

Alibaba Cloud

Hologres Product Introduction

Document Version: 20220615

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.
2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.
3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.
4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).
5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.
6. Please directly contact Alibaba Cloud for any errors of this document.

Document conventions

Style	Description	Example
 Danger	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	 Danger: Resetting will result in the loss of user configuration data.
 Warning	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	 Warning: Restarting will cause business interruption. About 10 minutes are required to restart an instance.
 Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	 Notice: If the weight is set to 0, the server no longer receives new requests.
 Note	A note indicates supplemental instructions, best practices, tips, and other content.	 Note: You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click Settings> Network> Set network type .
Bold	Bold formatting is used for buttons, menus, page names, and other UI elements.	Click OK .
Courier font	Courier font is used for commands	Run the <code>cd /d C:/window</code> command to enter the Windows system folder.
<i>Italic</i>	Italic formatting is used for parameters and variables.	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[] or [a b]	This format is used for an optional value, where only one item can be selected.	<code>ipconfig [-all -t]</code>
{ } or {a b}	This format is used for a required value, where only one item can be selected.	<code>switch {active stand}</code>

Table of Contents

- 1.What is Hologres? ----- 05
- 2.Benefits ----- 08
- 3.Scenarios ----- 09
- 4.Core technical capabilities ----- 11
 - 4.1. Architecture ----- 11
 - 4.2. QE ----- 13
- 5.Terms ----- 16

1. What is Hologres?

This topic describes the overview and features of Hologres.

Hologres is an all-in-one real-time data warehouse service developed by Alibaba Cloud that allows you to write, update, and analyze large amounts of data in real time. Hologres supports standard SQL syntax. It is compatible with PostgreSQL and supports most PostgreSQL functions. Hologres supports online analytical processing (OLAP) and ad hoc analysis for petabytes of data, and provides high-concurrency, low-latency online data services. Seamlessly integrated with MaxCompute, Flink, and DataWorks, Hologres empowers enterprises with full-stack online and offline data warehouse solutions.

Hologres is designed to provide a real-time compute engine that features high performance, high reliability, cost efficiency, and high scalability. Hologres provides real-time data warehouse solutions that help manage large amounts of data and interactive query services that can respond in sub-seconds. Hologres is widely used in scenarios such as construction of real-time data mid-ends, fine-grained analysis, self-service analysis, marketing profile setup audience grouping, and real-time risk control.

Functions and features

- Queries and analysis in multiple scenarios

Hologres supports multiple index types and storage models such as row-oriented storage and column-oriented storage. It also supports diversified queries and analytics, such as simple queries, complex queries, and ad hoc queries. By using a massively parallel processing (MPP) architecture, Hologres processes SQL statements in distributed mode. This improves resource utilization and accelerates the analysis of large amounts of data.

- Interactive analysis in sub-seconds

Hologres performs parallel computing based on a scalable MPP architecture, and uses vectorization operators to maximize the computing power of CPUs. It provides column-oriented storage in the ORC format to optimize indexes and SSD storage to improve I/O throughput. This way, Hologres supports interactive analysis in sub-seconds for petabytes of data.

- High-performance point queries by using primary keys

Owing to primary key indexes in row-oriented tables and optimized shortest path queries, Hologres supports hundreds of thousands of high-performance point queries per second and data updates with high throughput. Compared with open source systems, Hologres improves the performance by more than 10 times. This makes Hologres suitable for scenarios such as ID mapping and dimension table associations for real-time data processing.

- Federated queries and accelerated queries by using foreign tables

Seamlessly integrated with MaxCompute, Hologres allows you to use foreign tables to accelerate queries on MaxCompute data. Compared with direct queries on MaxCompute data, the accelerated queries can be 5 to 10 times faster. Hologres supports the association analysis of hot and cold data. To simplify data import into data lakes or warehouses, Hologres allows millions of rows to be synchronized per second from MaxCompute tables to Hologres tables. In addition, Hologres allows you to read data from and write data to Object Storage Service (OSS) external tables.

- Native real-time data warehouse

To tackle frequent data updates, simple data models, and quick data analysis in real-time data warehouses, Hologres supports real-time high-concurrency data writes and updates, as well as isolation and atomicity among transactions. This ensures that data can be queried the moment after it is written.

