Intelligently identifying hot row keys and offering the corresponding traffic limiting mechanisms to form a complete closed-loop solution for hot issues.

5 Doris

5.1 Doris Basic Principles

Doris Overview

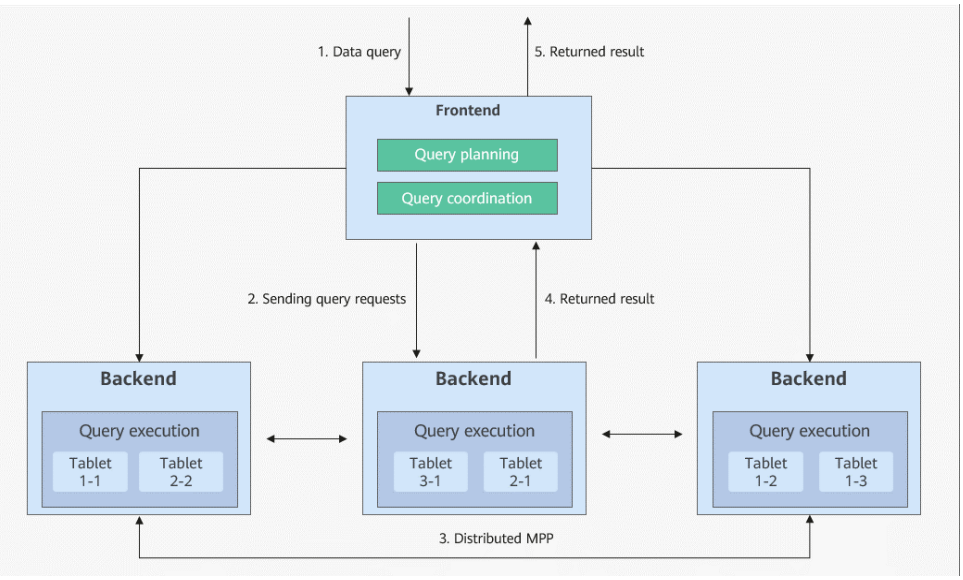
Doris is a high-performance, real-time analytical database based on MPP architecture. It can return query results of mass data in sub-seconds and can support high-concurrency point queries and high-throughput complex analysis. All this makes Doris an ideal tool for report analysis, ad-hoc query, unified data warehouse, and data lake federated query acceleration. On Doris, users can build various applications, such as user behavior analysis, AB test platform, log retrieval analysis, user portrait analysis, and order analysis.

Doris, formerly known as Palo, was initially created to support ad reporting business. Currently, the Apache Doris community has gathered more than 300 contributors from hundreds companies in different industries, and the number of active contributors is close to 100 per month. In June 2022, Apache Doris graduated from Apache incubator as a Top-Level Project. Doris now has a wide user base in China and around the world. Doris has been used in the production environment of more than 500 enterprises worldwide. Of the top 50 Chinese Internet companies by market capitalization (or valuation), more than 80% are long-term users of Doris. It is also widely used in some traditional industries such as finance, energy, and manufacturing.

Doris Architecture

Doris uses the MPP architecture to query data. Data is concurrently queried between nodes and within a node. Distributed shuffle join of multiple large tables is supported. The performance of multi-table joint query is excellent, which can better cope with service query in various complex scenarios.

Figure 5-1 Doris architecture



The overall architecture of the Doris engine is very simple. There are two types of processes:

- Frontend nodes process user access requests, plan query parsing, and manage metadata and nodes.
- Backend nodes are responsible for both storing data and executing query plans.

Both types of processes can be scaled out horizontally. Nodes in a single cluster can be flexibly scaled, and the storage capacity can be increased to dozens of petabytes. In addition, the two types of processes use a consistency protocol to ensure high service availability and data reliability. This highly integrated architecture reduces O&M costs.

