

ALIBABA CLOUD

阿里云

交互式分析Hologres

产品简介

文档版本：20220602

 阿里云

法律声明

阿里云提醒您阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
 危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 危险 重置操作将丢失用户配置数据。
 警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告 重启操作将导致业务中断，恢复业务时间约十分钟。
 注意	用于警示信息、补充说明等，是用户必须了解的内容。	 注意 权重设置为0，该服务器不会再接受新请求。
 说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置>网络>设置网络类型。
粗体	表示按键、菜单、页面名称等UI元素。	在结果确认页面，单击确定。
Courier字体	命令或代码。	执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

目录

1.什么是实时数仓Hologres	05
2.产品优势	07
3.Hologres的应用场景	08
4.核心技术能力	09
4.1. 产品架构	09
4.2. 执行引擎	11
5.基本概念	14

1.什么是实时数仓Hologres

本文为您介绍什么是实时数仓Hologres以及Hologres的功能。

Hologres是阿里巴巴自主研发的一站式实时数仓引擎（Real-Time Data Warehouse），支持海量数据实时写入、实时更新、实时分析，支持标准SQL（兼容PostgreSQL协议，支持大部分PostgreSQL函数），支持PB级数据多维分析（OLAP）与即席分析（Ad Hoc），支持高并发低延迟的在线数据服务（Serving），与MaxCompute、Flink、DataWorks深度融合，提供企业级离在线一体化全栈数仓解决方案。

Hologres致力于高性能、高可靠、低成本、可扩展的实时计算引擎研发，为用户提供海量数据的实时数据仓库解决方案和亚秒级交互式查询服务，广泛应用在实时数据中台建设、精细化分析、自助式分析、营销画像、人群圈选、实时风控等场景。

功能特性

● 多场景查询分析

Hologres支持行存、列存等存储模式和多种索引类型，同时满足简单查询、复杂查询、即席查询等多样化的分析查询需求。Hologres使用大规模并行处理架构，分布式处理SQL，提高资源利用率，实现海量数据极速分析。

○ 亚秒级交互式分析

Hologres采用可扩展的大规模并行处理（MPP）架构全并行计算、向量化算子发挥CPU最佳算力、ORC格式列存优化索引、SSD存储优化IO吞吐，支持PB级数据亚秒级交互式分析体验。

○ 高性能主键点查

基于行存表的主键索引和查询引擎的短路径优化，Hologres支持每秒数十万QPS高性能服务型点查、支持高吞吐更新，相比开源系统性能提升10倍以上，可用于实时加工链路的维表关联、ID-Mapping等场景。

○ 联邦查询，外表加速

Hologres无缝对接MaxCompute，支持外部表透明加速查询，相比原生MaxCompute访问加速5-10倍，支持冷热数据关联分析，同时支持MaxCompute与Hologres之间百万行每秒高速同步，支持OSS外部表读写，简化数据入湖入仓。

● 原生实时数仓

针对实时数仓数据更新频繁、数据模型简单和分析场景敏捷的特性，Hologres支持高并发实时写入与更新，支持事务隔离与原子性，数据写入即可查。

○ 高吞吐实时写入与更新

Hologres与Flink、Spark等计算框架原生集成，通过内置Connector，支持高通量数据实时写入与更新，支持源表、结果表、维度表多种场景，支持多流合并等复杂操作。

○ 所见即所得的开发

数据实时写入即可查询，支持DB、Schema、Table三级体系，支持视图View，原生支持Update/Delete，支持关联、嵌套、窗口等丰富表达能力，原生支持半结构化JSON数据分析，支持MySQL等数据库数据整库一键入库，实时同步。

○ 全链路事件驱动

支持表更新事件的Binlog透出能力，通过Flink消费Hologres Binlog，实现数仓层次间全链路实时开发，满足分层治理的前提下，缩短数据加工端到端延迟。

