

数据仓库服务  
8.2.0

技术白皮书

文档版本 09  
发布日期 2024-08-08



版权所有 © 华为云计算技术有限公司 2024。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## 商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

# 目录

<b>1 DWS 产品简介</b>	<b>1</b>
1.1 产品定位	1
1.2 应用场景	1
1.3 产品架构和优势	5
<b>2 DWS 支持平台和技术指标</b>	<b>9</b>
2.1 技术指标	9
<b>3 DWS 核心技术</b>	<b>11</b>
3.1 Share-nothing 架构	11
3.2 分布式环境的数据布局	12
3.2.1 数据分布式存储	12
3.2.2 数据分区	13
3.2.3 数据并行导入	14
3.3 全并行的数据查询处理	16
3.4 向量化执行和行列混合引擎	17
3.5 资源监控及管控	18
3.5.1 负载管理	19
3.5.2 空间管控	19
3.6 分布式事务	20
3.7 在线扩容	22
3.8 SQL on Anywhere	24
3.8.1 SQL on Hadoop	24
3.8.2 基于 GDS 的集群互联互通	26
3.9 集群管理与高可用	27
3.10 SQL 自诊断	29
3.11 透明加密	30
3.12 数据脱敏	31
3.13 备份与容灾	33
3.13.1 备份恢复	33
3.13.2 集群容灾	34
<b>4 DWS 工具</b>	<b>36</b>
4.1 客户端工具	36
4.1.1 gsql	36

---

4.1.2 Data Studio.....	36
4.1.3 Database Schema Convertor.....	38
4.1.4 数据管理服务(Data Admin Service, 简称 DAS).....	40
4.2 数据库监控工具.....	43
4.3 三方工具兼容.....	53
<b>5 DWS 对外接口.....</b>	<b>54</b>

# 1 DWS 产品简介

## 1.1 产品定位

数据仓库服务（ GaussDB(DWS)，以下简称DWS）是一种基于公有云基础架构和平台的在线数据处理数据库，提供即开即用、可扩展且完全托管的分析型数据库服务。DWS是基于华为融合数据仓库GaussDB产品的云原生服务，兼容标准ANSI SQL 99和SQL 2003，同时兼容PostgreSQL/Oracle数据库生态，为各行业PB级海量大数据分析提供有竞争力的解决方案。

DWS提供标准数仓、IoT数仓和实时数仓三种产品形态，围绕企业级内核、实时分析、协同计算、融合分析、云原生五大方向构筑业界第一数据仓库。

标准数仓面向数据分析场景，为用户提供高性能、高扩展、高可靠、高安全、易运维的企业级数仓服务，支持2048节点、20PB级超大规模数据分析能力。支持冷热数据分析，存储、计算弹性伸缩，并按需、按量计价，为用户提供弹性灵活、极致性价比的体验。适用于“库、仓、市、湖”一体化的融合分析业务。

IoT数仓在标准数仓基础上，通过融合流、时序引擎，为用户提供实时数据接入和高并发实时分析能力。适用于物联网、IoT等实时分析场景。

实时数仓在大规模数据查询和分析能力基础上，提供高并发、高性能、低时延的事务处理能力。适用于HTAP混合负载场景，“一库两用，生产即分析”。支持集群和单机两种部署模式。

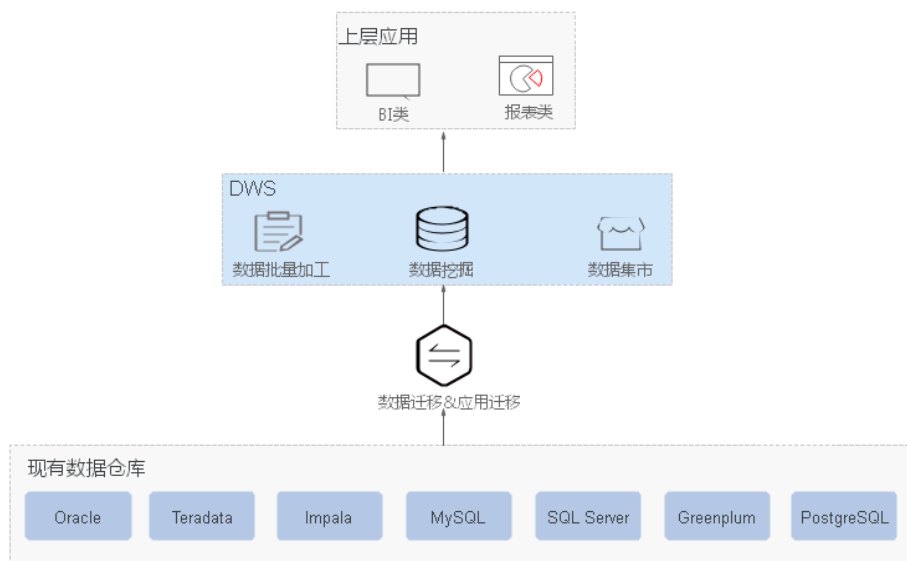
DWS可广泛应用于金融、政企、IoT、互联网、电商、能源、电信等各个行业，已连续三年入选Gartner发布的数据管理解决方案魔力象限，相比传统数据仓库，性价比提升数倍，具备大规模扩展能力和企业级可靠性。

## 1.2 应用场景

### 数据仓库迁移

数据仓库是企业的重要数据分析系统，随着业务量的增长，自建数仓性能逐渐不能满足实际需求，同时扩展性差、成本高，导致扩容极为困难。DWS作为云上企业级数据仓库，具备高性能、低成本、易扩展等特性，满足大数据时代企业数据仓库业务诉求。

图 1-1 数据仓库迁移



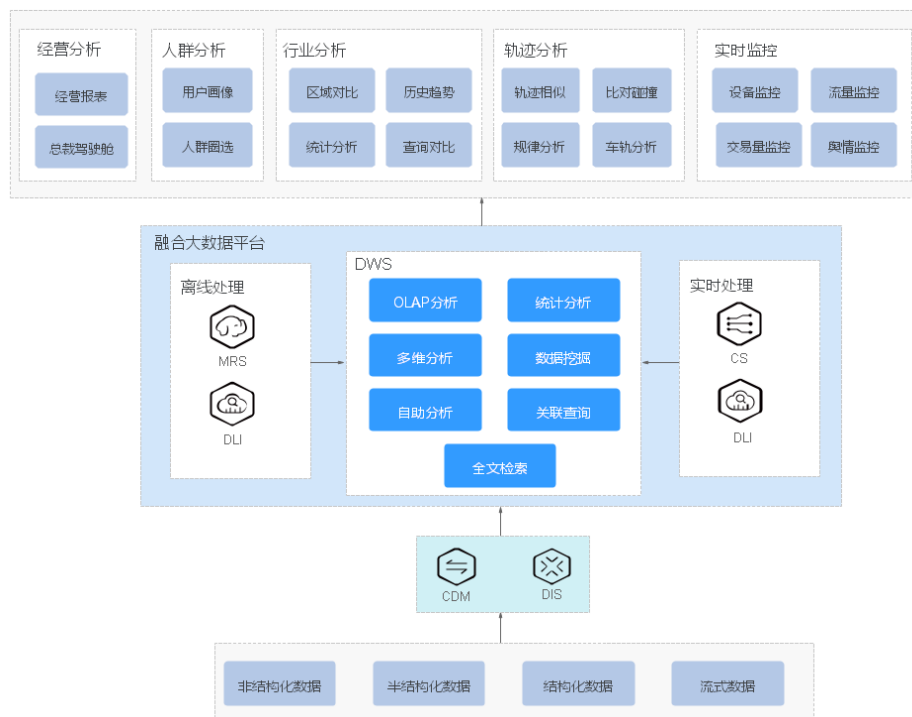
### 优势

- 平滑迁移  
DWS提供配套的迁移工具，可支持Teradata、Oracle、MySQL、SQL Server、PostgreSQL、Greenplum、Impala等常用数据分析系统的平滑迁移。
- 兼容传统数据仓库  
DWS支持SQL 2003标准，兼容Oracle的部分语法和数据结构，支持存储过程，可与常用BI（business intelligence）工具无缝对接，业务迁移修改量极小。
- 安全可靠  
DWS支持数据加密，同时可与数据库安全服务对接，保证云上数据安全。同时DWS支持数据自动全量、增量备份，提升数据可靠性。

### 大数据融合分析

随着IT、信息技术的发展和进步，数据资源已经成为企业的核心资源。整合数据资源，构建大数据平台，发现数据价值，成为企业经营的新趋势和迫切诉求。而如何从海量数据中快速挖掘“价值”，成为助力客户实现预测性分析的关键要素。

图 1-2 大数据融合分析



### 优势

- 统一分析入口  
以DWS的SQL作为上层应用的统一入口，应用开发人员使用熟悉的SQL语言即可访问所有数据。
- 实时交互分析  
针对即时的分析需求，分析人员可实时从大数据平台中获取信息。
- 弹性伸缩  
增加节点，即可扩展系统的数据存储能力和查询分析的性能，可支持PB级数据的存储和计算。

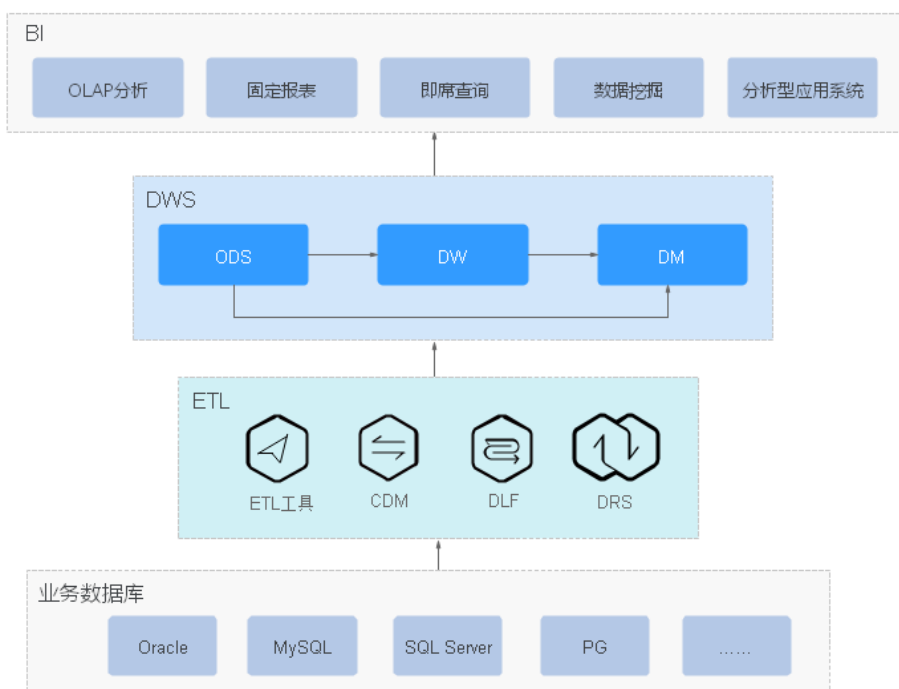
## 增强型 ETL 和实时 BI 分析

数据仓库在整个BI系统中起到了支柱的作用，更是海量数据收集、存储、分析的核心。为IoT（Internet of things）、金融、教育、移动互联网、O2O（Online to Offline）等行业提供强大的商业决策分析支持。

### 优势

- 数据迁移  
多数据源，高效批量、实时数据导入。
- 高性能  
PB级数据低成本的存储与万亿级数据关联分析秒级响应。
- 实时  
业务数据流实时整合，及时对经营决策进行优化与调整。