Advantages

- High performance: Doris is equipped with an efficient column storage engine, which not only reduces the amount of data scanning, but also implements an ultra-high data compression ratio. At the same time, Doris also uses various index technologies to speed up data reading and filtering. Using the partition and bucket pruning function, Doris can support ultra-high concurrency of online service business, and a single node can support up to thousands of QPS. Further, Doris combines the vectorized execution engine to give full play to the modern CPU parallel computing power. Doris supports materialized view to accelerate pre-aggregation, and uses the query optimizer to optimize queries based on planning and costs.
- Ease of use: CloudTable Doris adheres to standard ANSI SQL syntax, encompassing single-table aggregation, sorting, filtering, multi-table joins, subqueries, and advanced SQL constructs like window functions and GROUPING SETS. In addition, it is also compatible with MySQL protocol, which allows users access Doris through various BI tools.
- Simple architecture: Doris has only two types of processes, that is, Frontend (FE) and Backend (BE). The FE node is responsible for user request access, query plan parsing, metadata storage, and cluster management. The BE node

is used to store data and execute query plans. Doris can function as a complete distributed database management system and users can run the Doris cluster without installing any third-party management and control components. In addition, both FE and BE nodes support horizontal expansion. A cluster can be expanded to hundreds of nodes and can store more than 10 petabytes of data.

- Stability and reliability: Data can be stored in multiple copies and Doris clusters are capable of self-healing. Its distributed management framework can automatically manage the distribution, repair, and balancing of data copies. When a data backup is damaged, the system can automatically detect the damage and repair it.
- Rich ecosystem: Doris provides rich data ingest methods, supports fast loading of data from localhost, Hadoop, Flink, Spark, Kafka, SeaTunnel and other systems, and can also directly access data in MySQL, PostgreSQL, Oracle, S3, Hive, Iceberg, Elasticsearch and other systems without data replication. At the same time, the data stored in Doris can also be read by Spark and Flink, and can be output to the upstream data application for display and analysis.
- Flexible billing: For long-term stable services, you can purchase compute and cache resources in yearly/monthly mode. For temporary and ever-changing services, you can purchase compute and cache resources in pay-per-use mode. By default, storage resources are billed based on the actual data volume.

5.2 Doris Application Scenarios

Based on Doris, CloudTable provides fully managed real-time data warehouse services and query results of mass data can be returned in subseconds. Doris supports high-concurrency point queries and high-throughput complex analysis. All this makes Doris an ideal tool for report analysis, ad-hoc query, unified data warehouse, and data lake query acceleration. On Doris, users can build various applications, such as user behavior analysis, AB test platform, log retrieval analysis, user portrait analysis, and order analysis.

Application Scenarios

- Reporting analysis
  - Real-time dashboards
  - Reports for in-house analysts and managers
  - Highly concurrent user-oriented or customer-oriented report analysis (customer-facing analytics): such as website analysis and ad reporting that usually require thousands of QPS and quick response times measured in milliseconds. A successful user case is that Doris has been used by an e-commerce company in ad reporting, where it receives 10 billion rows of data per day, handles over 10,000 QPS, and delivers a 99th percentile query latency of 150 milliseconds.
- Ad-hoc query. Analyst-oriented self-service analytics has irregular query patterns and needs high throughput requirements. Doris helps build growth analytics platforms (Growth Analytics, GA), using user behavior data for business growth analysis, with an average query latency of 10 seconds and a 95th percentile query latency of 30 seconds or less, and tens of thousands of SQL queries per day.

- Unified data warehouse construction. Doris allows users to build a unified data warehouse via one single platform and save the trouble of handling complicated software stacks. A unified data warehouse with Doris simplifies the architecture by replacing its old complex one that consisting of Spark, Hive, Kudu, HBase, and Phoenix.
- Data lake federated query. Doris avoids data copying by federating the data in Hive using external tables, and thus achieves outstanding query performance.