● 企业级运维能力

支持计算负载、访问权限等细粒度管控要求，提供丰富的监控和告警指标，支持计算资源弹性扩展，支持系统热升级，满足企业级安全可靠的运维需求。

- 数据安全

支持细粒度访问控制策略，支持BYOK数据存储加密和数据脱敏，支持数据保护伞、IP白名单，支持RAM、STS及独立账号等多种认证体系，通过PCI-DSS安全认证。

- 负载隔离

支持基于资源组的负载隔离，隔离不同业务需求，不同查询类型，写入和读取等资源竞争场景，保障系统的持续稳定。

- 高可靠设计

多个计算实例组成高可靠部署模式，实例间共享存储，支持故障隔离和在线服务高可用，支持故障节点快速自动恢复。无需本地盘，盘古三副本高可靠冗余存储。

- 生态与可扩展性

兼容PostgreSQL生态，与大数据计算引擎及大数据智能研发平台DataWorks无缝打通。无需额外学习，即可上手开发。

- 兼容PostgreSQL生态

Hologres兼容PostgreSQL生态，提供JDBC/ODBC接口，轻松对接第三方ETL和BI工具，包括Quick BI、DataV、Tableau、帆软等。支持GIS空间数据分析。

- DataWorks开发集成

Hologres与DataWorks深度集成，提供图形化、智能化、一站式的数仓搭建和交互式分析服务工具，支持数据资产、数据血缘、数据实时同步、数据服务等企业级能力。

- 达摩院Proxima向量检索

Hologres与机器学习平台PAI紧密结合，内置达摩院Proxima向量检索插件，支持在线实时特征存储、实时召回、向量检索。

2. 产品优势

Hologres兼容PostgreSQL生态，支持快速查询分析MaxCompute的数据、实时查询实时写入的数据、联邦分析实时数据与离线数据，帮助您快速搭建企业实时数据仓库。

Hologres专注实时数仓需求，从以下几个方面创新，提高数仓开发效率，降低应用门槛。

- 专注实时场景

数据实时写入、实时更新，无需批处理，写入即可见。Hologres与Flink、Spark原生集成，支持高吞吐、低延时、有模型、高质量的实时数仓开发，满足业务洞察实时性需求。

- 亚秒级交互式分析

Hologres支持海量数据亚秒级交互式分析，无需预计算，支持多维分析、即席分析、探索式分析、MaxCompute加速分析，满足所见即所得式数据分析。Hologres采用向量化计算和智能索引优化技术，性能和并发能力大幅领先开源系统。

- 统一数据服务出口

一个引擎支持多维分析、高性能点查、数据检索等多样化的场景，同时支持负载隔离，简化数据架构，统一数据访问接口与安全策略。

- 开放生态

标准SQL接口，兼容PostgreSQL 11协议，无缝对接主流BI和SQL开发框架，支持19+款主流BI工具，无需应用重写，无额外学习成本。支持数据湖场景，支持JSON等半结构化数据，OSS、DLF简易入湖入仓。

- MaxCompute查询加速

快速查询分析MaxCompute的数据，Hologres与MaxCompute无缝连接，您无需导入导出即可查询MaxCompute表的数据，快速获取查询结果。支持MaxCompute对接BI工具，支持交互式分析场景。

- 计算存储分离架构

- 采用计算与存储分离的云原生架构，存储资源和计算资源分离部署并独立扩展。
- 支持动态升降配，您可以根据业务需求灵活地扩容或缩容Hologres资源。资源越多，查询的并发数量越大。

3.Hologres的应用场景

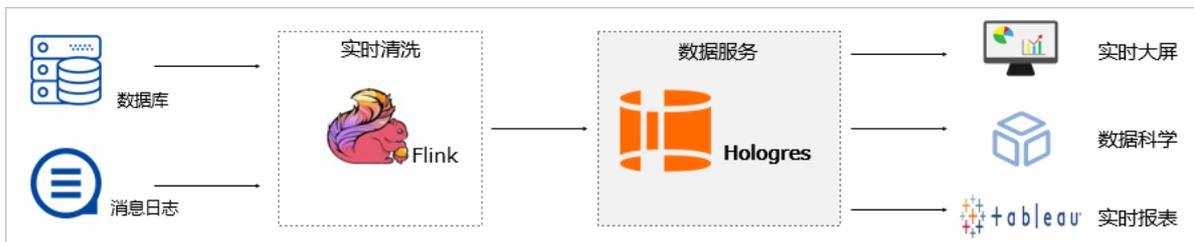
Hologres兼容PostgreSQL生态，是新一代的产品，与大数据生态无缝连接，支持实时与离线数据，对接第三方BI工具，实现可视化分析业务。本文为您介绍基于Hologres核心功能的典型应用场景。