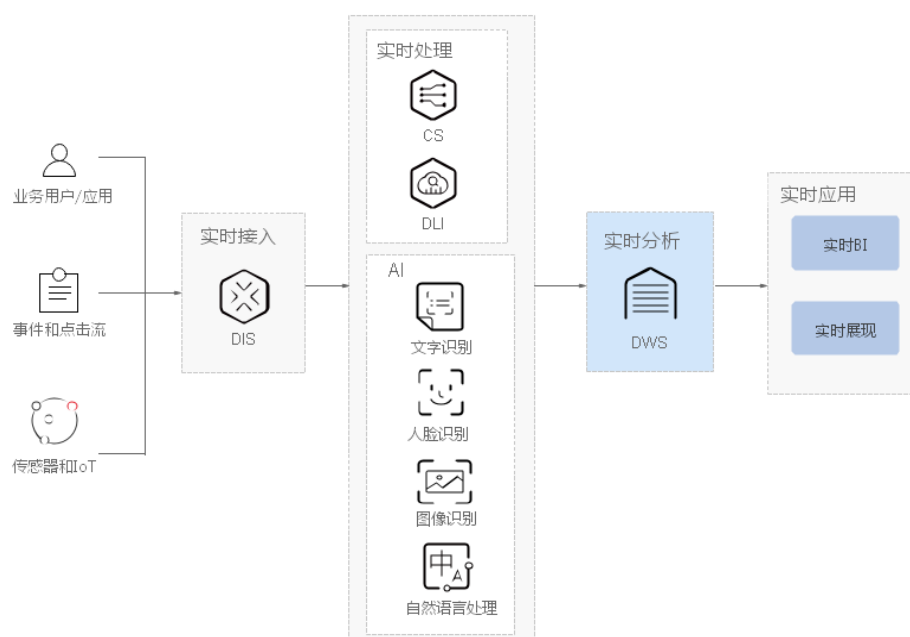
图 1-3 增强型 ETL+实时 BI 分析



## 实时数据分析

移动互联网、IoT场景下会产生大量实时数据，为了快速获取数据价值，需要对数据进行实时分析，DWS通过+时序、+流、+AI引擎，实现快速入库和查询能力，可支持实时数据分析。

图 1-4 实时数据分析

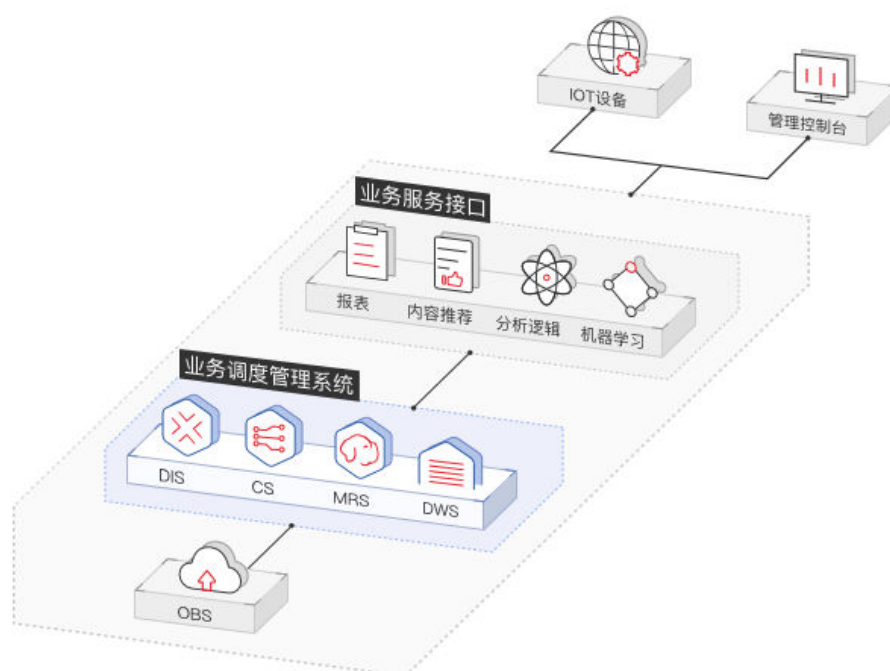


## 优势



- 流式数据实时入库  
IoT、互联网等数据经过流计算及AI服务处理后，可实时写入DWS。
- 实时监控与预测  
围绕数据进行分析和预测，对设备进行监控，对行为进行预测，实现控制和优化。
- AI融合分析  
AI服务对图像、文本等数据的分析结果可在DWS中与其他业务数据进行关联分析，实现融合数据分析。
- IoT场景

图 1-5 IoT 场景



物联网（IoT）所产生的数据，通过构建GaussDB(DWS)，围绕海量的数据进行实时分析并进行反馈优化。应用在工业IoT、O2O业务系统、车联网等解决方案。

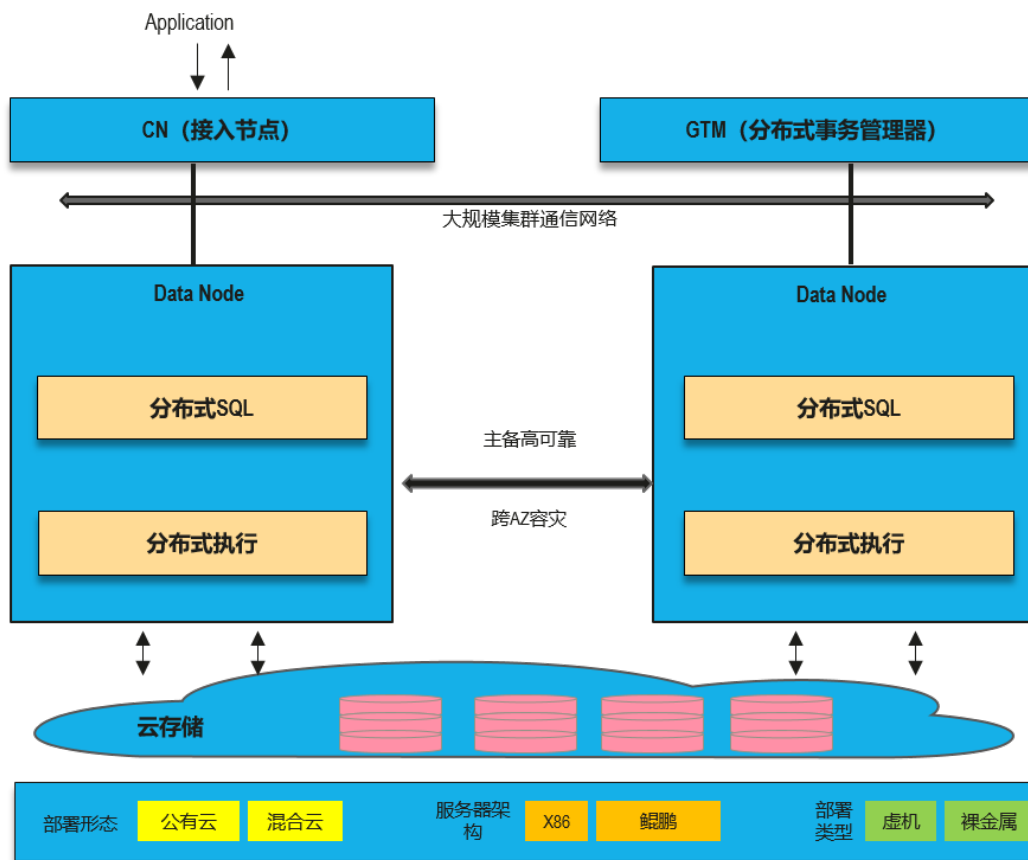
优势如下：

- 流式数据实时入库：IoT设备及网关汇集的流式数据经华为云DIS导入至GaussDB(DWS)。
- 设备监控与预测：围绕数据，进行分析和预测，对设备进行监视、控制和优化补给，以及自我诊断和修复。
- 信息推荐：结合围绕在用户周围的联网设备汇集的数据，为用户进行定向的信息推荐。

## 1.3 产品架构和优势

DWS是基于Shared-nothing架构的分布式、并行数据库集群，其产品架构请参见下图。

图 1-6 产品架构图



**该架构保证DWS具有以下关键优势：**

DWS服务可以在多种云形态下使用，支持华为公有云形态、混合云本地运维HCS形态、混合云长期连线HCSO形态，以及边缘小站IES形态，为不同类型用户对部署形态和运维模式的差异化提供选择。

DWS数据库内核统一使用华为自主研发的GaussDB数据库，兼容PostgreSQL的数据库内核引擎，从单机OLTP数据库改造为企业级MPP（大规模并行处理）架构的OLAP分布式数据库，其主要面向海量数据分析场景。

DWS产品提供标准数仓、IoT数仓和实时数仓三种产品形态，与传统数据仓库相比，主要有以下特点与显著优势，可解决多行业超大规模数据处理与通用平台管理问题：

**易使用**

- 一站式可视化便捷管理  
DWS让您能够轻松完成从项目概念到生产部署的整个过程。通过使用DWS管理控制台，您不需要安装数据仓库软件，也不需要部署数据仓库服务器，就可以在几分钟之内获得高性能、高可靠的企业级数据仓库集群。  
您只需点击几下鼠标，就可以轻松完成应用程序与数据仓库的连接、数据备份、数据恢复、数据仓库资源和性能监控等运维管理工作。
- 提供一键式异构数据库迁移工具  
DWS提供配套的迁移工具，可支持Oracle和Teradata的SQL脚本迁移到DWS。

**高性能**

- 云化分布式架构  
DWS采用全并行的MPP架构数据库，业务数据被分散存储在多个节点上，数据分析任务被推送到数据所在位置就近执行，并行地完成大规模的数据处理工作，实现对数据处理的快速响应。
- 查询高性能，万亿数据秒级响应  
DWS后台通过算子多线程并行执行、向量化计算引擎实现指令在寄存器并行执行，以及LLVM动态编译减少查询时冗余的条件逻辑判断，助力数据查询性能提升。  
DWS支持行列混合存储，可以同时为用户提供更优的数据压缩比（列存）、更好的索引性能（列存）、更好的点更新和点查询（行存）性能。
- 数据加载快  
DWS提供了GDS极速并行大规模数据加载工具。

### 高扩展

- 按需扩展：Shared-Nothing开放架构，可随时根据业务情况增加节点，扩展系统的数据存储能力和查询分析性能，最高支持2048节点规模。
- 扩容后性能线性提升：容量和性能随集群规模线性提升，线性比0.8。
- 扩容不中断业务：扩容过程中支持数据增、删、改、查，及DDL操作（DROP/TRUNCATE/ALTER TABLE），表级别在线扩容技术，扩容期间业务不中断、无感知。

### 高可靠

- ACID  
支持分布式事务ACID（Atomicity, Consistency, Isolation, Durability），数据强一致保证。
- 全方位HA设计  
DWS所有的软件进程均有主备保证，集群的协调节点（CN）、数据节点（DN）等逻辑组件全部有主备保证，能够保证在任意单点物理故障的情况下系统依然能够保证数据可靠、一致，同时还能对外提供服务。
- 安全  
DWS支持数据透明加密，同时可与数据库安全服务（DBSS）对接，基于网络隔离及安全组规则，保护系统和用户隐私及数据安全。DWS还支持自动数据全量、增量备份，提升数据可靠性。

### 融合分析

- 多模融合：支持流、时序、GIS、全文、AI等类型数据在库内直接计算，融合分析。
- 多源融合：与大数据无缝融合，您可以使用标准SQL查询HDFS、OBS上的数据，数据无需搬迁。
- 加速集群：基于OBS数据访问，共享Express加速集群，提供更高效的融合计算分析能力。