5.3 Doris Enhanced Features

Vectorized Computing

A vectorized engine is a query technology used by the data storage engine. Vectorization transforms operations that would traditionally be applied one-by-one to individual data elements (individualized operations) into a single operation applied to a "vector". Vector calculations are indeed implemented by leveraging specialized hardware within the CPU, specifically vector registers. This leads to faster query execution of databases. Thanks to the optimization of column-based storage and vectorized executor, the query execution speed will be improved by three to five times.

Column Pruning

Column pruning means that only required columns are read during query, and unnecessary columns are ignored, reducing data read overhead. When you query large tables containing numerous columns, you can use column pruning to exclude columns that are not required to reduce the amount of data to be read. This not only reduces the overhead of network transmission, but also reduces the overhead of storage on memory and disks. In addition, column pruning can improve the query response speed for better query performance, especially when processing massive data.

Pipeline Engine Optimization

The pipeline execution engine is built upon multi-core CPUs and is a data-driven execution engine. It fully leverages the computing power of multi-core CPUs, and limits the number of Doris query threads, resolving the problem of Doris execution thread bloat.

6 ClickHouse

6.1 ClickHouse Basic Principles

ClickHouse Overview

ClickHouse offers easy-to-use, flexible, and stable hosting services in the cloud. A data warehouse can be created in minutes for massive real-time data query and analysis, improving the overall efficiency of data value mining. By leveraging the massively parallel processing (MPP) architecture, ClickHouse can query data several times faster than traditional data warehouses.

ClickHouse Principles

ClickHouse is a columnar, distributed database management system (DBMS) that mainly applies to Online Analytical Processing (OLAP) workloads. It can generate analysis data reports in real time using SQL queries, and allows you to create tables and databases, load data, and run queries during runtime. ClickHouse is simple, reliable, and fault-tolerant. ClickHouse achieves high-performance data querying through a combination of features such as the MPP architecture, distributed computing, and historical storage.