- Real-time high-throughput data writes and updates

Hologres is integrated with computing frameworks such as Flink and Spark. Therefore, Hologres allows you to use built-in connectors to write and update large amounts of data in real time. You can use various tables such as source tables, result tables, and dimension tables, and perform complex operations, such as merging multiple data streams.

- A development environment in which what you see is what you get

Hologres allows you to query data the moment after it is written. You can query data in a specific table, in all tables in a schema, or in a database. Hologres also allows you to create, delete, or update a view for one or more tables. In addition to data update and delete operations, you can join tables, perform nested queries, and use window functions to query data in Hologres. Hologres provides native support for the analysis of semi-structured JSON data and allows you to synchronize databases from sources such as MySQL to Hologres with a few clicks in real time.

- Event-driven from end to end

Hologres allows you to parse the binary logs of table update events. You can use Flink to consume Hologres binary logs in order to realize end-to-end real-time development across warehouse layers. This way, you can reduce the end-to-end latency of data processing while meeting the requirements for tiered data governance.

- Enterprise-level O&M capabilities

Hologres supports fine-grained management in computing loads and access permissions. It provides diversified monitoring and alerting metrics, and supports elastic scaling of computing resources as well as hot system updates. These secure and reliable solutions can meet enterprise-level O&M requirements.

- Data security

Hologres provides fine-grained access control policies and data security features, including Bring Your Own Key (BYOK) encryption, data masking, Data Security Guard, and IP address whitelists. It also supports multiple authentication systems such as Resource Access Management (RAM), Security Token Service (STS), and independent account systems. Hologres has passed Payment Card Industry Data Security Standard (PCI DSS) assessment.

- Load isolation

Hologres supports the isolation of loads based on resource groups. This allows you to isolate resources for different business requirements, various query types, and data reads and writes. This way, you can make sure the sustainability and stability of the system.

- High reliability

Hologres allows you to use multiple compute instances to build a high-reliability mode. In this mode, Hologres supports storage sharing among compute instances, fault isolation, high availability of online services, and rapid automatic recovery of failed nodes. Hologres allows you to store data in a highly reliable triplicate redundant storage in Apsara Distributed File System. Therefore, you do not need to use local disks.

- Ecosystem and scalability

Hologres is compatible with the PostgreSQL ecosystem and seamlessly integrated with DataWorks. DataWorks is the big data computing engine and big data development platform of Alibaba Cloud. You can get started with Hologres without additional learning.

- Compatibility with PostgreSQL

Compatible with PostgreSQL, Hologres provides a Java Database Connectivity (JDBC) or Open Database Connectivity (ODBC) interface to connect to third-party extract, transform, load (ETL) and business intelligence (BI) tools, such as Quick BI, DataV, Tableau, and FanRuan. It also supports spatial data analysis based on geographic information systems (GIS).

- DataWorks development and integration

Hologres is seamlessly integrated with DataWorks. Together with DataWorks, Hologres provides visualized, intelligent, and all-in-one data warehouse construction and interactive analysis tools. This way, Hologres provides enterprise-level solutions to data asset management, data lineage management, real-time data synchronization, and data services.

- Vector search engine: Proxima

Hologres is also integrated with Alibaba Cloud Machine Learning Platform for AI (PAI) and has a built-in vector search engine named Proxima. Proxima supports online real-time feature storage, real-time retrievals, and vector searches.

2. Benefits

Compatible with PostgreSQL, Hologres enables you to query and analyze MaxCompute data and query the data written in real time. It also allows you to perform federated analytics on real-time and MaxCompute data. Hologres helps you construct a real-time data warehouse for your enterprise.

To meet your requirements for a real-time data warehouse, Hologres improves the efficiency to develop data warehouses and simplifies operations by innovating in the following aspects:

- Real-time data processing

Hologres supports real-time data writes and updates. You can view the data the moment after it is written, without the need for batch processing. Hologres is integrated with Flink and Spark to help you develop high-throughput, low-latency, modeled, and high-quality real-time data warehouses. This way, it meets your real-time requirements for business insights.

- Interactive analysis in sub-seconds

Hologres supports interactive analysis in sub-seconds for large amounts of data, without the need for pre-computing. It also supports diversified types of data analyses, such as multidimensional analysis, ad hoc analysis, query analysis, and accelerated queries on MaxCompute data, to meet your requirements for data analysis based on what you see is what you get. By using vectorization and intelligent indexing optimization technologies, Hologres surpasses open source systems in performance and concurrency.