### 高安全

- 透明加密：对数据库的数据文件进行加密，可以避免恶意攻击者在OS层面绕过数据库权限控制机制或窃取磁盘直接访问用户数据的情况。

- 数据脱敏：内置数值、字符、时间类型脱敏函数，并提供可自定义的脱敏规则，实现在大数据高效互访的同时，对敏感数据进行有效保护。

# 2 DWS 支持平台和技术指标

## 2.1 技术指标

### 软件指标

DWS的技术指标如表2-1所示。

表 2-1 技术指标

技术指标	最大值
数据容量	20PB
集群节点数	2048
单表大小	1PB
单行数据大小	1GB
每条记录单个字段的大小	1GB
单表记录数	$2^{55}$
单表列数	1600
单表中的索引个数	无限制
单表索引包含列数	32
单表约束个数	无限制

技术指标	最大值
并发连接数	分钟级复杂查询并发数：80 秒级短查询并发数：500 毫秒级短事务并发数：5000 <b>说明</b> 集群中并发连接数的最大值基于数仓规格48 vCPU和64 vCPU。例如，标准数仓云盘单节点规格dwsk.12xlarge（48 vCPU   384GB   24000GB SSD）或dwsx2.16xlarge.m7（64 vCPU   512GB   32000GB SSD）。
分区表的分区个数	32768
分区表的单个分区大小	1PB
分区表的单个分区记录数	2 <sup>55</sup>

## 产品性能

DWS的产品性能指标如表2-2所示。

表 2-2 产品性能指标

性能项	支持指标	服务器配置
数据导入能力	单节点80MB/s（行存表）， 150MB/s（列存表）	CPU: 2路12核 E5-2690 内存: 256GB 硬盘: 20块 600GB SAS盘 网络: 10GE
数据导出能力	单节点100MB/s	
全表扫能力	单节点3亿条/s	
点查询能力	万亿条精确查询秒级响应	
分组能力（group by）	单节点2000万条/s	
连接能力（join）	单节点500万条/s	
排序能力（order by）	单节点300万条/s	

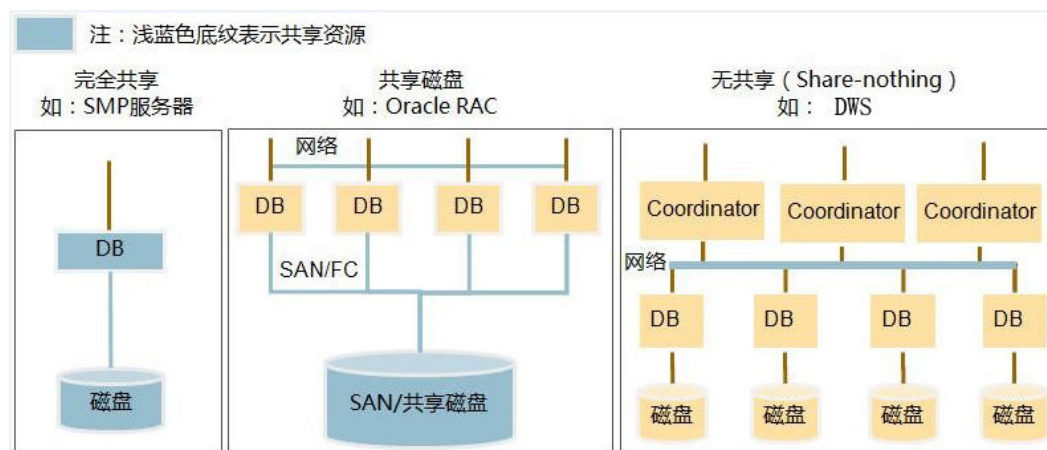
# 3 DWS 核心技术

## 3.1 Share-nothing 架构

DWS是采用Shared-nothing架构的MPP系统，它是由众多拥有独立且互不共享CPU、内存、存储等系统资源的逻辑节点组成。在这样的系统架构中，业务数据被分散存储在多个物理节点上，数据分析任务被推送到数据所在位置就近执行，通过控制模块的协调，并行地完成大规模的数据处理工作，实现对数据处理的快速响应。

Share-nothing又称为无共享架构，和其他架构的对比请参见图3-1。

图 3-1 架构对比



Share-nothing架构具备如下优点：

- 最易于扩展的架构。
  - 为BI和数据分析的高并发、大数据量计算提供按需扩展的能力。
  - 自动化的并行处理机制。
- 内部处理自动化并行，无需人工分区或优化。
  - 数据加载和访问方式与一般数据库相同。
  - 数据分布在所有的并行节点上。

- 每个节点只处理其中一部分数据。
- 最优化的I/O处理。
  - 所有的节点同时进行并行处理。
  - 节点之间完全无共享，无I/O冲突。
- 增加节点实现线性扩展：增加节点可线性增加存储、查询和加载性能。

## 3.2 分布式环境的数据布局

### 技术背景

为了解决PB级海量数据的高性能查询和数据导入，DWS采用了两层数据布局机制来利用并发度提高性能：第一层，用户可在创建表时指定数据分布策略（Hash分布、复制分布），数据写入系统时根据对应的分布策略确定存储在哪个节点上。第二层，节点内部数据进一步通过分区规则进行细分。

### 3.2.1 数据分布式存储

DWS采用水平分表的方式，将业务数据表的元组分散存储到各个节点内，该优势在于，查询中通过查询条件过滤不必要的数据，快速定位到数据存储位置，可极大提升数据库性能。

水平分表方式将一个数据表内的数据，按合适分布策略分散存储在多个节点内，DWS支持如表3-1所示的数据分布策略。用户可在CREATE TABLE时增加DISTRIBUTE BY参数，对指定的表应用数据分布功能。

表 3-1 分布式策略

策略	描述	适用场景	优势与劣势
复制表 (Replication)	集群中每一个DN实例上都有一份全量表数据。	小表、维度表。	<ul style="list-style-type: none"> <li>• Replication优点是每个DN上都有此表的全量数据，在join操作中可以避免数据重分布操作，从而减小网络开销，同时减少了plan segment（每个plan segment都会起对应的线程）。</li> <li>• Replication缺点是每个DN都保留了表的完整数据，造成数据的冗余。一般情况下只有较小的维度表才会定义为Replication表。</li> </ul>
哈希表 (Hash)	表数据通过hash方式散列到集群中的所有DN实例上。	数据量较大的事实表。	<ul style="list-style-type: none"> <li>• 在读/写数据时可以利用各个节点的IO资源，大大提升表的读/写速度。</li> <li>• 一般情况下大表（1000000条记录以上）定义为Hash表。</li> </ul>



策略	描述	适用场景	优势与劣势
轮询表 (Roundrobin) <b>8.1.2及以上版本支持</b>	表的每一行被轮番地发送给各个DN，数据会被均匀地分布在各个DN中。	数据量较大的事实表，且使用Hash分布时找不到合适的分布列。	<ul style="list-style-type: none"> <li>Roundrobin优点是保证了数据不会发生倾斜，从而提高了集群的空间利用率。</li> <li>Roundrobin缺点是无法像Hash表一样进行DN本地化优化，查询性能通常不如Hash表。</li> <li>一般在大表无法找到合适的分布列时，定义为Roundrobin表，若大表能够找到合适的分布列，优先选择性能更好的Hash分布。</li> </ul>

## 3.2.2 数据分区

### 功能描述

数据分区是数据库产品普遍具备的功能。在DWS分布式系统中，数据分区是在一个节点内部按照用户指定的策略对数据做进一步的水平分表，将表按照指定范围划分为多个数据互不重叠的部分（Partition）。

DWS支持范围分区（Range Partitioning）和List分区功能，即根据表的一列或者多列，将要插入表的记录分为若干个范围（这些范围在不同的分区里没有重叠），然后为每个范围创建一个分区，用来存储相应的数据。用户在CREATE TABLE时增加PARTITION参数，即表示针对此表应用数据分区功能。

### 效果收益

例如，[表3-2](#)描述了一个xDR（详单）场景下，基于时间分片的方式分区后带来的收益。

表 3-2 分区收益

场景描述	收益
当表中访问率较高的行位于一个单独分区或少数几个分区时。	大幅减少搜索空间，从而提升访问性能。
当需要查询或更新一个分区的大部分记录时。	仅需要连续扫描对应分区，而非扫描整个表，因此可大幅提升性能。
当需要大量加载或者删除的记录位于一个单独分区或少数几个分区时。	可直接读取或删除对应分区，从而提升处理性能；同时由于避免大量零散的删除操作，可减少清理碎片工作量。

数据分区带来的好处在于：

- **改善可管理性：**利用分区，可以将表和索引划分为一些更小、更易管理的单元。这样，数据库管理员在进行数据管理时就能采取“分而治之”的方法。有了分区，维护操作可以专门针对表的特定部分执行。
- **可提升删除操作的性能：**删除数据时可以删除整个分区，与分别删除每行相比，这种操作非常高效和快速。  
删除分区表与删除普通表的语法一致，都是通过DROP TABLE语法进行删除。
- **改善查询性能：**通过限制要检查或操作的数据数量，分区可带来许多性能优势。  
**分区剪枝：**分区剪枝（也称为分区消除）是CN在执行时过滤掉不需要扫描的分区，只对相关的分区进行扫描的技术。分区剪枝通常可以将查询性能提高若干数量级。  
**智能化分区联接：**通过使用一种称为智能化分区联接的技术，分区还可以改善多表联接的性能。当将两个表联接在一起，并且至少其中一个表使用联接键进行分区时，可以应用智能化分区联接。智能化分区联接将一个大型联接分为多个较小的联接，这些较小的联接包含与联接的表“相同”的数据集。这里，“相同”定义为恰好包含联接的两端中相同的分区键值集，因此可以确保只有这些“相同”数据集的联接才会有效，而不必考虑其他数据集。

### 3.2.3 数据并行导入

#### 实现原理

数据并行导入（加载）的核心思想是充分利用所有节点的计算能力和I/O能力以达到最大的导入速度。DWS的数据并行导入实现了对指定格式（支持CSV/TEXT格式）的外部数据高速、并行入库。

所谓高速、并行入库是和传统的使用INSERT语句逐条插入的方式相比较。并行导入过程中：

- CN只负责任务的规划及下发，把数据导入的工作交给了DN，释放了CN的资源，使其有能力处理外部请求。
- 各个DN都参与数据导入的工作，充分利用各个设备的计算能力及网络带宽，提高数据导入的整体性能。