Figure 6-1 ClickHouse architecture

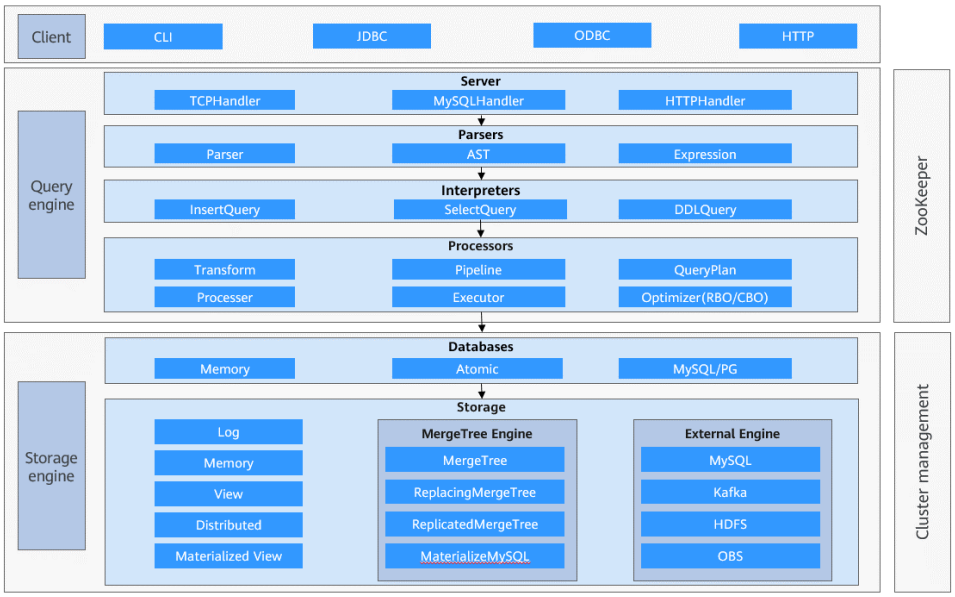


Table 6-1 Modules in the ClickHouse architecture

Item	Description
MergeTree	The MergeTree engine is the most commonly used storage engine in ClickHouse and applies to scenarios requiring efficient data insertion and query. Data is initially written in multiple parts and then merged into larger ones to optimize query performance. Partitioning, sorting keys, primary keys, and secondary indexes are supported.
Query engine	ClickHouse leverages the MPP architecture and vectorized execution engine to significantly improve the performance of complex analysis and query. After a task is delivered to a cluster, the execution process consists of three parallel levels: • Parallel query on cluster nodes • Collaborative computing between multiple replicas • Multi-core CPU parallel computing within a node and parallel computing between independent operators
Storage engine	ClickHouse employs column-oriented storage to store data of the same type by column. This structure is suitable for analytical scenarios. It helps ClickHouse to efficiently read columns required by queries, reducing overhead and improving query performance.
Materialized view	A materialized view stores the results of a query as a physical table to optimize the query performance. By storing precomputed and complex query results, a materialized view eliminates the need for repeated computation. Unlike regular views, materialized views store query results persistently and can be directly accessed to improve query efficiency.

Item	Description
RBO	A rule-based optimizer (RBO) uses predefined logical rules and physical rules to convert and optimize query plans. These rules restructure the execution logic while ensuring the correctness of query results, significantly improving query performance.
CBO	A cost-based optimizer (CBO) is a core component of the query optimizer. It dynamically selects the optimal execution plan using analysis statistics. Compared with a RBO, a CBO can optimize complex queries more accurately, especially in key scenarios such as join policy selection, aggregation calculation optimization, and predicate pushdown.
Atomicity	Atomicity ensures that a transaction is treated as an indivisible work unit, meaning that either all of its operations are successfully committed, or none of them are applied, and the system is rolled back to its original state. There is no partial execution state.
DDL	Data Definition Language (DDL) is a standardized language used to create, modify, and manage database structures. It defines the logical framework of data entities (such as tables, views, and indexes) and data schema.
Index	ClickHouse optimizes query performance through an indexing mechanism. Based on the vertical pruning of column-oriented storage, ClickHouse supports partition keys, primary keys, and multiple secondary indexes. These index structures efficiently locate data blocks horizontally, quickly filter out irrelevant data blocks, and significantly reduce the disk scanning scope, thereby improving query efficiency.

Advantages

- High performance: ClickHouse employs column-oriented storage. This means data of the same type is stored into the same column, bringing a higher data compression ratio. Generally, the compression ratio can reach 10:1, significantly reducing storage costs and read overhead, and improving query performance.
- Replication mechanism: ClickHouse supports data replication using ZooKeeper and the ReplicatedMergeTree engine (of Replicated series). When creating a table, you can specify a storage engine and determine whether to replicate the table.
- Easy-of-use: You can create a ClickHouse analysis cluster in minutes on the console. No underlying infrastructure management is needed, helping you focus on analyzing data value with complete SQL statements.
- Superior performance: Queries are processed as quickly as possible by using distributed MPP architecture and all available hardware. The query efficiency is several times faster than traditional data warehouses and a single query can process up to terabytes of data per second.

- Security and reliability: Your clusters are independently deployed in isolated VPCs for more secure data access.
- Lower costs: Cost-effective devices on the cloud are used to build a cost-effective hosted ClickHouse cluster.

6.2 ClickHouse Application Scenarios

ClickHouse, an abbreviation for "Click Stream + Data WareHouse", serves as a web traffic analysis tool initially designed for OLAP analysis within data warehouses based on page click event flows. ClickHouse has a large number of applications and practices worldwide. It is widely used in various fields such as Internet advertising, apps, web, telecommunications, finance, and IoT. It suits business intelligence ideally.

Application Scenarios

- User behavior analysis

  Collect use data such as user clicks and browsing duration from the Internet (websites, APPs, and games) and import the data to ClickHouse to construct a wide table for user feature analysis. With the excellent query performance of ClickHouse, multi-dimensional and multi-mode analysis requests can be responded within subseconds. This helps quickly analyze user behaviors for precision marketing and member conversion.