- All-in-one data services

Hologres supports diversified scenarios such as multidimensional analysis, high-performance point queries, and data retrieval. It also supports load isolation, simplified data architectures, unified data access, and security policies.

- Open ecosystem

Compatible with PostgreSQL 11, Hologres supports standard SQL syntax and SQL development frameworks. In Hologres, you can connect to more than 19 mainstream business intelligence (BI) tools without the need to rewrite the tools, which improves efficiency. Hologres also supports data lakes and semi-structured data, such as JSON data. You can import data from Object Storage Service (OSS) or Data Lake Formation (DLF) to data lakes or warehouses with ease.

- Accelerated queries on MaxCompute data

Hologres allows you to accelerate queries on MaxCompute data. It is seamlessly integrated with MaxCompute, so you can directly query MaxCompute data in Hologres, without the need to export and import data. In addition, you can connect MaxCompute to BI tools and perform interactive analysis.

- Separated computing and storage resources

- Hologres uses a cloud-native architecture in which computing and storage resources are separated. This allows you to scale in or out storage resources and computing resources independently.
- This architecture supports dynamic configuration changes and auto scaling. You can scale in or out Hologres resources based on your business needs. The more resources you deploy support the greater number of concurrent queries.

3.Scenarios

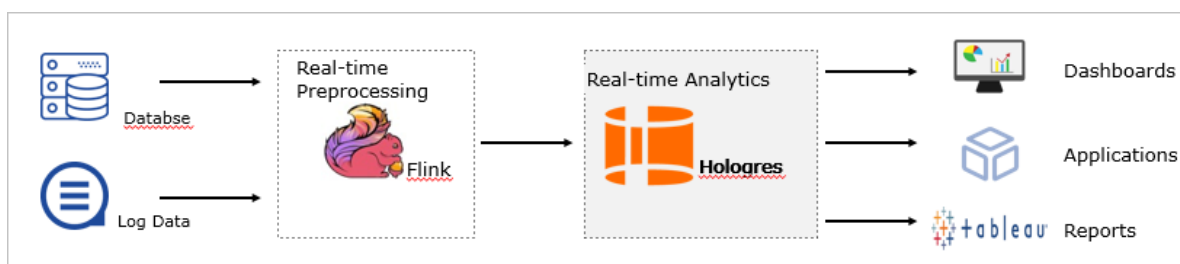
Hologres is a next-generation provided by Alibaba Cloud. It is compatible with PostgreSQL and integrates seamlessly with the big data ecosystem. Hologres supports of real-time and offline data. You can connect Hologres to third-party BI tools to analyze data in a visualized manner. This topic describes typical scenarios where the core features of Hologres are used.

Hologres is typically used in the following scenarios:

- Construct a real-time data warehouse.

Business data is written to Realtime Compute in real time for extract, transform, load (ETL) operations. Then, you can use Hologres to query data in real time and export data to a third-party analytics tool for real-time analytics. For example, you can use Hologres to meet the following requirements:

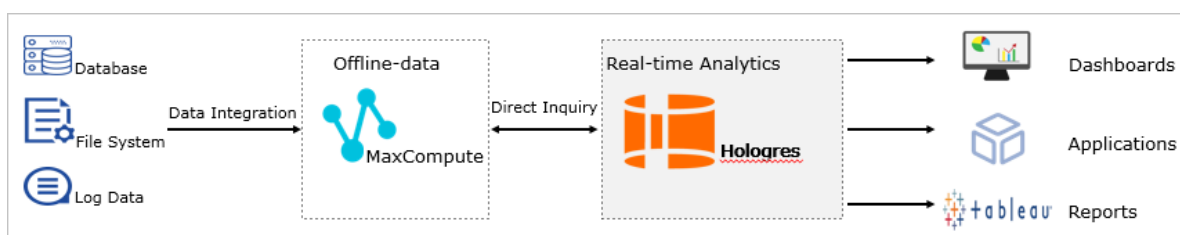
- The data department wants to construct a , view real-time data on a dashboard, and create real-time reports.
- The O&M and data application departments require real-time monitoring, real-time exception detection and alerting, and real-time debugging.
- The business department requires real-time risk control, real-time recommendation, real-time effect analytics, and real-time training.