下面，将以Hash分布策略为例介绍DWS的数据导入过程。数据并行导入的流程图请参见图3-2。

图 3-2 数据并行导入

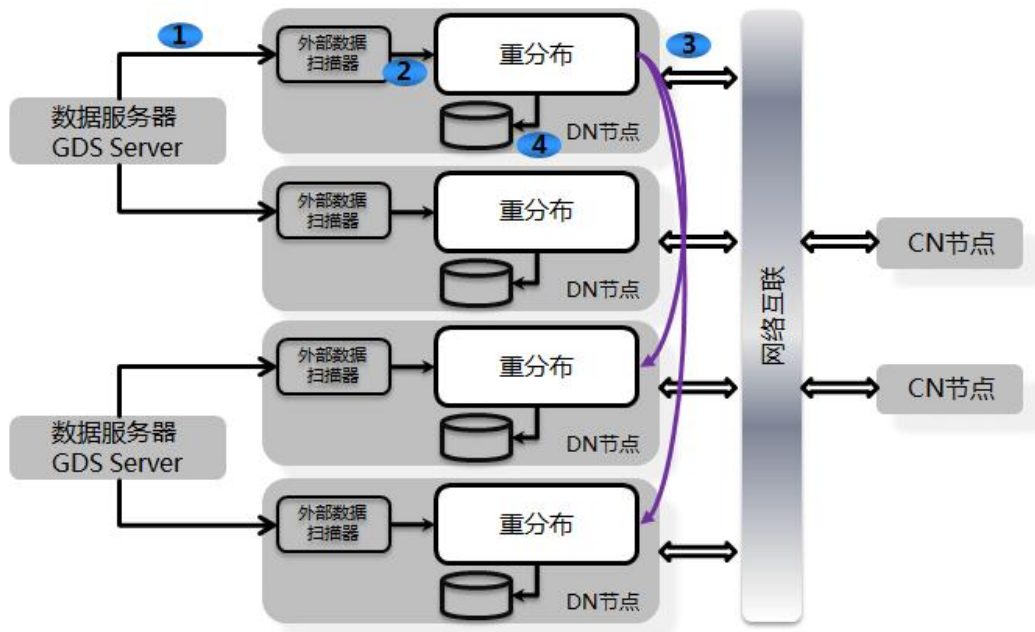


表 3-3 流程说明

流程	说明
创建Hash分布策略的表	业务应用在CREATE TABLE时预先设定Hash分布策略（指定表的某个属性作为分布字段）。
设定分区策略	应用程序在CREATE TABLE时还可以预先设定分区（指定表的一个属性作为分区字段），每个数据节点内部的每个Hash的数据都将按照设定的分区规则做相同的分区处理。
①	启动数据导入后，GDS将指定的数据文件分割成固定大小的数据块。
②	每个数据节点并行的从GDS下载这些数据块。
③ ④	各个数据节点并行的处理数据块，从中解析出一条数据元组，每一个元组根据分布列计算出来的Hash值判断存储的物理位置： <ul style="list-style-type: none"> <li>• 如果Hash在其他网络节点，则需要通过网络重分布到目标数据节点。</li> <li>• 如果Hash在本地节点，则存储在本地数据节点。</li> </ul>
数据写入分区	数据到达Hash所在的节点后还将根据分区逻辑写入对应的分区数据文件。 在数据写入分区表时，DWS还提供了Exchange（交换分区）的技术来提升写入性能。
GDS：全称General Data Service，GDS服务用来管理数据源，可以在数据服务器上部署多个GDS服务来提升数据加载的性能。	

### 3.3 全并行的数据查询处理

#### 功能描述

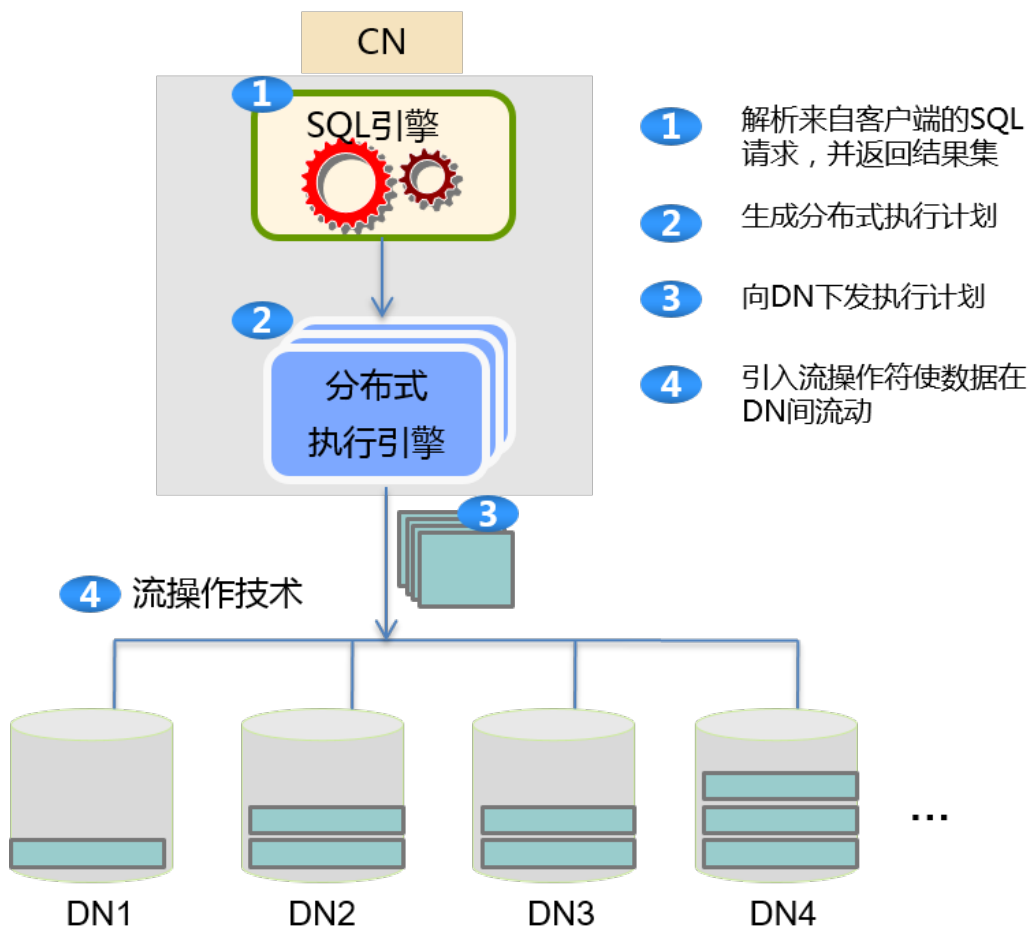
全并行的分布式查询处理是DWS中最核心的技术，它可以最大限度的降低查询时节点之间的数据流动，以提升查询效率。

DWS为达成高性能数据分析目标，实现了一套高性能的分布式执行引擎，执行引擎以SQL引擎生成的执行计划为输入，将元组按执行计划的要求进行加工并将结果返回给客户端。

#### 技术原理

图3-3展示了DWS的全并行分布式查询技术。

图 3-3 分布式查询示意图



- 运行在CN上的分布式执行引擎实现了分布式执行调度的功能。
- 节点内引入新的执行算子来支撑数据在计算节点之间的流动，这些新的执行算子称其为数据流操作符，根据数据流的输入、输出关系，可以细分为聚合流（Gather）、广播流（Broadcast）和重分布流（Redistribution）。聚合流将数

据从多个查询片段聚合到一个。广播流将数据从一个查询片段的数据向多个传输。重分布流则将多个查询片段的数据，按照一定规则重组后向多个传输。

- 跨计算节点的数据传输依赖于查询分析阶段根据数据分布以及代价模型构建的数据流动拓扑结构，并根据此结构来建立节点之间的网络连接，驱动数据流动于此拓扑结构之上。

## 3.4 向量化执行和行列混合引擎

### 技术背景

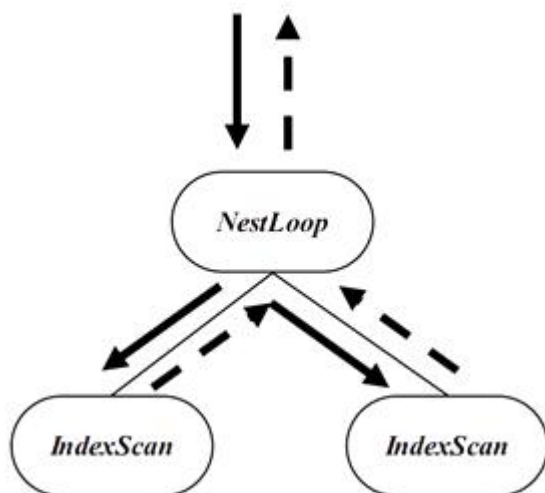
在大宽表，数据量比较大、查询经常关注某些列的场景中，行存储引擎查询性能比较差。例如气象局的场景，单表有200~800个列，查询经常访问10个列，在类似这样的场景下，向量化执行技术和列存储引擎可以极大的提升性能和减少存储空间。

### 向量化执行

标准的迭代器模型如图3-4所示。控制流向下（下图实线）、数据流向上（下图虚线）、上层驱动下层（上层节点调用下层节点要数据）、一次一元组（下层节点每次只返回一条元组给上层节点）。

而向量化执行相对于传统的执行模式改变是对于一次一元组的模型修改为一次一批元组，配合列存特性，可以带来巨大的性能提升。

图 3-4 向量化执行引擎



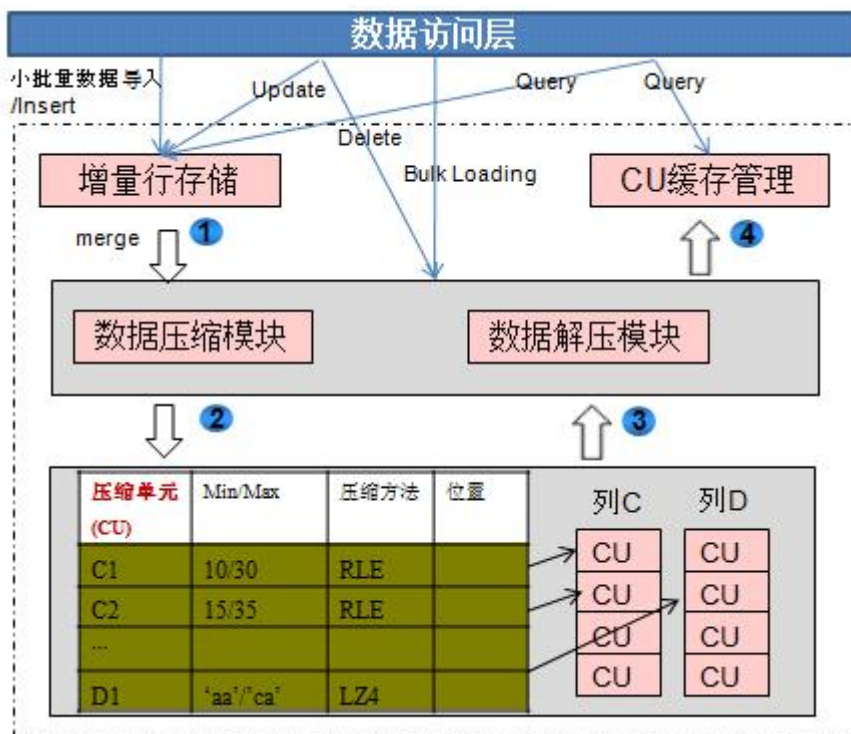
### 行列混合存储引擎

DWS支持行存储和列存储两种存储模型，用户可以根据应用场景，建表的时候选择行存储还是列存储表。

一般情况下，如果表的字段比较多（大宽表），查询中涉及到的列不很多的情况下，适合列存储。如果表的字段个数比较少，查询大部分字段，那么选择行存储比较好。

如图3-5所示，行列混合存储引擎可以同时为用户提供更优的数据压缩比（列存）、更好的索引性能（列存）、更好的点更新和点查询（行存）性能。

图 3-5 行列混存引擎



列存模式下支持数据压缩，对于非活跃的早期数据可以通过压缩来减少空间占用，降低采购和运维成本。

DWS列存储压缩支持Delta Value Encoding、Dictionary、RLE、LZ4、ZLIB等压缩算法，且能够根据数据特征自适应的选择压缩算法，平均压缩比7:1。压缩数据可直接访问，对业务透明，极大缩短历史数据访问的准备时间。