Hologres的典型应用场景如下：

- 搭建实时数仓。

实时写入业务数据至实时计算后，使用ETL（Extract Transformation Load）方式清洗、转换及整理数据。您可以通过Hologres实时查询并输出数据至第三方分析工具进行实时分析。典型应用场景如下：

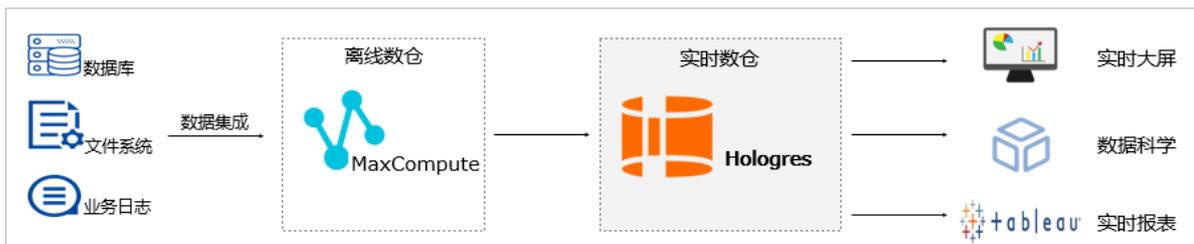
- 数据部门搭建、展示实时大屏和分析实时Reporting报表。
- 运维和数据应用部门执行实时监控、实时异常检测预警与实时调试。
- 业务部门进行实时风控、实时推荐、实时效果分析和实时训练。



- 查询。

写入业务数据至离线数据仓库MaxCompute，通过Hologres直接加速查询或导入数据至Hologres查询，并对接BI分析工具，实现实时分析离线数据。典型应用场景如下：

- 实时查询MaxCompute离线数据。
- 分析MaxCompute离线数据报表。
- 输出MaxCompute离线数据的在线应用，例如RESTful API的使用。



- 联邦分析实时数据和离线数据。

业务数据分为冷数据和热数据，冷数据存储在离线数据仓库MaxCompute中，热数据存储在Hologres中。Hologres可以联邦分析实时数据和离线数据，对接BI分析工具，快速响应简单查询与复杂查询的业务需求。



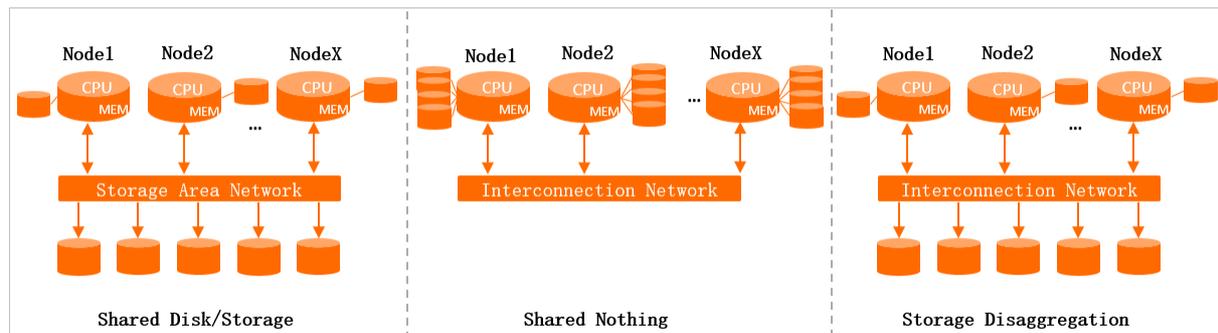
4.核心技术能力

4.1. 产品架构

本文为您介绍Hologres的产品架构以及每个组件的作用。

架构优势

在传统的分布式系统中，常用的存储计算架构有如下三种。



- Shared Disk/Storage（共享存储）

有一个分布式的存储集群，每个计算节点像访问单机数据一样访问这个共享存储上的数据。这种架构的存储层可以比较方便的扩展，但是计算节点需要引入分布式协调机制保证数据同步和一致性，因此计算节点的可扩展性有一个上限。