- Enterprise operations analysis

  Large-scale transaction data is imported to ClickHouse to construct a large wide table with hundreds of millions of records and hundreds of dimensions. Queries are responded within subseconds. ClickHouse supports personalized statistics and uninterrupted analysis at any time, facilitating business decision-making. ClickHouse can query data several times faster than conventional data warehouses. It features high performance, cost-effectiveness, and scalability, meeting service requirements in the big data era.

- Visitor source analysis

  User behaviors are associated in user access logs through batch offline computing and a wide table of user behavior paths is generated and then synchronized to ClickHouse. A visualization system is built based on ClickHouse to display the interactive visitor source analysis results.

- BI reports

  Use ClickHouse to construct real-time interactive query reports to analyze core service metrics such as orders, revenues, and number of users in real time.

- User segmentation statistics

  Construct a user information table, select user attribute, tag data, and filter criteria in real time, and perform people feature statistics analytics based on a large number of data records.

- User profile analysis: Generate user profiles, filter users by a combination of conditions, and quickly extract target groups.

6.3 ClickHouse Enhanced Features

MPP Parallel Query

MPP parallel query: When a task is delivered to a cluster, parallel query and multi-copy parallel coordinated computing are performed on all nodes. Multi-core CPU parallel computing within a node and parallel computing between independent operators are performed as well.

Column-based Storage

Data of the same type is stored in a separate column. Only a few columns are obtained from multiple columns for analysis, which is efficient for analytical requests.

Vectorized Execution

Compared with the row-by-row processing mode in the traditional volcano model, the vectorized execution engine uses batch processing, which greatly reduces the function calling overhead, reduces the cache miss of instructions and data, and improves the CPU utilization.

Encoding and Compression

ClickHouse employs column-oriented storage. Data in the same column is continuously stored, and the underlying data is sorted during storage, which makes the regularity of data very strong and brings a higher data compression ratio. In some scenarios, the compression ratio of ClickHouse can reach 10:1, greatly reducing storage costs. In addition, the ultra-high compression ratio reduces the storage read overhead and enhances the system cache capability, thereby improving the query performance.

Multi-index Technology

ClickHouse supports partitions, primary keys, and secondary indexes. There are various types of secondary indexes. On the basis of column pruning optimization, table records are segmented horizontally. Indexes can be used to locate required block records, further improving query performance.

Materialized View

For high-frequency query modes, a small number of materialized views can be created to further improve query performance by consuming acceptable compute and storage resources.

7 Security

7.1 Asset Identification and Management

CloudTable allows you to identify and manage resources using projects.

Application Scenarios

Generally, your service system may use multiple Huawei Cloud services. You can set tags for different resource instances of these cloud services. These tags are also displayed in the billing records of each service. If your system is composed of multiple applications, setting the same tag for all resource instances used for each application helps you easily analyze resource usage and costs.

For CloudTable, tags are used to identify and classify purchased CloudTable clusters. If you add tags to a cluster, CDRs generated by the requests for this cluster will contain these tags. Using the tags, you can classify CDRs for detailed cost analysis.

7.2 Identity Authentication and Access Control

Identity Authentication

You can access CloudTable through the CloudTable console or CloudTable APIs. In either way, access requests are sent through the RESTful APIs provided by CloudTable.

CloudTable APIs can be accessed upon successful authentication. Requests sent through the CloudTable console and requests for calling APIs can both be authenticated using tokens.

Access Control

You can use Identity and Access Management (IAM) to implement fine-grained permissions management. IAM provides identity authentication, permissions management, and access control, helping you secure access to your Huawei Cloud resources.

You can grant permissions by using roles and policies.

- Roles: A coarse-grained authorization mechanism provided by IAM to define permissions based on job responsibilities. This mechanism provides a limited number of service-level roles for authorization. When using roles to grant permissions, you also need to assign the roles that the permissions depend on. Roles are not ideal for fine-grained authorization and least privilege access.
- Policies: A fine-grained authorization mechanism that defines permissions required to perform operations on specific cloud resources under certain conditions. This type of authorization is more flexible and is ideal for secure access control. For example, a specific user group is not allowed to delete a cluster. Only basic CloudTable operations (such as creating and querying jobs) are allowed.