当前列存储引擎有以下约束：

- DDL仅支持CREATE/DROP/TRUNCATE TABLE的功能。  
兼容分区的DDL管理功能（如：ADD/DROP/MERGE PARTITION，EXCHANGE功能）。  
支持CREATE TABLE LIKE语法。  
支持ALTER TABLE的部分语法。  
其他功能都不支持。
- DML支持UPDATE/COPY/BULKLOAD/DELETE。
- 不支持触发器，不支持主外键。
- 支持Psort index、B-tree index和GIN index，具体约束参见“[CREATE INDEX](#)”章节。

## 3.5 资源监控及管控

### 技术背景

为实现资源的合理利用，GaussDB提供了资源监控及管控手段，对影响作业运行的计算资源和存储资源进行分配和利用，通过对系统资源的合理分配，避免发生资源的

合理占用导致系统运行效率下降甚至引发运行问题。资源监控及管控包括资源监控、负载管理及磁盘空间管控。

## 3.5.1 负载管理

### 功能描述

负载管理主要通过业务的并发控制实现系统计算资源的均衡，避免业务间出现资源争抢，实现作业的和谐共处，达到资源利用最优，同时通过引入cgroup技术实现了cpu的配额管理和限额管理。

### 技术原理

负载管理分为静态负载管理和动态负载管理，使用参数enable\_dynamic\_workload控制，同时GaussDB提供优先级控制用于对租户作业优先级进行控制。

- 静态负载管理  
以CN为单位实现租户的并发控制和资源管控，各CN分别进行内存管控和并发控制，集群实际并发数为各CN并发之和，集群实际使用内存为各CN运行作业所占内存之和。因为各CN分别进行内存管控，因此DN上租户实际使用内存有可能超过租户内存限制。
- 动态负载管理  
增加CCN用于复杂作业并发控制和内存管控，各CN向CCN请求排队信息，CCN将租户运行作业估算内存累加得到各租户已用内存，租户可用内存不足时触发排队，租户作业运行结束后尝试唤醒本租户正在排队的作业；各租户已用内存之和累加得到集群已用内存，同时CCN定期收集DN上内存使用信息用于更新集群可用内存，可用内存不足以运行作业时触发排队，作业运行需同时满足租户内存限制和集群内存限制。
- 优先级控制  
基于query\_band可以实现多维度的负载识别和队列内优先级控制，一方面提供了更为灵活的负载识别手段，可根据作业类型、应用名称、脚本名称等识别负载队列，使用户根据业务场景可灵活配置query\_band识别队列；另一方面实现了队列内作业下发优先级控制，后续将逐步实现队列内资源优先级控制。

### 效果收益

负载管理可以实现租户间的资源隔离及优先级控制，在保证租户间资源互不影响的情况下保证高优先级租户作业的优先运行，达到资源的有效及最优利用。

了解更多请参见[资源管理](#)。

## 3.5.2 空间管控

### 功能描述

用户空间管控包含三部分：永久表空间管控、临时表空间管控以及中间计算结果集落盘空间管控，数据库管理员创建组用户和业务用户时，可以指定用户永久表空间、临时表空间以及中间计算结果集落盘空间限额。在用户执行具体操作涉及到受管控空间的增加减少时，系统会根据操作对象所属用户当前指定的限额进行判断是否超过限额，如若超出限制，则不允许进行当前操作，将可使用磁盘空间限制到所设定的空间额度之中。

除此之外，产品还支持schema粒度的永久空间管控，当操作该schema内的对象进行空间增删时，对超限操作进行管控。

## 技术原理

业务用户执行业务时，其操作的数据空间大小将记录在业务用户的空间配额中；对于永久表空间，每个DN节点分摊空间使用配额大小。DN在数据文件、临时表文件、临时文件的新增和扩展时进行实时的空间统计，当业务用户进行业务操作时，判断是否已达到空间限额，决定是否运行该操作，超出限额的进行回滚。在事务或者session结束时，释放空间并刷新对应的数值。

DN节点会实时统计用户执行具体操作时涉及到的磁盘变化，保存到内存中，等待CN进行统计，并且会接收CN下推的信息来更新整体系统所使用的临时空间，以正确进行管控。

## 效果收益

通过限制用户使用空间的最大使用限额，保证用户业务不会出现挤压或者侵占过多存储资源而导致的正常业务数据无法写入的情况，从而保证系统的稳定性和数据可用性。

了解更多请参见[空间管理](#)。

## 3.6 分布式事务

### 技术背景

在分布式share nothing架构下，表的数据分布在不同的节点上。客户端的一条或多条语句可能会同时修改多个节点上的数据，这种情况下，会产生分布式事务。分布式事务需要关注：

- 在各个节点上事务的原子性，分布式事务在所有节点上要么全部成功要么全部失败。
- 事务的一致性，查询在各个节点上返回一致的数据。当节点出现故障时，是否会返回不一致的数据。

其中，分布式事务的原子性是必须要满足的，分布式事务的一致性根据CAP理论会有不同的标准，常见的是支持强一致2pc、3pc协议等的CP系统，或者支持最终一致性TCC、消息表等的AP系统。

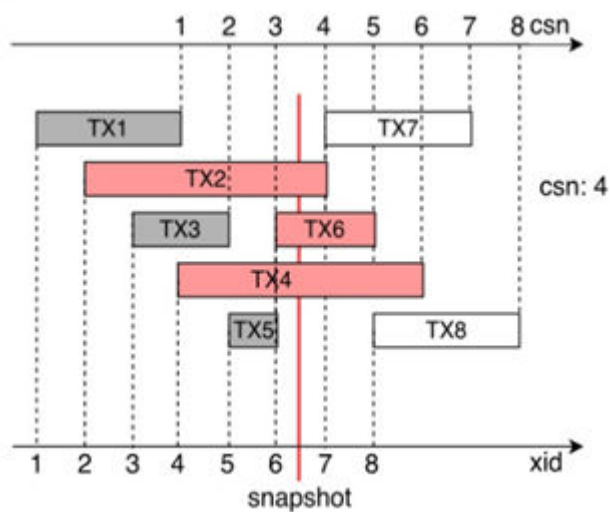
### 技术原理

DWS支持强一致的分布式事务，在满足CP的基础上最大程度的提供高可用服务。



## CSN

图 3-6 CSN 的更新以及与 XID 的映射关系

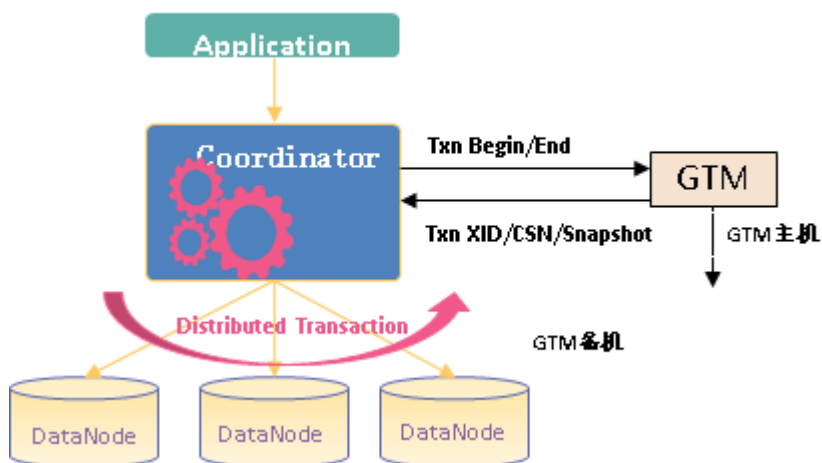


CSN ( Commit Sequence Number ) 表示事务提交号。

1. CSN是一个单调递增的8字节无符号整数，由GTM维护。
2. 事务结束时，从GTM上获取并更新CSN值。
3. 采用CSN机制后，查询只需从GTM上获取CSN号。

## GTM

图 3-7 GTM



GTM是GaussDB分布式框架下的一个组件：

- 管理和分配事务号（只增不减）。
- 管理和维护CSN号（只增不减）。

CN在执行修改操作时会从GTM上获取事务号。

CN在语句开始时会从GTM获取一个CSN用于查询。

CN在事务开始和结束会与GTM通信，注册和销毁事务相关信息。

## 故障处理

在节点出现故障造成分布式事务残留后，通过gs\_clean工具进行自动清理。

gs\_clean从各个节点上查询残留的两阶段事务，根据残留的事务号，去其它节点上查看该事务提交还是回滚，并根据最终结果对两阶段残留事务进行清理。

## 效果收益

DWS支持强一致的分布式事务，用户可以像使用单机数据库一样来使用DWS数据库，不用担心事务问题，非常简单方便。并且在采用基于CSN的事务机制后，并发性能有了很大的提升。

# 3.7 在线扩容

## 技术背景

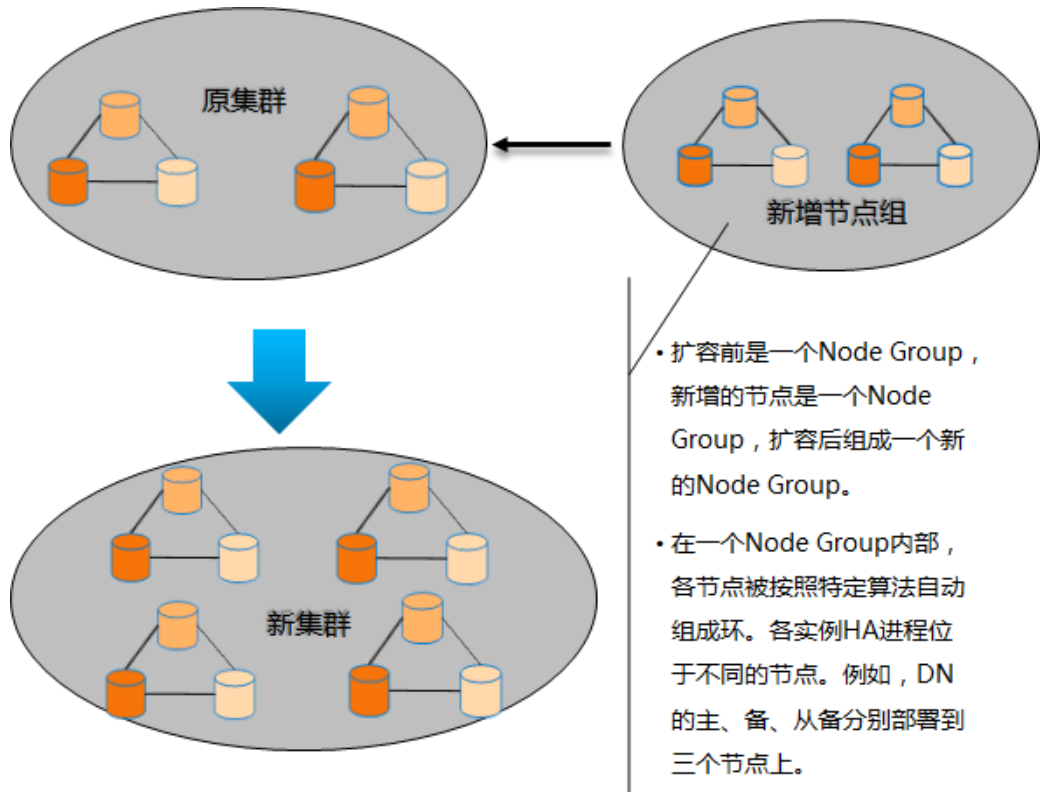
随着客户业务的发展，现有系统在磁盘容量、性能等方面将逐步呈现瓶颈。DWS分布式数据库集群提供scale-out线性扩展能力，满足客户业务增长和利旧的诉求（将闲置的机器加入系统）。