- Shared Nothing

每个计算节点自己挂载存储，一个节点只能处理一个分片的数据，节点之间可以通信，最终有一个汇总节点对数据进行汇总。这种架构能比较方便的扩展，但是它的缺点是节点Failover需要等待数据加载完成之后才能提供服务；并且存储和计算需要同时扩容，不够灵活，扩容后，有漫长的数据Rebalance过程。

- Storage Disaggregation（存储计算分离架构）

存储和Shared Storage类似，有一个分布式的共享存储集群，计算层处理数据的模式和Shared Nothing类似，数据是分片的，每个shard只处理自己所在分片的数据，每个计算节点还可以有本地缓存。

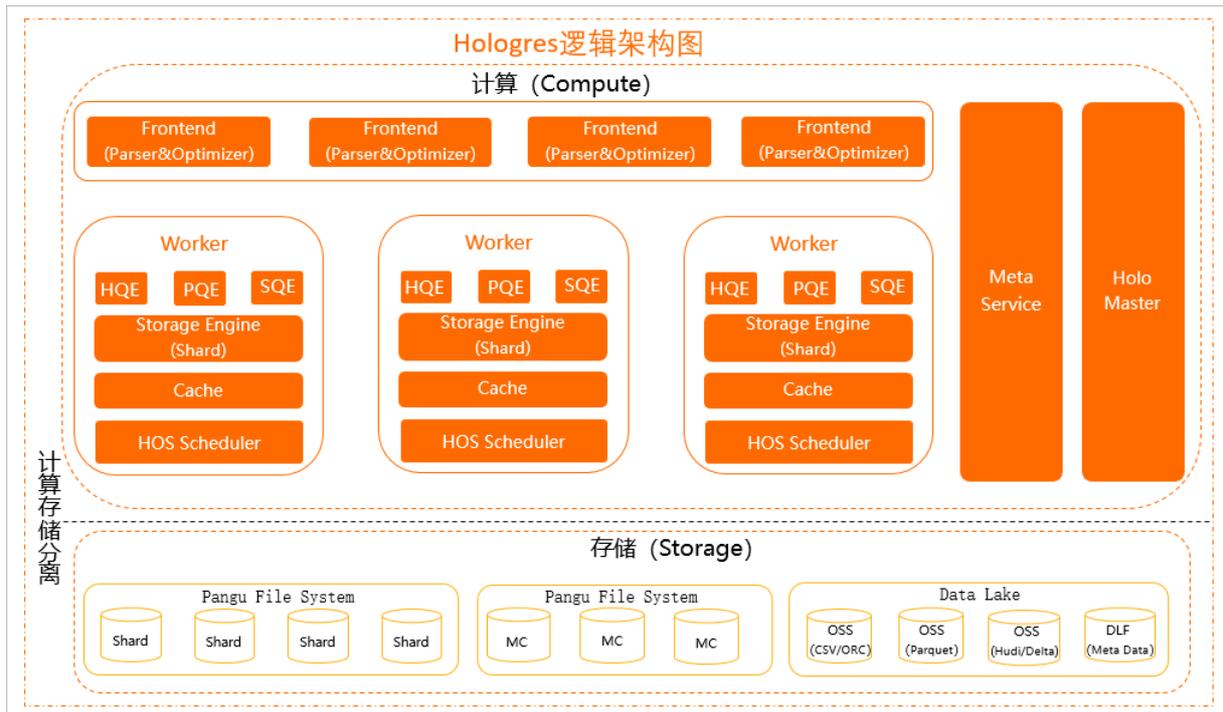
存储计算分离的架构存在以下优势。

- 一致性问题处理简单：计算层只需要保证同一时刻有一个计算节点写入同一分片的数据。
- 扩展更灵活：计算和存储可以分开扩展，计算不够扩计算节点，存储不够扩存储节点。这样在大促等场景上会非常灵活。计算资源不够了，马上扩容计算就好了，不需要像Shared Nothing那样做耗时耗力的数据rebalance；也不会出现Shared Nothing那样，出现单机的存储容量瓶颈。
- 计算节点故障恢复快：计算节点发生Failover之后，数据可以按需从分布式的共享存储异步拉取。因此Failover的速度非常快。