- Perform .

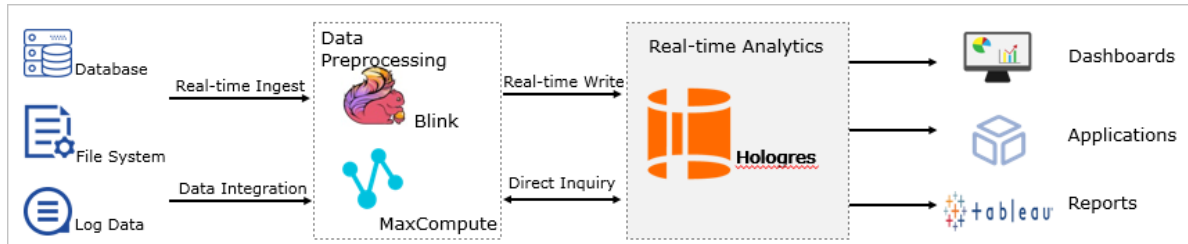
After business data is written to the offline data warehouse MaxCompute, you can directly query data by using Hologres or import data to Hologres for queries. You can also connect Hologres to BI tools to analyze offline data in real time. For example, you can use Hologres to meet the following requirements:

- Query offline MaxCompute data in real time.
- Analyze offline MaxCompute data and generate reports.
- Apply offline MaxCompute data for online use. For example, expose your table data over a RESTful API.



- Perform federated analytics on offline and real-time data.

Business data is divided into hot data and cold data based on their access frequency. Cold data is stored in the offline data warehouse MaxCompute, whereas hot data is stored in Hologres. You can use Hologres to perform federated analytics on offline and real-time data. You can also connect Hologres to BI tools to respond to business requirements of simple and complex queries.



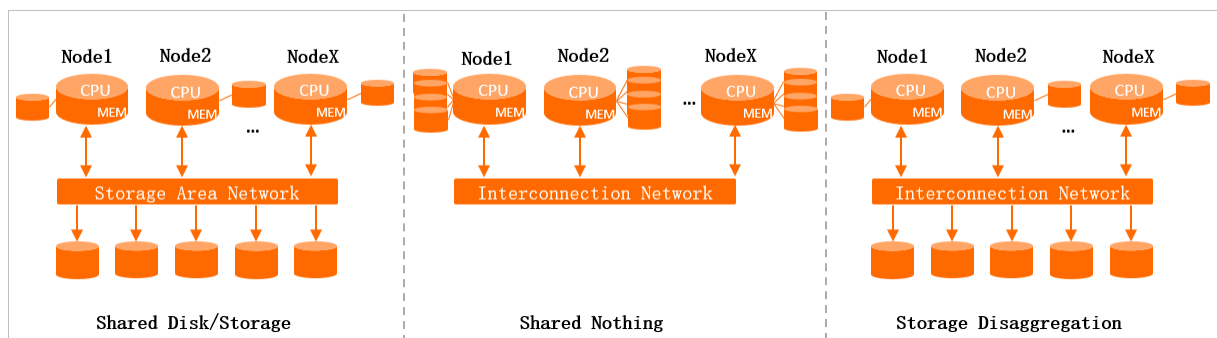
4. Core technical capabilities

4.1. Architecture

This topic describes the architecture of Hologres and the role of each component.

Hologres architecture

The following figure shows the three storage-computing architectures that are commonly used in traditional distributed systems.



- Shared Disk or Shared Storage

A distributed storage cluster is deployed in this architecture. Each compute node accesses data in this shared storage cluster as if the data is stored in a standalone database. The storage cluster can be easily scaled. Compute nodes must adopt a distributed coordination mechanism to implement data synchronization and ensure data consistency. The maximum number of compute nodes is limited.

- Shared Nothing

Each compute node mounts its own storage. A compute node can process data only from one shard. Compute nodes can communicate with each other. A data summary node is required to summarize the data. This architecture can be easily extended, but also has its drawbacks. When a compute node goes online upon a failover, the compute node must wait for data to be loaded before the compute node can provide services. In addition, storage and computing resources must be scaled out at the same time, which lacks flexibility. After a scale-out, it takes an extended period to rebalance data.