## 技术原理

DWS采用Node Group技术，支持多表并行扩容，扩容速度高达400G/小时/新增节点。

DWS的扩容过程示意如图3-8所示。

图 3-8 扩容示意图

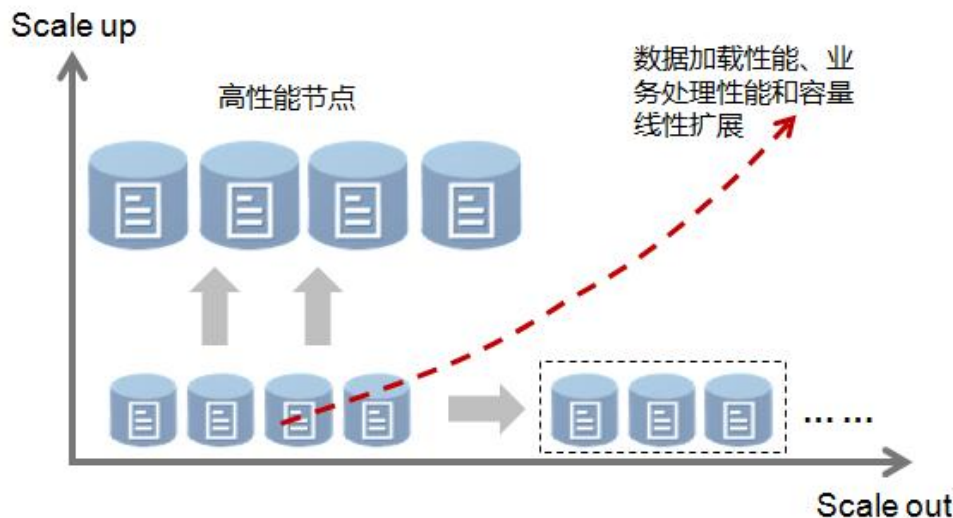


## 效果收益

DWS在线扩容具有如下特点：

- 扩容不中断业务。  
DWS扩容过程中支持数据持续入库、查询业务不中断。
- 一致性Hash技术和多表并行扩容等技术，助力扩容性能提升。  
采用一致性Hash技术使得重分布过程中需要迁移的数据量最小。  
重分布过程中支持多表并行及用户自主选择表的重分布顺序。  
支持扩容进度查询，实时监控扩容进度。
- 随着节点数增加，集群性能线性增长。  
如图3-9所示，在全并行分布式架构下，DWS随着节点数的增加，集群的数据加载性能、业务处理性能和容量可线性扩展。

图 3-9 高性能线性扩展



## 3.8 SQL on Anywhere

### 技术背景

当前用于大数据处理的引擎组件种类繁多，且各自提供了丰富的接口供用户使用。但对传统数据库用户来说，SQL语言依然是最熟悉和方便的一种接口。如果能在一个客户端中使用SQL语句操作不同的大数据组件，将极大提升使用各种大数据组件的效率。

DWS支持SQL on Anywhere，基于DWS可以操作Hadoop、Spark，构筑起统一的大数据计算平台。

### 3.8.1 SQL on Hadoop

DWS支持直接读取存储在Hadoop HDFS文件系统上的结构化数据，对外提供标准SQL语言查询接口，通过向量化引擎完成Hadoop原生数据的复杂分析查询工作。

### 技术架构

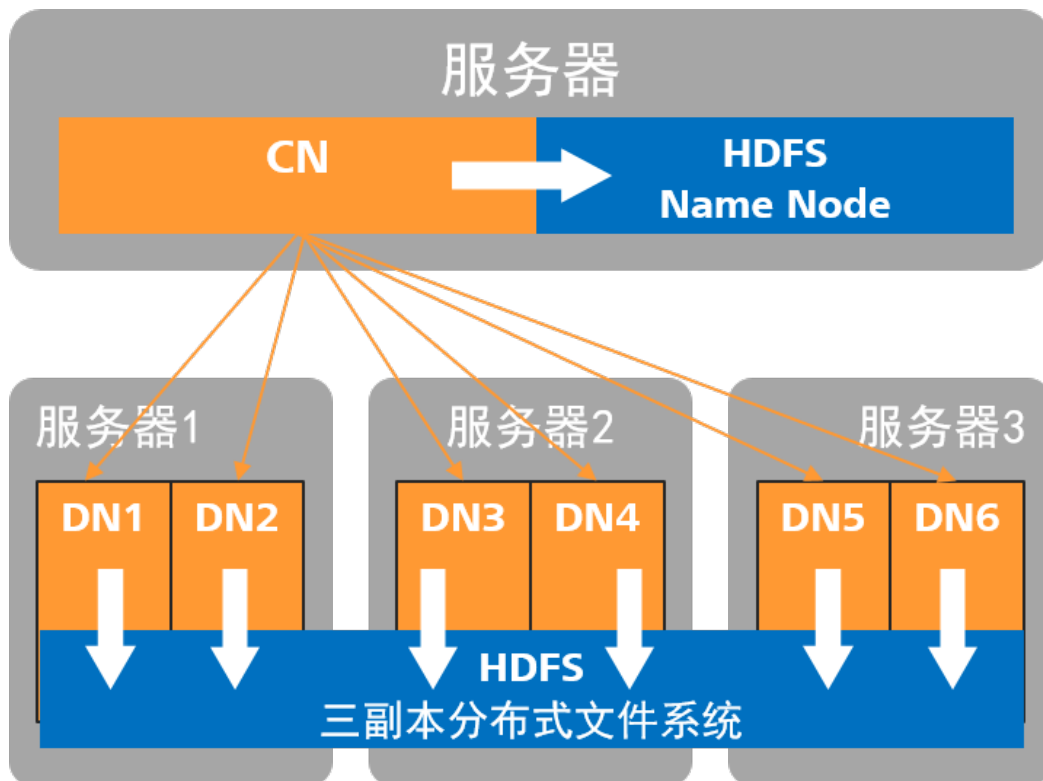
DWS将HDFS上存储的结构化数据映射为外表表，从而重用已有的成熟数据库SQL编译器和向量化执行引擎，对外提供标准SQL接口，完成HDFS上Hadoop原生数据的分析查询。

DWS的SQL编译器中引入了新的Scheduler（调度器）组件。在SQL编译时，如果有针对HDFS外表表的访问，Scheduler组件会访问HDFS Name Node元数据信息，获得外表文件在HDFS上的存储路径。CN的SQL引擎在编译查询语句时，将会调用Scheduler获得并指定每个DN将要访问的HDFS文件的具体路径。

HDFS外表数据文件在DN间的分配原则是：

- 首先以本地读为主任务分配策略。
- 其次考虑DN之间的任务负载均衡。

图 3-10 逻辑架构



## 技术特点

- 支持标准SQL查询语法**  
 完全继承GaussDB(DWS)的SQL查询语法的支持，实现HDFS分布式文件系统上的结构化数据分析查询，支持连接、聚合、字符日期等系统函数，支持子查询以及HDFS结构化数据和GaussDB(DWS)本地数据的联合访问，支持窗口函数等。
- 针对HDFS数据的代价估算模型**  
 在GaussDB(DWS)基于代价估算（Cost-based）的SQL优化器中，增加了针对HDFS分布式文件系统上结构化数据访问的代价估算模型，从而可以制定HDFS数据的最优执行计划。
- 智能扫描**  
 GaussDB(DWS)可以将适用的谓词条件直接下推到存储在HDFS上的Hadoop原生数据上，在压缩数据上进行谓词过滤，且针对Hadoop ORC（Optimized Row Columnar）列式存储实现后期物化（Late-materialization），从而显著减少HDFS数据读取。
- LLVM（Low Level Virtual Machine）优化**  
 基于智能扫描，将下推到Hadoop原生数据上的适用谓词进行LLVM优化，使其谓词条件生成中间表示IR，最终生成与平台相关的机器码，进而提高谓词过滤及查询性能。
- 支持信息约束（Informational Constraint）**  
 若表上的列具有唯一性特征，则在数据库中创建该表时指定对应列具有信息约束属性，在执行阶段利用该属性提供查询性能。
- 向量引擎**

针对HDFS上存储的ORC列式结构化数据，直接对接GaussDB(DWS)已有成熟的向量化执行引擎，提升分析查询性能。

- **支持分区表**

适配HDFS上依据Hive语法定义的分区表数据，由GaussDB(DWS) SQL优化器来实现自动的分区表访问的剪枝操作，提高分析查询性能。

- **分布式高效读取HDFS数据**

GaussDB(DWS) SQL编译器引入Scheduler组件，来达到GaussDB(DWS) Data Nodes的HDFS数据访问负载均衡，并可以通过HDFS组件的本地读特性（short-circuit）来提升数据读取性能。

了解更多：请参见[CREATE FOREIGN TABLE \(SQL on Hadoop or OBS\)](#)

### 3.8.2 基于 GDS 的集群互联互通

#### 技术背景

在大中型企业中，因数仓分层、业务规划等原因，通常会部署多个GaussDB(DWS)集群，并进行跨集群的数据同步。面向集群间的数据同步，通常需满足大数据量、并行执行等要求，如何提供高效的跨集群数据同步能力成为当下数据领域热点之一。

#### 功能描述

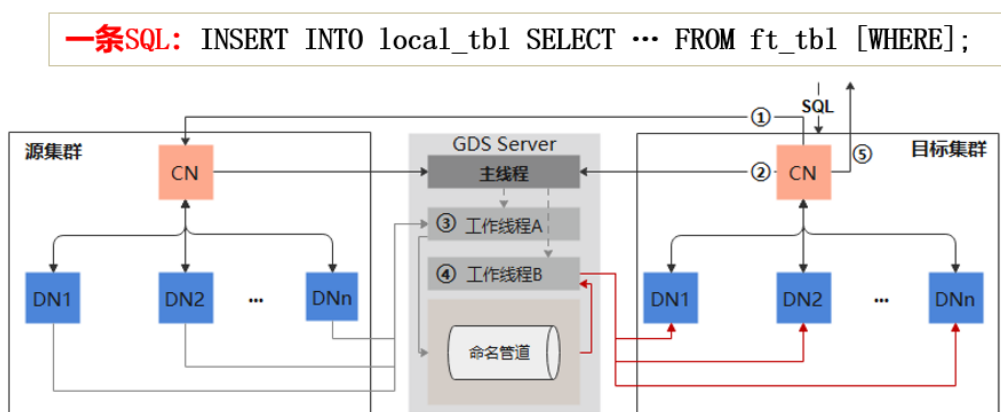
GaussDB(DWS)集群间全量数据迁移同步。

GaussDB(DWS)集群间过滤条件数据迁移同步。

#### 技术原理

将触发同步的“一条SQL”，通过查询重写转化为一对GDS不落地导入导出作业，分别在目标集群、源集群上执行，形成高效实时的数据中转通路，实现数据迁移同步。执行时，可以选择在目标集群或源集群上发起，分别对应“拉”、“推”两种方式。