Hologres采用的是第三种存储计算分离架构，Hologres的存储使用的是阿里自研的Pangu分布式文件系统（类似HDFS）。用户可以根据业务需求进行弹性扩缩容，轻松应对在线系统不同的流量峰值。

架构组件介绍

Hologres架构图如下所示。



整个架构从上往下分为如下组件。

- 计算层

- 接入节点 (Frontend, FE)

Hologres接入节点，主要用于SQL的认证、解析、优化，一个实例有多个FE接入节点。在生态上兼容Postgres 11，因此用户可以使用Postgres标准语法进行开发，也可以用Postgres兼容的开发工具和BI工具直接连接Hologres。

- 计算HoloWorker

HoloWorker分为执行引擎、存储引擎、调度等组件，主要负责用户任务的计算、调度。

- 其中执行引擎（Query Engine，QE）主要有三个，执行引擎的技术原理请参见[Hologres执行引擎技术揭秘](#)。

- HQE（Hologres Query Engine）

- Hologres自研存储引擎，采用可扩展的MPP架构全并行计算，向量化算子发挥CPU极致算力，从而实现极致的查询性能。（QE主要由HQE组成）。

- PQE（Postgres Query Engine）

- 用于兼容Postgres提供扩展能力，支持PG生态的各种扩展组件，如PostGIS，UDF（PL/JAVA，PL/SQL，PL/Python）等。部分HQE还没有支持的函数和算子，会通过PQE执行，每个版本都在持续优化中，最终目标是去掉PQE。

- SQE（Seahaws Query Engine）

- 无缝对接MaxCompute（ODPS）的执行引擎，实现对MaxCompute的本地访问，无需迁移和导入数据，就可以高性能和全兼容的访问各种MaxCompute文件格式，以及Hash/Range clustered table等复杂表，实现对PB级离线数据的交互式分析，技术原理请参见[Hologres加速查询MaxCompute技术揭秘](#)。

- 存储引擎Storage Engine（SE）

主要用于管理和处理数据，包括创建、查询、更新和删除（简称CRUD）数据等，关于存储引擎详细的技术原理请参见[Hologres存储引擎技术揭秘](#)。

- Cache（缓存）

主要是结果缓存，提高查询性能。

- HOS Scheduler

轻量级调度。

- Meta Service

主要用于管理元数据Meta信息（包括表结构信息以及数据在Storage Engine节点上的分布情况），并将Meta信息提供给FE节点。

- Holo Master

Hologres原生部署在K8s上，当某个Worker出现故障时，由K8s进行快速拉起创建一个新的Worker，保障Worker级别的可用性。在Worker内部，每个组件的可用性则由Holo Master负责，当组件出现状态不正常时，Holo Master则会快速重新拉起组件，从而恢复服务。，高可用技术原理请参见[Hologres高可用技术揭秘](#)

- 存储层

- 数据直接存储在Pangu File System。
- 与MaxCompute在存储层打通，能直接访问MaxCompute存储在盘古的数据，实现高效相互访问。
- 支持直接访问OSS、DLF数据，类型包含CSV、ORC、Parquet、Hudi、Delta，加速数据湖探索，也可以将数据回流至OSS，降低存储成本。

4.2. 执行引擎

本文为您介绍Hologres的执行引擎以及内部各组件。

执行引擎优势

Hologres的执行引擎（主要以HQE为主）是自研的执行引擎，通过与大数据领域最新技术结合，实现了对各种查询类型的高性能处理，主要具有如下优势。

- 分布式执行

执行引擎是一个和存储计算分离架构配合的分布式执行模型。执行计划由异步算子组成的执行图DAG（有向无环图）表示，可以表达各种复杂查询，并且完美适配Hologres的数据存储模型，方便对接查询优化器，利用各种查询优化技术。