- Storage Disaggregation

Like the Shared Storage architecture, a distributed storage cluster is deployed and shared in this architecture. The mode in which compute nodes process data is similar to that in the Shared Nothing architecture. Data is sharded. Each compute node processes data only from one shard. Each computing node can have a local cache.

The Storage Disaggregation architecture has the following benefits:

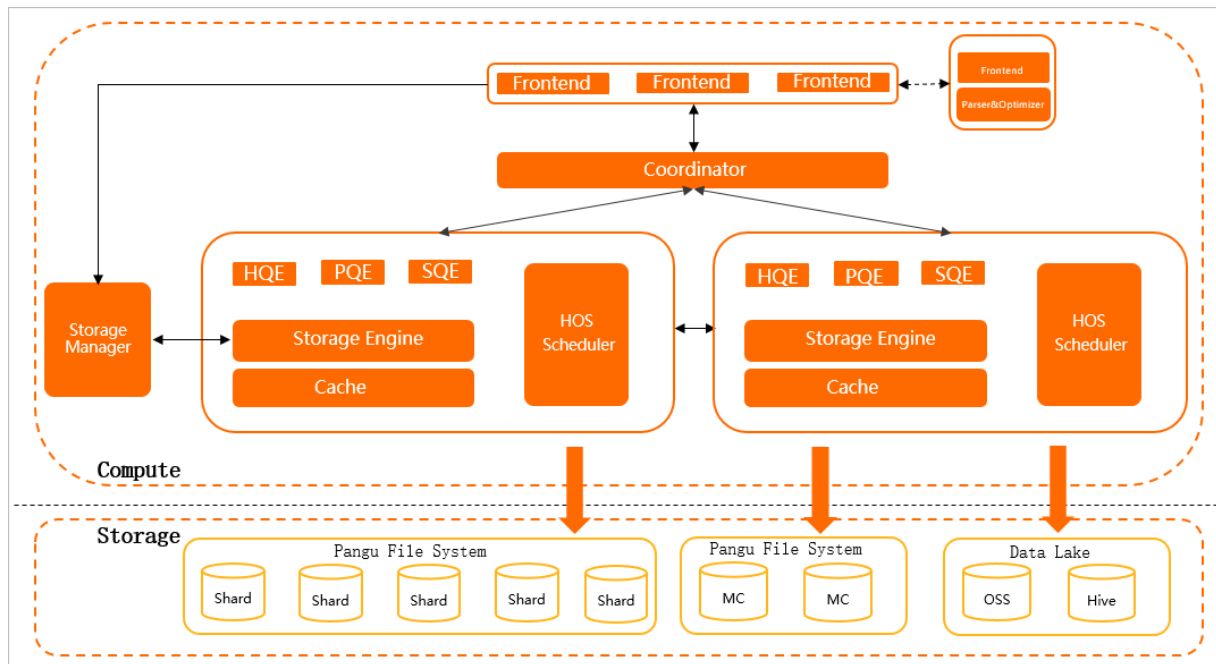
- Effortless data consistency: The system needs only to ensure that only one compute node is writing data to a shard at the same time.
- Flexible scaling: Computing and storage resources can be separately scaled out as needed. This improves the flexibility in scenarios such as big promotions. If computing resources are insufficient, you can immediately scale out only computing resources. This eliminates time-consuming and labor-intensive data rebalancing that is required in the Shared Nothing architecture. Unlike the Shared Nothing architecture, the storage capacity is unlimited for each compute node.

- Fast recovery from computing node failures: After a failover, data can be asynchronously pulled from the distributed and shared storage as needed. This accelerates recovery from failovers.

Hologres uses the Storage Disaggregation architecture. Hologres uses Alibaba-developed Pangu as the storage system, which is a distributed file system similar to Hadoop Distributed File System (HDFS). You can scale the capacity of Hologres based on the business requirements to easily cope with traffic peaks in your online systems.

Components in Hologres architecture

The following figure shows the architecture of Hologres.



The architecture consists of the following components:

- Computing layer
 - Frontend (FE)

An FE authenticates, parses, and optimizes SQL statements. A Hologres instance has multiple FEs. Hologres is ecologically compatible with PostgreSQL 11. You can use the standard PostgreSQL syntax for development or use PostgreSQL-compatible development tools and Business Intelligence (BI) tools to connect to Hologres.