图 3-11 拉取方式



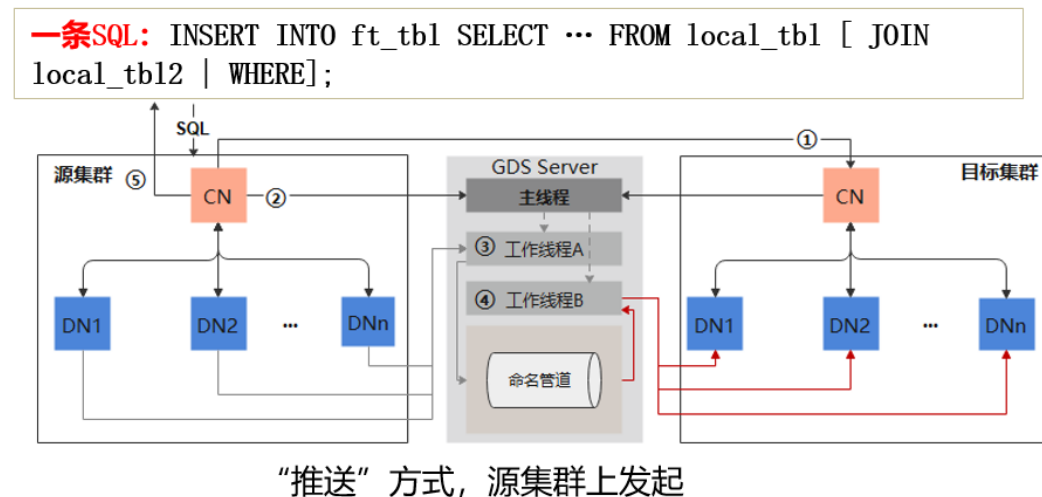
“拉取”方式，目标集群上发起

① 远程连接源集群，创建GDS只写外表，发起导出作业。

② 创建GDS只读外表，发起导入作业。

- ③ 工作线程A接收源集群数据，并写入本地文件。
- ④ 工作线程B读取本地文件数据，发送给目标集群。
- ⑤ 目标集群综合两端作业结果，得出最终结果返回给用户。

图 3-12 推送方式



- ① 远程连接目标集群，创建GDS只读外表，发起导入作业。
- ② 创建GDS只写外表，发起导出作业。
- ③ 工作线程A接收源集群数据，并写入命名管道。
- ④ 工作线程B读取命名管道数据，发送给目标集群。
- ⑤ 源集群综合目标集群导入作业结果、本集群导出作业结果，返回最终结果给用户。

## 效果收益

通过一条SQL启动迁移业务，借助GDS不落地，充分利用两端集群的节点算力，提供便捷、高效的GaussDB(DWS)集群间数据逻辑同步/搬迁能力，不占用磁盘空间，提高系统资源利用率。

了解更多请参见[基于GDS的跨集群互联互通](#)。

## 3.9 集群管理与高可用

### 功能描述

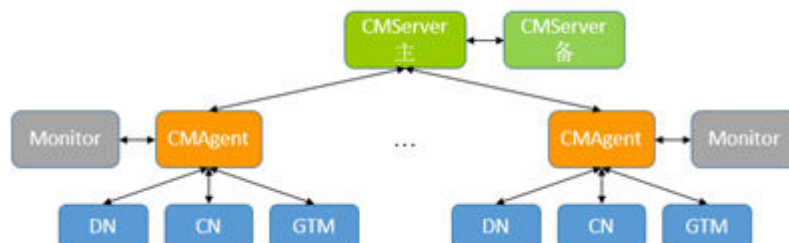
GaussDB(DWS)提供集群管理CM ( Cluster Manager ) 模块，负责管理和监控分布式系统中各个功能单元和物理资源的运行情况，确保整个系统的稳定运行。CM可分为主CM和备CM两部分。正常情况下，只由主CM提供GaussDB(DWS)集群管理服务；当主CM发生故障时，备CM会主动升为主CM提供集群管理服务。

### 技术原理

集群管理模块由CM Server，CM Agent和Monitor组件组成，并提供集群状态查询、集群起停、主备切换、实例重建等工具。CM Server只部署在主备CM上，作为整个

GaussDB(DWS)集群的大脑，负责处理CMAgent上报的各种状态信息，并决定是否需  
要状态变更。CMAgent部署于所有节点，作为实例代理进程，负责上报CN/DN/GTM  
等实例状态至CMServer，并接收和执行CMServer下发的命令。Monitor部署于所有节  
点，作为定时任务，在CMAgent停止的情况下将其重启。

图 3-13 集群管理模块架构



## Hang 检测

集群管理模块通过短连接机制，检测实例进程是否处在网络故障、磁盘IO阻塞、进程/  
线程僵死等异常状态，必要时触发DN/GTM主备切换或CN剔除流程。

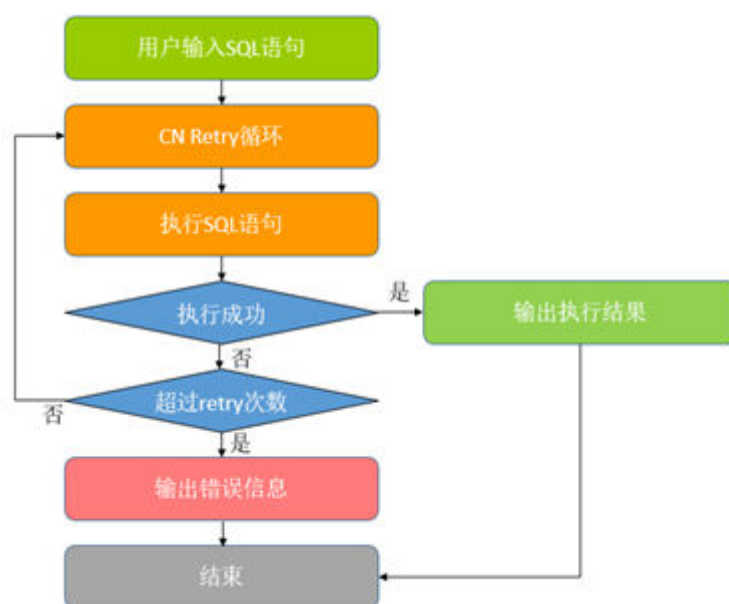
以DN为例，默认情况下CMAgent每隔180秒新建实例连接，若失败则每隔84秒进行重  
试。如果连续5次均连接失败，则触发DN主备切换流程。完整的hang检测周期大约为  
600秒。

## CN Retry

GaussDB(DWS)提供CN Retry功能用于SQL语句执行异常时的自动重试，提高业务的  
连续性。

CN在语句执行报错时触发retry机制。对于可重试的错误类型，将已执行的操作进行回  
滚，然后重新执行该语句；如果依然无法成功，则将错误信息上报给客户端。

用户不感知SQL语句的重试执行过程。





## 3.10 SQL 自诊断

### 技术背景

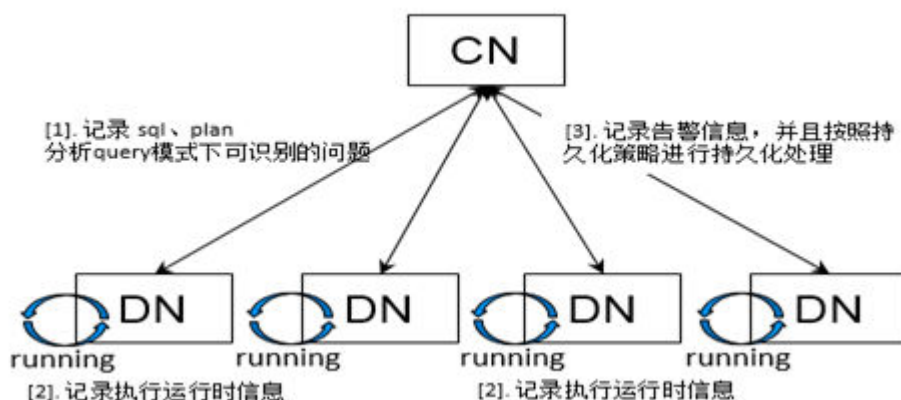
业务中包含大量查询，这些查询在执行计划、执行层面有什么样的问题，比如估算是否不准确、是否存在数据倾斜、是否存在统计信息未收集并且如何收集统计信息等。SQL自诊断为用户提供了一种更为高效易用的性能问题定位方法。主要帮助用户对批处理作业的SQL调优过程进行简化，希望输入SQL之后能够方便的批量得到SQL所存在的问题和针对问题给出的调优建议，而不是让用户依次去把作业集中的SQL单独提取出来改写，重现有性能问题的SQL语句并且通过查看Explain Performance依次排查问题，改写SQL，调整参数等复杂的调优。

### 功能描述

在执行作业之前，配置有关GUC参数参见《开发指南》中“resource\_track\_level”和“resource\_track\_cost”参数。运行用户作业之后，通过查看相关系统视图获得相关查询作业可能存在的性能问题。这些系统视图中会给出导致性能问题的可能原因，根据这些“性能告警”，参考《开发指南》中“SQL自诊断”章节，就可以对存在性能问题的作业进行调优。

### 技术原理

图 3-14 SQL 自诊断技术原理



1. CN端对SQL进行查询编译生成计划树，通过诊断分析器对执行计划树诊断出查询模式下可识别的问题（包括统计信息未收集和SQL不下推的告警）。
2. DN端执行SQL语句，在运行时将有关统计信息写入DN共享内存。如果当前SQL运行是算子模式（可以诊断所有支持的告警场景），则在SQL运行过程中收集算子执行统计信息。
3. DN端完成SQL的执行，向CN返回运行结果。如果是算子模式，则DN向CN端返回收集的运行时状态，并由诊断分析器对其内容进行分析，将有关分析结果进行记录。
4. 内存中统计信息3分钟会被清理一次。如果需要记录所有历史信息，打开GUC参数enable\_resource\_record，内存中统计信息每3分钟会被持久化到特定系统表中。

## 效果收益

场景描述	收益
多列/单列统计信息未收集	通过analyze收集统计信息，生成更好的执行计划。
SQL不下推	上报导致不下推的原因。通过SQL调优尽量将SQL下推。
HashJoin中大表做内表	数据量较大并且产生了下盘，对性能影响较大。通过SQL调优避免这种情况。
大表等值连接使用Nestloop	Nestloop在数据量较大时对性能影响较大。通过SQL调优避免这种情况。
大表Broadcast	在网络上发送大量数据，对性能影响较大。通过SQL调优避免这种情况。
数据倾斜	造成某些节点成为系统瓶颈。
索引不合理	索引不合理，导致扫描行数过多。修改索引。
估算不准	估算与实际执行行数偏差较大，导致选取计划不是最优。

## 3.11 透明加密

### 功能描述

透明数据加密（TDE）是指加密DWS的数据文件。通常在数据库的安全防护措施里面，可以采取一些消减措施来帮助保护数据安全。例如，设计一个安全系统、加密机密资产以及在数据库服务器的周围构建防火墙。但是，如果遇到物理介质（如硬盘）被黑客或者内部人员盗取的情况，恶意破坏方只需还原或附加数据库即可浏览用户数据。有一种解决方案是加密数据库中的敏感数据，并保护加密数据的密钥，该方案可以防止任何没有密钥的人使用这些数据，但这种保护必须事先计划。DWS提供了完整的解决方案TDE。