Table 7-1 lists all the system-defined roles and policies supported by CloudTable.

Table 7-1 CloudTable system-defined role

System-Defined Role	Description	Category	Dependencies
cloudtable Administrator	Administrator permissions for CloudTable	System-defined role	The **Tenant Guest** and **Server Administrator** roles need to be assigned in the same project.

7.3 Data Protection Technologies

Data Storage Security

CloudTable's all system-defined policies encrypt your personal data (such as username, password, and mobile number) during transmission to prevent the data from being obtained by unauthorized or unauthenticated entities or individuals.

Data Destruction Mechanism

If you delete CloudTable resources, any personal data you have stored in the cluster will also be deleted.

When you delete your mobile number and email address on the console and disable message notifications, the mobile number and email address will also be deleted from the database.

Data Transmission Security

Your personal data is encrypted using TLS 1.2 or TLS 1.3 during transmission. All API call data of CloudTable can be encrypted for transmission.

7.4 Auditing and Logging

Auditing

Cloud Trace Service (CTS) records operations on the cloud resources in your account. You can use the logs generated by CTS to perform security analysis, audit compliance, track resource changes, and locate faults.

For details about the CloudTable operations that can be recorded by CTS, see **CloudTable Operations That Can Be Recorded by CTS**. After you enable CTS and create and configure a tracker, CTS starts recording operations for auditing. For details about how to enable CTS and view trackers, see **Overview**

Creating a Key Event Notification is supported by CTS. With CTS, you can monitor high-risk and sensitive operations related to IAM in real time. If you perform such an operation when using CloudTable, CTS sends a notification to subscribers in real time.

Logging

CloudTable reports all its component operation logs to Log Tank Service (LTS). LTS collects log data from cloud services. By processing massive amounts of logs efficiently, securely, and in real time, LTS provides useful insights for you to optimize the availability and performance of cloud services and applications. It also helps you efficiently perform real-time decision-making, device O&M management, and service trend analysis.

8 Permission Management

If you need to assign different permissions to employees in your enterprise to access your CloudTable resources, IAM is a good choice for fine-grained permissions management. IAM provides identity authentication, permissions management, and access control, helping you efficiently manage access to your cloud resources.

You can use your cloud account to create IAM users, and assign permissions to the users to control their access to specific resources. For example, some software developers in your enterprise need to use CloudTable resources but must not delete them or perform any high-risk operations. To achieve this result, you can create IAM users for the software developers and grant them only the permissions required for using CloudTable resources.

If your cloud account does not require individual IAM users for permissions management, you can skip this section.

IAM is free. You pay only for the resources in your account. For more information about IAM, see **IAM Service Overview**.

CloudTable Permissions

By default, new IAM users do not have any permissions assigned. You need to add a user to one or more groups, and assign permissions policies or roles to these groups. The user then inherits permissions from the groups it is a member of. This process is called authorization. After authorization, the users can perform specified operations on CloudTable based on the permissions.

CloudTable is a project-level service deployed and accessed in specific physical regions. To assign CloudTable permissions to a user group, specify the scope as region-specific projects and select projects for the permissions to take effect. If **All projects** is selected, the permissions will take effect for the user group in all region-specific projects. When accessing CloudTable, the users need to switch to a region where they have been authorized to use cloud services.

Table 8-1 describes the system-defined role supported by CloudTable. Because cloud services interact with each other, the CloudTable role is dependent on the roles of other services to implement functions. When assigning a CloudTable role to users, you need to also assign dependent roles for the CloudTable permissions to take effect.

Table 8-1 CloudTable system-defined role

System-Defined Role	Description	Category	Dependencies
cloudtable Administrator	Administrator permissions for CloudTable	System-defined role	The **Tenant Guest** and **Server Administrator** roles need to be assigned in the same project.