- Coordinator

The coordinator distributes FE-parsed data to different compute nodes to implement fast diversion and node switching.

- HoloWorker

A HoloWorker consists of components such as Query Engine (QE), Storage Engine (SE), and HOS Scheduler. HoloWorkers schedule and compute user tasks.

- QE

- Hologres Query Engine (HQE)

Developed by Alibaba Cloud, HQE uses a scalable Massively Parallel Processing (MPP) architecture to implement full parallel computing. HQE uses vectorization operators to make maximum use of CPUs and achieve ultimate query performance. HQE is the main module of Hologres QE.

- PostgreSQL Query Engine (PQE)

PQE provides compatibility with PostgreSQL. PQE supports a variety of PostgreSQL extensions, such as PostGIS and user-defined functions (UDFs) that are written in PL/Java, PL/SQL, and PL/Python. The functions and operators that are not supported by HQE can be executed by using PQE. HQE has been continuously optimized in each version. The final goal is to integrate all features of PQE.

- Seahawks Query Engine (SQE)

SQE allows Hologres to seamlessly connect to MaxCompute. This provides high-performance access to all types of MaxCompute files, without the need to migrate or import data. This also allows Hologres to access complex tables such as hash tables and range-clustered tables, and implement interactive analysis of PB-level batch data.

- SE

SE manages and processes data. SE allows you to perform create, read, update, and delete (CRUD) operations on data.

- Cache

The Cache component caches query results to improve query performance.

- HOS Scheduler

HOS Scheduler provides lightweight scheduling capabilities.

- Storage Manager (SM)

SM manages meta information and coordinates the work of FEs and SE. SM detects compute node failures and failovers, and uses HOS Scheduler to switch services between compute nodes. SM is the key component that ensures the high availability of Hologres.

- Storage layer

- Hologres data is stored in the Pangu file system.
- Hologres can access MaxCompute data that is stored in Pangu. Pangu implements efficient mutual access between Hologres and MaxCompute.
- Hologres can store data to Object Storage Service (OSS). This accelerates analysis in data lakes and reduces storage costs.

4.2. QE

This topic describes the benefits of Query Engine (QE) used in Hologres. This topic also describes how a query is executed in Hologres.

Benefits

Developed by Alibaba Cloud, Hologres QE supports latest big data technologies and provides high performance in processing all types of queries. Hologres Query Engine (HQE) is the main module of Hologres QE. Hologres QE provides the following benefits:

- Distributed execution

Hologres QE is a distributed execution model that works with the Storage Disaggregation architecture. An execution plan is represented by a directed acyclic graph (DAG) that consists of asynchronous operators. Execution plans can express a variety of complex queries and fit the data storage model of Hologres. This way, Query Optimizer (QO) can easily optimize the queries based on a variety of optimization technologies.

- Fully asynchronous execution

Hologres QE provides an end-to-end fully asynchronous framework that can eliminate the bottleneck of a high concurrency system and make full use of resources. This minimizes the impact of the read latency caused by the Storage Disaggregation architecture.

- Vectorization and column-oriented processing

Hologres QE processes data in a vectorized manner in operators where possible. Hologres QE is deeply integrated with Storage Engine (SE). Flexible execution models are built to leverage a variety of indexes. Vectorization and materialization are deferred as much as possible to prevent unnecessary data reads or computing.