- 全异步执行

端到端的全异步处理框架，可以避免高并发系统的瓶颈，充分利用资源，并且最大可能地避免存储计算分离系统带来的读数据延迟的影响。

- 向量化和列处理

算子内部处理数据时最大可能地使用向量化执行，与存储引擎深度集成，通过灵活的执行模型，充分利用各种索引，最大化地延迟向量物化和延迟计算，避免不必要的读数据和计算。

- 自适应增量处理

对常见实时数据应用查询模式进行自适应增量处理。

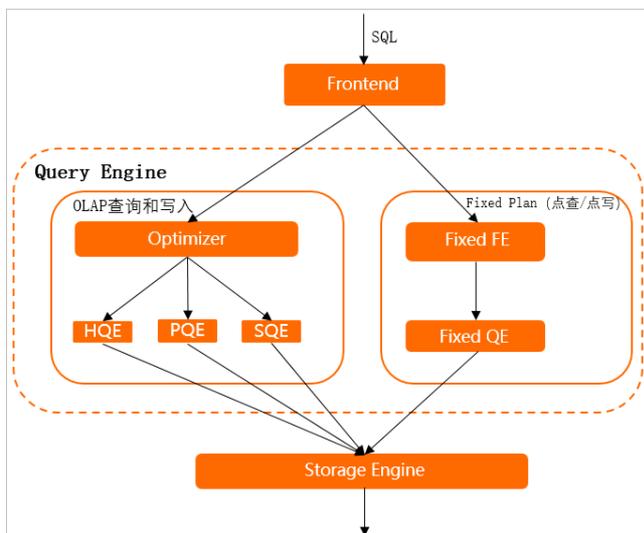
- 特定查询深度优化

对一些特定查询模式的独特优化。

更多技术原理请参见[Hologres执行引擎揭秘](#)。

Query执行过程

当客户端下发一个Query后，在执行引擎中实际上会有多个worker节点，以其中的一个worker节点为例，执行过程如下图所示。



当客户端发起一个SQL后，执行过程如下。

1. Frontend (FE) 节点对SQL进行解析和认证，并分发至执行引擎（Query Engine）的不同执行模块。
2. 执行引擎（Query Engine）会根据SQL的特征走不同的执行路径。
 - 如果是点查/点写的场景，会跳过优化器（Query Optimizer, QO），直接分发至后端获取数据，减少数据传送链路，从而实现更优的性能。整个执行链路也叫Fixed Plan，点查（与HBase的KV查询）、点写场景会直接走Fixed Plan。

-
- 如果是OLAP查询和写入场景：首先会由优化器（Query Optimizer, QO）对SQL进行解析，生成执行计划，在执行计划中会预估出算子执行Cost、统计信息、空间裁剪等。QO会通过生成的执行计划，决定使用HQE、PQE、SQE或者Hive QE对算子进行真正的计算。
3. 执行引擎决定正确的执行计划，然后通过存储引擎（Storage Engine, SE）进行数据获取，最后对每个Shard上的数据进行合并，返回至客户端。

5.基本概念

本文为您介绍Hologres中，账号、用户角色、实例、数据库、Schema、内部表、外部表、分区表、Table Group及Shard Count的基本概念。

账号

- 阿里云账号

阿里云账号即主账号，用于创建和管理Hologres实例，可以授权给其他用户管理员权限。

主账号支持的表达格式如下表所示。

账号格式	描述	示例
ALIYUN\$<AccountName>@aliyun.com	<AccountName>为主账号的登录名称。	ALIYUN\$company@aliyun.com
<AccountName>@aliyun.com		company@aliyun.com
<Account ID>	主账号的ID。	1344445678xxx

- RAM用户

RAM用户即子账号。阿里云账号授予RAM用户一定权限后，RAM用户可以在权限范围内创建和管理实例中的数据库和表。

RAM用户支持的表达格式如下表所示。

账号格式	描述	示例
p4_<Account ID>	<ul style="list-style-type: none"> ◦ p使用小写字母。 ◦ <Account ID>为主账号的ID。 	p4_12345678xxx
<subUserName>@<Account Name>.onaliyun.com	<ul style="list-style-type: none"> ◦ <subUserName>为子账号的登录名称。 ◦ <Account Name>为主账号的登录名称。 ◦ <Account ID>为主账号的ID。 	holouser@company.onaliyun.com
<subUserName>@<Account Name>		
<subUserName>@<Account ID>.onaliyun.com		
RAM\$<subUserName>		
RAM\$<Account Name>:<subUserName>		
RAM\$<Account ID>:<subUserName>		
<subUserName>@<Account ID>		