Table 8-2 lists the common operations supported by each system policy of CloudTable. Please choose proper system policies according to this table.

Table 8-2 Common operations supported by each system policy

Mode	Operation	cloudtable Administrator
Cluster mode	Creating a cluster	√
	Restarting a cluster	√
	Expanding cluster capacity	√
	Deleting a cluster	√
	Configuring parameters	√
	Viewing the cluster list and cluster details of CloudTable	√
	Viewing monitoring information	√
	Viewing audit logs	√

9 Notes and Constraints

This section describes the constraints on using CloudTable.

Constraints on Using HBase

- Currently, CloudTable does not have a security authentication mechanism. If HBase with an authentication mechanism is required, you are advised to use the HBase component in MRS.
- Do not enable the MOB feature for HBase. Using this feature may lead to table data read failure and JVM crash.

  For existing HBase tables, run the following command in the command line **hbase shell** to check whether the table description contains the keyword **MOB**. If it is contained, contact O&M engineers to set the table non-MOB.

  desc '*Table name*'

  For example, if the value of **IS_MOB** is **true** in the following command output, the HBase MOB feature is enabled:

  ```
  hbase:009:0> desc 't3'
  t3
  COLUMN FAMILIES DESCRIPTION
  {NAME => 'd', MOB_THRESHOLD => '102400', VERSIONS => '1', KEEP_DELETED_CELLS => 'FALSE',
  DATA_BLOCK_ENCODING => 'NONE', TTL => 'FOREVER', MIN_VERSIONS => '0', REPLICATION_SCOPE
  => '0', BLOO
  MFILTER => 'ROW', IN_MEMORY => 'false', IS_MOB => 'true', COMPRESSION => 'NONE',
  BLOCKCACHE => 'true', BLOCKSIZE => '65536'}
  ```

Constraints on Using Doris

- When creating a table in Doris, it is advisable to avoid using a single backup to ensure data reliability. Cloud services disclaim liability for any data loss or tablet damage arising from a single backup.
- Restrictions on Doris specifications: 8 vCPUs and 32 GB memory or higher are recommended for production. 4 vCPUs and 16 GB memory or 8 vCPUs and 16 GB memory can be used only for test. The cloud service does not assume any responsibility for production problems caused by small specifications.
- The Doris cluster is considered as abnormal if more than half of its FE nodes are faulty or its faulty BE nodes are not less than 3.
- The default query timeout duration is 300 seconds. If a query is not completed within 300 seconds, Doris will cancel the query. The timeout

duration can be customized by parameter configuration. You can also implement a blocking mode similar to wait timeout.

```
mysql> SHOW VARIABLES LIKE "%query_timeout%";
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| QUERY_TIMEOUT | 300   |
+---------------+-------+
1 row in set (0.00 sec)
```

Run the following commands to change the timeout duration. (The following is an example for changing the timeout duration to 1 minute.)

```
mysql> SET query_timeout = 60;
Query OK, 0 rows affected (0.00 sec)
```

The current timeout check interval is 5 seconds. Therefore, timeout duration less than 5 seconds is not recommended.

The preceding modification example is of the session level. You can run the **SET** command to implement global modification.

Constraints on Using ClickHouse

- When creating a table in ClickHouse, it is advised to employ the Replicated series engines to enhance data reliability. Again, cloud services disclaim liability for any data loss or damage arising from a single backup.
- ClickHouse does not support the JSON and Object('json') data types.
- ClickHouse's open source lab features are unstable and are not recommended for production services.

Constraints on Passwords

Table 9-1 Password constraints

Item	Description
Password	• The password must meet the following requirements: – The password must contain 8 to 16 characters (ClickHouse) or 8 to 12 characters (Doris). – The password must contain at least four types of the following characters: uppercase letters, lowercase letters, digits, and special characters ($@!%*?&._{}() +=^<>) – The password cannot be the same as the username or the username spelled backwards. • CloudTable does not save the initial password you set for logging in to a node. Please set and keep the password. To prevent malicious attacks, you are advised to set a password with a high complexity.