- Adaptive incremental processing

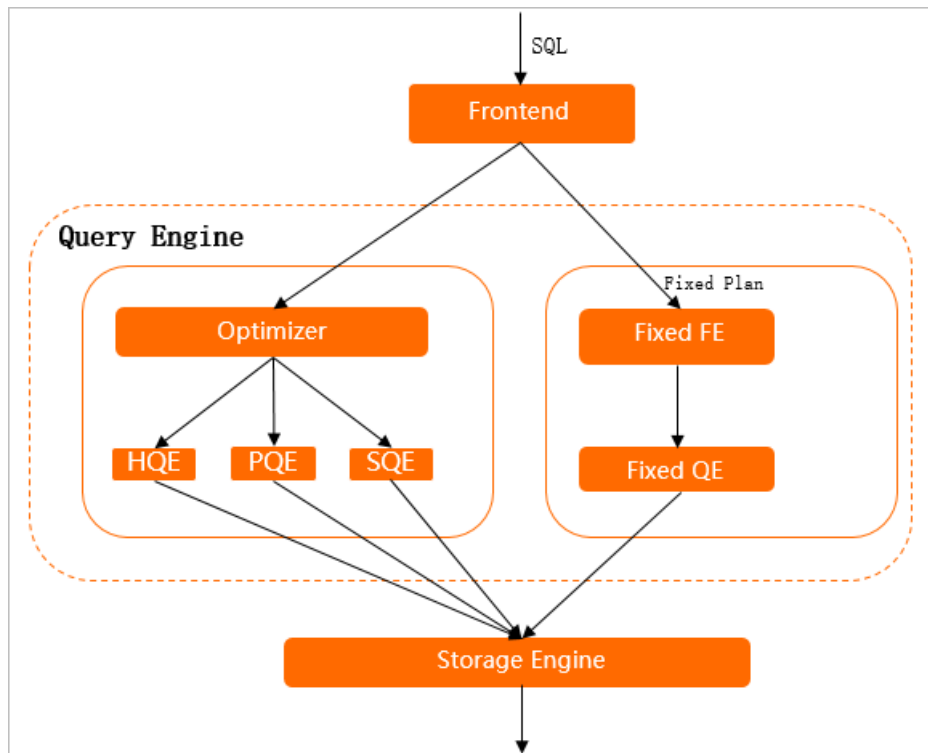
Hologres QE implements adaptive incremental processing for regular real-time data.

- Optimization of specific queries

Hologres QE provides unique optimization for specific queries.

Query execution process

Hologres QE consists of multiple workers. The following figure shows how a worker executes a query that is initiated by a client.



After a client initiates an SQL query, the query is executed in the following process:

1. A frontend (FE) authenticates and parses the SQL query and distributes the SQL query to different modules of Hologres QE.
2. Hologres QE selects an execution path based on the characteristics of the SQL query.
 - If the SQL query is a point query, the FE distributes the SQL query to Fixed QE to obtain data by skipping QO. This shortens the execution path and improves query performance. This execution path is also called a fixed plan. Point queries, key-value queries in ApsaraDB for HBase, and point writes are processed by using fixed plans.
 - If the SQL query is an online analytical processing (OLAP) query, the FE distributes the SQL query to QO. QO parses the SQL query and generates an execution plan. The execution plan includes information such as the estimated costs of operators, statistics, and range that is narrowed down. QO determines to use HQE, PostgreSQL Query Engine (PQE), Seahawks Query Engine (SQE), or Hive Query Engine (Hive QE) to compute the operators based on the execution plan.
3. After Hologres QE determines the correct execution plan, Hologres QE uses Storage Engine (SE) to obtain data, merges the data from different shards, and then returns the query result to the client.

5. Terms

This topic describes the terms related to Hologres, including account, role, instance, database, schema, internal table, foreign table, partitioned table, table group, and shard count.

Account

- Alibaba Cloud account

An Alibaba Cloud account is used to create and manage Hologres instances. You can use an Alibaba Cloud account to grant permissions to RAM users.

The following table describes the formats of Alibaba Cloud accounts in Hologres.

Format	Description	Example
ALIYUN\$<AccountName>@aliyun.com	<AccountName> indicates the logon name of the Alibaba Cloud account.	ALIYUN\$company@aliyun.com
<AccountName>@aliyun.com		company@aliyun.com
<Account ID>	The ID of the Alibaba Cloud account.	1344445678xxx

- RAM user

RAM users can create and manage objects, such as databases and tables, in an instance based on their permissions granted by Alibaba Cloud accounts.

The following table describes the formats of RAM users in Hologres.

Format	Description	Example
p4_<Account ID>	<ul style="list-style-type: none"> In this format, p is in lowercase. <Account ID> indicates the ID of the Alibaba Cloud account. 	p4_12345678xxx
<subUserName>@<Account Name>.onaliyun.com	<ul style="list-style-type: none"> <subUserName> indicates the logon name of the RAM user. <AccountName> indicates the logon name of the Alibaba Cloud account. <Account ID> indicates the ID of the Alibaba Cloud account. 	holouser@company.onaliyun.com
<subUserName>@<Account Name>		
<subUserName>@<Account ID>.onaliyun.com		
RAM\$<subUserName>		
RAM\$<Account Name>: <subUserName>		
RAM\$<Account ID>: <subUserName>		

Format	Description	Example
<subUserName>@<Account ID>		

Role

- Superuser

By default, the system sets the Alibaba Cloud account that is used to purchase an instance as a superuser of the instance. The superuser has all permissions on the instance. The superuser can create databases, delete databases, create roles, and grant permissions to roles.