用户角色

- 超级管理员

系统默认设置购买实例的主账号为超级管理员Superuser。Superuser拥有整个实例的所有权限，例如创建数据库、删除数据库、创建角色以及为角色授权。

- 普通用户

普通用户需要被Superuser授权后才能访问实例，并在权限范围内进行数据开发。普通用户也可以被授权为Superuser。

实例

实例（Instance）是您使用和管理数据库存储服务的实体。您对数据库的所有操作都需要在实例中完成。

数据库

数据库是按照数据结构来组织、存储和管理数据的仓库。您可以在数据库中创建、更新或删除表，以及执行与函数相关的操作。

购买Hologres实例后，系统自动创建postgres数据库。该数据库分配到的资源较少，仅用于管理，开发实际业务需要新建数据库。新建数据库请参见[CREATE DATABASE](#)。

Schema

Schema是数据库对象的集合，相当于文件系统中的目录。实例中的表和函数等对象存放于各个Schema中。

成功创建数据库后，系统默认创建命名为public的Schema，您对实例对象的所有操作默认都在该Schema中执行。您也可以新建Schema，详情请参见[CREATE SCHEMA](#)。

说明

- Schema不能嵌套Schema。
- 同一个Schema中的对象名称不能重复。
- 不同Schema中的对象名称可以重复。

表

表是由行和列组成的数据存储单元。列的数量和顺序是固定的，每一列拥有一个名称。行的数量是变化的，表示在给定的时刻，表中存储的数据量。

表的每一列都有一个数据类型，表示该列的取值类型，详情请参见[数据类型汇总](#)。

Hologres中的表包括以下两种类型：

- 内部表

指数据存储在Hologres中的表。表中的数据类型可以是Hologres支持的任意一种类型。

- 外部表

指在Hologres中不存储数据，只进行字段映射的表。外部表的数据都是只读的，您在外部表中不能执行DML语句或创建索引。

您可以在Hologres中创建外部表，加速查询外部数据源的数据，例如查询MaxCompute的数据。

Hologres中创建外部表请参见[CREATE FOREIGN TABLE](#)。

分区表

分区表是根据键值属性划分父表为许多小的子集，这些子集称为分区。不同类型的数据存放于不同分区，Hologres当前仅支持对单个键值进行列表分区。查询数据时，您可以通过分区键值过滤查询语句，排除无关的分区，从而提升查询速度。

Hologres中创建或删除分区表请参见[CREATE PARTITION TABLE](#)、[DROP PARTITION TABLE](#)。

引擎组件

Hologres是一个分布式实时数仓，在内部会有很多组件，实际业务中，无需所有组件都有所了解，只需要在实际业务中了解需要使用的组件，助您更好的理解和使用Hologres。

- 接入节点（Front end, FE）

Hologres接入节点，主要用于SQL的认证、解析、优化，一个实例有多个FE接入节点。在生态上兼容Postgres 11，因此您可以使用Postgres标准语法进行开发，也可以用Postgres兼容的开发工具和BI工具直接连接Hologres。

- 执行引擎（Query Engine, QE）主要有4个，执行引擎的技术原理请参见[Hologres执行引擎技术揭秘](#)。

- HQE（Hologres Query Engine）

Hologres自研存储引擎，采用可扩展的MPP架构全并行计算，向量化算子发挥CPU极致算力，从而实现机制的查询性能。（QE主要由HQE组成）。

- PQE（Postgres Query Engine）

用于兼容Postgres提供扩展能力，支持PG生态的各种扩展组件，如PostGIS, UDF（PL/JAVA, PL/SQL, PL/Python）等。部分HQE还没有支持的函数和算子，会通过PQE执行，每个版本都在持续优化中，终极目标是去掉PQE。

- SQE（Seahaws Query Engine）

无缝对接MaxCompute(ODPS)的执行引擎，实现对MaxCompute的本地访问，无需迁移和导入数据，就可以高性能和全兼容的访问各种MaxCompute文件格式，以及Hash/Range clustered table等复杂表，实现对PB级离线数据的交互式分析，技术原理请参见[Hologres加速查询MaxCompute技术揭秘](#)。

- 存储引擎Storage Engine（SE）

主要用于管理和处理数据，包括创建、查询、更新和删除（简称CRUD）数据等，关于存储引擎详细的技术原理请参见[Hologres存储引擎技术揭秘](#)。