Constraints on Browsers

Table 9-2 Browser compatibility

Browser	Recommended Version
Google Chrome	36.0 or later
Microsoft Edge	Updated with the Windows OS

10 Related Services

Identity and Access Management (IAM)

CloudTable uses Identity and Access Management (IAM) for authentication.

Elastic Cloud Server (ECS)

CloudTable uses an Elastic Cloud Server (ECS) as a node in the cluster.

Virtual Private Cloud (VPC)

CloudTable uses Virtual Private Cloud (VPC) to provide a network topology for clusters to isolate clusters and control access.

For details, see **Creating a VPC with a Subnet**.

Object Storage Service (OBS)

CloudTable uses Object Storage Service (OBS) to store data backups and snapshots, making storage secure, reliable, and cost-effective.

Cloud Trace Service (CTS)

CloudTable uses Cloud Trace Service (CTS) to provide users with operation records of CloudTable resource operation requests and request results for querying, auditing, and backtracking.

Cloud Eye

CloudTable uses CES to monitor cluster performance metrics, delivering status information in a concise and efficient manner. CES supports alarm customization so that you can keep track of all exceptions in real time.

Log Tank Service (LTS)

CloudTable users can view collected cluster logs or dump logs on the LTS console.

11 Basic Concepts

HBase Table

An HBase table is conceptually a three-dimensional mapping. It maps a row key, a column primary key, and a timestamp to a cell value. All data within HBase is stored in these table cells.

Column

Column is a dimension of an HBase table. The column name is in the format of *<family>*:*<label>*, where *<family>* and *<label>* can be any combination of characters. An HBase table consists of a set of column families. Each column in the HBase table belongs to a column family.

Column Family

A column family is a collection of columns stored in the HBase schema. To create columns, you must create a column family first. A column family organizes data with the same property in HBase. Each row of data in the same column family is stored on the same server. Each column family can be one attribute, such as compressed packages, timestamps, and data block cache.

Timestamp

A timestamp is a 64-bit integer used to index different versions of the same data. A timestamp can be automatically assigned by HBase when data is written or assigned by users.

Index

CloudTable is a big data storage service that provides efficient key value (KV) random query. On this basis, CloudTable introduces self-developed distributed multidimensional term index feature. The storage format and computing are based on a bitmap. You can define which fields in HBase need to build a term index based on service requirements. Term data is automatically generated when you write data. In addition, the term index provides efficient multidimensional term query APIs based on the Lucene syntax. The APIs are applicable to scenarios such as user profile, recommendation system, AI, and spatiotemporal data analysis.

CloudTable supports a term index (a terminology used in Apache Lucene to represent tag index). You only need to create a CloudTable cluster to develop a client application on an ECS for a multidimensional term query.

Partition

Partitions divide a table's data into distinct logical segments based on defined criteria. Logically, a single table is split into multiple partitions, which simplifies data management.

Bucketing

Data is divided into different buckets based on the hash values of bucketing columns.

FE (Frontend)

Frontend nodes process user access requests, plan query parsing, and manage metadata and nodes.

BE (Backend)

Backend nodes are responsible for both storing data and executing query plans. Data is divided into shards and replicated across multiple backend nodes for redundancy and availability.

Replicas

To ensure data security and maintain high service availability during exceptional circumstances, ClickHouse offers a replica configuration that replicates data from a single server to two or more redundant servers.

ClickHouse Shard

In ultra-large-scale massive data processing scenarios, the storage and compute resources of a single server can become a significant bottleneck. To enhance service efficiency, the cloud database ClickHouse employs a distributed architecture where massive datasets are stored across multiple servers. Each server is responsible for storing and processing a subset of the overall data. Within this architecture, each such server is referred to as a shard.