- Regular user

A regular user can log on to a Hologres instance and perform allowed data analytics operations only after the regular user obtains the required permissions from a superuser. A regular user can also be assigned the superuser role.

Instance

In Hologres, an instance is an entity in which you can use and manage databases. All database operations are performed in the instance.

Database

A database is a data repository that organizes, stores, and manages data based on data structures. In a database, you can create, update, and delete tables. You can also execute functions.

When you purchase a Hologres instance, a default database named **postgres** is created. This database is allocated a few resources and is used only for management purposes. You can create a database based on your business needs. For more information about how to create a database, see [CREATE DATABASE](#).

Schema

A schema is a collection of database objects, similar to a directory in a file system. Objects such as tables and functions in an instance are stored in schemas.

When you create a database, a default schema named **public** is created. By default, operations on objects in Hologres instances are performed in this schema. You can also create a schema based on your business needs. For more information, see [CREATE SCHEMA](#).

Note

- A schema cannot contain nested subschemas.
- Objects in a schema must have unique names.
- Objects in different schemas can have the same name.

Table

A table is a unit that stores data in columns and rows. The number and order of columns are fixed, and each column has a name. The number of rows is variable, which reflects the data amount stored at a given moment.

Each column has a data type. A data type constrains the set of possible values that can be assigned to a column. For more information, see [Data types](#).

Hologres supports the following two types of tables:

- Internal table

Internal tables store data in Hologres. The data types of internal tables must be supported by Hologres.

- Foreign table

Foreign tables map external data sources and do not store data in Hologres. Foreign tables are read-only. Therefore, you cannot perform DML operations or create indexes for foreign tables.

You can create foreign tables in Hologres to query data directly from external data sources such as MaxCompute.

For more information about how to create a foreign table in Hologres, see [CREATE FOREIGN TABLE](#).

Partitioned table

Partitioned tables are parent tables that are divided into multiple segments based on partition keys. These segments are called partitions. Different types of data are stored in different partitions. In Hologres, only one partition key can be used for list partitioning. When you query data, you can specify filters based on partition key values in query statements to exclude irrelevant partitions. This improves the query speed.

For more information about how to create and delete a partitioned table in Hologres, see [CREATE TABLE](#) and [CREATE/DROP PARTITION TABLE](#).

Engine

Hologres is a distributed real-time data warehouse. Hologres has many internal components. To use Hologres, you do not need to understand all the components. You need only to understand the components to be used in your business.

- Frontend (FE)

An FE authenticates, parses, and optimizes SQL statements. A Hologres instance has multiple FEs. Hologres is ecologically compatible with PostgreSQL 11. You can use the standard PostgreSQL syntax for development or use PostgreSQL-compatible development tools and Business Intelligence (BI) tools to connect to Hologres.

- Query Engine (QE)

- Hologres Query Engine (HQE)

Developed by Alibaba Cloud, HQE uses a scalable Massively Parallel Processing (MPP) architecture to implement full parallel computing. HQE uses vectorization operators to make maximum use of CPUs and achieve ultimate query performance. HQE is the main module of Hologres QE.

- PostgreSQL Query Engine (PQE)

PQE provides compatibility with PostgreSQL. PQE supports a variety of PostgreSQL extensions, such as PostGIS and user-defined functions (UDFs) that are written in PL/Java, PL/SQL, and PL/Python. The functions and operators that are not supported by HQE can be executed by using PQE. HQE has been continuously optimized in each version. The final goal is to integrate all features of PQE.

- Seahawks Query Engine (SQE)

SQE allows Hologres to seamlessly connect to MaxCompute. This provides high-performance access to all types of MaxCompute files, without the need to migrate or import data. This also allows Hologres to access complex tables such as hash tables and range-clustered tables, and implement interactive analysis of PB-level batch data.

- Storage Engine (SE)

SE manages and processes data. SE allows you to perform create, read, update, and delete (CRUD) operations on data.