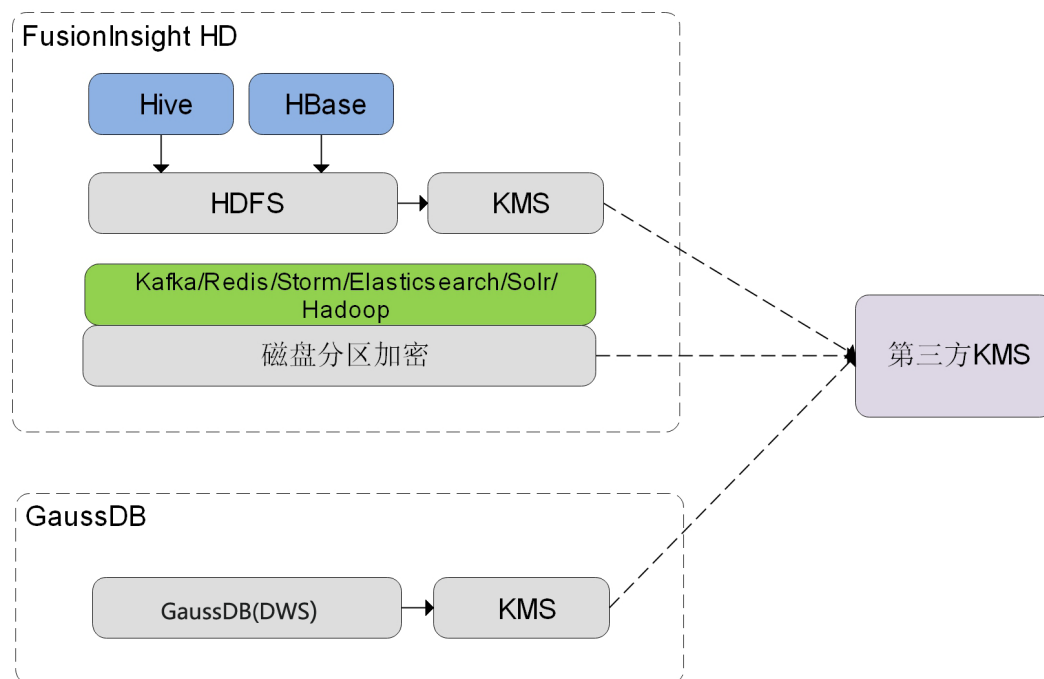
TDE可对数据实时I/O加密和解密，只要打开透明加密开关，对正常使用的用户是无感知的。这种加密使用数据库加密密钥（DEK），该密钥不会直接存储在数据库系统中。DEK是使用存储在KMS服务器的集群密钥（CEK）保护的对称密钥，数据库服务器仅仅保存其密文。在数据库启动阶段，数据库的每个CN/DN使用身份认证（例如Kerberos）来连接KMS服务器，并且解密DEK密文，从而获取到密钥明文，缓存在内存中。一旦机器下电或者集群关闭，密钥将会被清理。因此，需要保护好集群中的密钥文件，因为一旦丢失，则会造成不可恢复的危害。

### 使用场景

传统数据库集群中，用户数据明文保存在行存/列存文件中，集群的维护人员或者恶意攻击者可在OS层面绕过数据库的权限控制机制或者窃取磁盘直接访问用户数据。DWS引入了Hadoop KMS服务并进行增强，通过对接第三方KMS，可实现数据的透明加密，保障用户数据安全。

DWS通过Hadoop KMS从第三方KMS获取密钥。

图 3-15 数据存储加密



## DWS 透明加密

DWS数据库级透明加密，每个DWS集群有一个CEK，每个数据库单独配置DEK加密保护，DEK使用CEK加密保护，保存在DWS集群侧。密钥通过KMS服务申请和加解密，加密算法通过配置项统一配置。目前支持AES、SM4算法，其中SM4算法在hi 1620以上版本芯片中支持硬件加速。

当前支持数据库级别的透明加密，在创建集群的时候进行加密配置。

透明加密配置请参见[数据库加密](#)。

## 3.12 数据脱敏

### 技术背景

大数据时代中数据蕴藏的巨大价值同时也带来了隐私信息保护方面的难题，数据脱敏用于实现大数据高效共享的同时对敏感信息的安全保护。

### 功能描述

DWS支持客户以表的列为单元创建脱敏策略，客户可针对业务中的敏感数据进行策略创建，敏感数据的界定由客户结合自身业务场景识别。制定脱敏策略后，只有管理员和表对象的owner才能访问原始数据。同时，脱敏的数据可以参与实际运算及使用，仅在数据库服务最终返回结果时脱敏。

图 3-16 功能效果



## 技术原理

图 3-17 功能原理

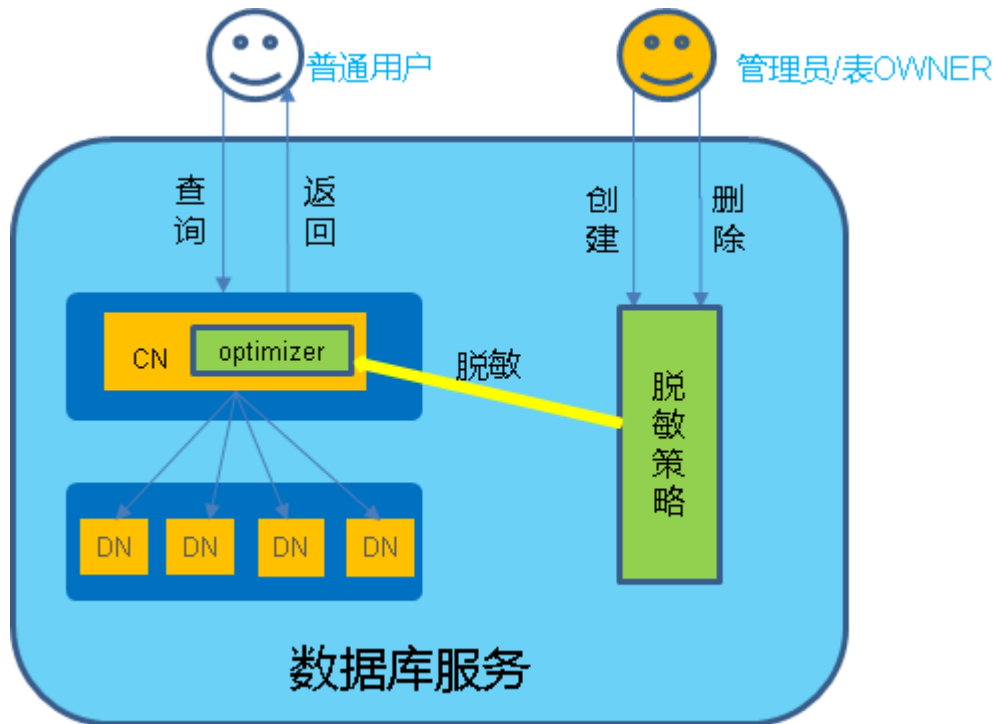


表 3-4 接口功能描述

外部接口名	接口功能
add_policy	创建脱敏策略。
alter_policy	修改脱敏策略。
drop_policy	删除脱敏策略。

外部接口名	接口功能
enable_policy	开启脱敏策略。
disable_policy	关闭脱敏策略。

在对表的列上创建好脱敏策略后，所有涉及脱敏列的查询便会受到脱敏策略的影响，只有管理员和表的OWNER才能查询返回脱敏列的原始值。

## 效果收益

涉及敏感信息的行业都对数据脱敏有着天然的需求，例如金融、政府、医疗等。其在应用开发、测试、培训等活动中都需要使用数据，因此都可以使用脱敏功能来避免敏感信息的泄露风险。

了解更多请参见[数据脱敏](#)。

## 3.13 备份与容灾

### 3.13.1 备份恢复

#### 快照功能

快照是DWS集群在某一时间点的完整备份，记录了这一时刻指定集群的所有配置数据和业务数据，用于还原创建快照时的集群数据。快照存储在OBS或NFS中。

#### 说明

- DWS提供了部分免费存储空间，用于存放您的快照数据。当快照数据存储空间超过免费空间大小时，超出部分按照OBS的计费规则进行计费。
- 免费空间大小是您的集群的总存储空间大小，即单节点存储空间大小 \* 节点数。
- 快照管理功能依赖OBS、SFS-Tubor服务。

快照中包含集群上运行的数据库的数据以及集群的相关信息，其中包括节点数量、节点规格和管理员用户名称等。如果您需要从快照恢复集群，DWS会使用这些集群信息创建新的集群，然后从快照数据中还原所有数据库。DWS从快照创建的新集群与制作快照的原始集群具有相同的配置（包括节点的数量和规格）。当您把快照恢复为新集群时，如果您没有指定其他值，则参数默认与快照中的备份信息保持一致。

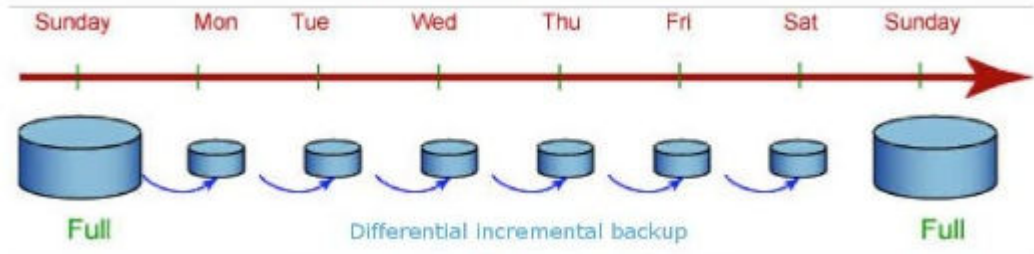
快照有两种类型：手动快照和自动快照。

#### 自动快照的备份与恢复策略

自动快照采用差异增量备份，第一次创建自动快照为全量备份，并且以后每间隔一段时间就会做一次全量备份，全量备份作为基础版本。两次全量备份之间都是做增量备份，增量备份记录基于前一次备份所发生的更改。在恢复快照时，DWS会将最近一次的全量备份到本次备份之间的所有备份一起用于恢复集群，因此不会产生数据丢失。为了保证每个增量快照都能够正常进行数据恢复，如果增量快照的保存时间超过了保留天数，DWS不会立即删除过期的增量快照，而是仍然会保留这部分快照，以便提供给后续其他增量快照恢复集群时使用。直到下一次做全量快照为止，DWS才会一并删

除已过期的上一次的**全量自动快照**和相关的**增量快照**。如果对已有**集群关闭自动快照**功能，则会删除该**集群所有的自动快照**，但**手动快照**不会被删除。

图 3-18 自动快照的备份与恢复策略



了解更多请参见**管理快照**。

### 3.13.2 集群容灾

#### 技术背景

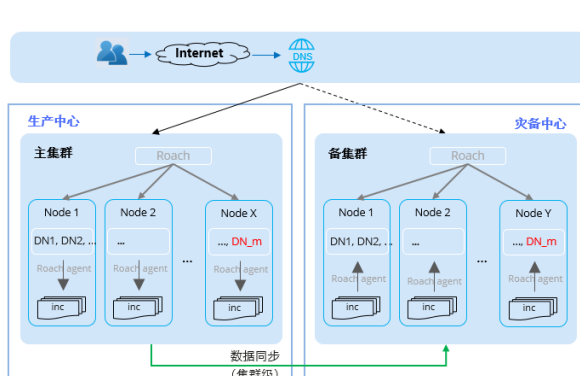
为了对于GaussDB(DWS)集群的提供容灾能力，采用双集群周期同步数据的方式，将主集群数据同步到跨AZ/Region的集群。当前的双集群方式是对比了业界几种典型的方式：应用侧双写，异地多副本之后，结合GaussDB(DWS)自身的架构特点以及容灾方案的复制性而确定的。

#### 技术原理

双集群容灾架构说明：

- 数据同步方式：两个DN数相同的集群间，使用节点间点对点的同步数据，充分复用MPP分布执行的特点。两个集群间需要打通网络，配置互信并确保DN数一致。
- 周期同步方式：通过可配置的时间间隔周期的进行数据同步。
- 数据内容：对于集群内的数据，行存、列存、lib库，配置信息都会同步。

图 3-19 双集群容灾



架构说明：

- 跨数据中心（跨AZ/跨Region/线下），部署两套DWS主备集群，通过Roach工具进行异步复制
- 容灾数据范围：集群级

适用场景：

- 适用企业级数仓同城、异地容灾

RTO/RPO：

- RTO/RPO 30min~小时级，与业务相关

主要约束：

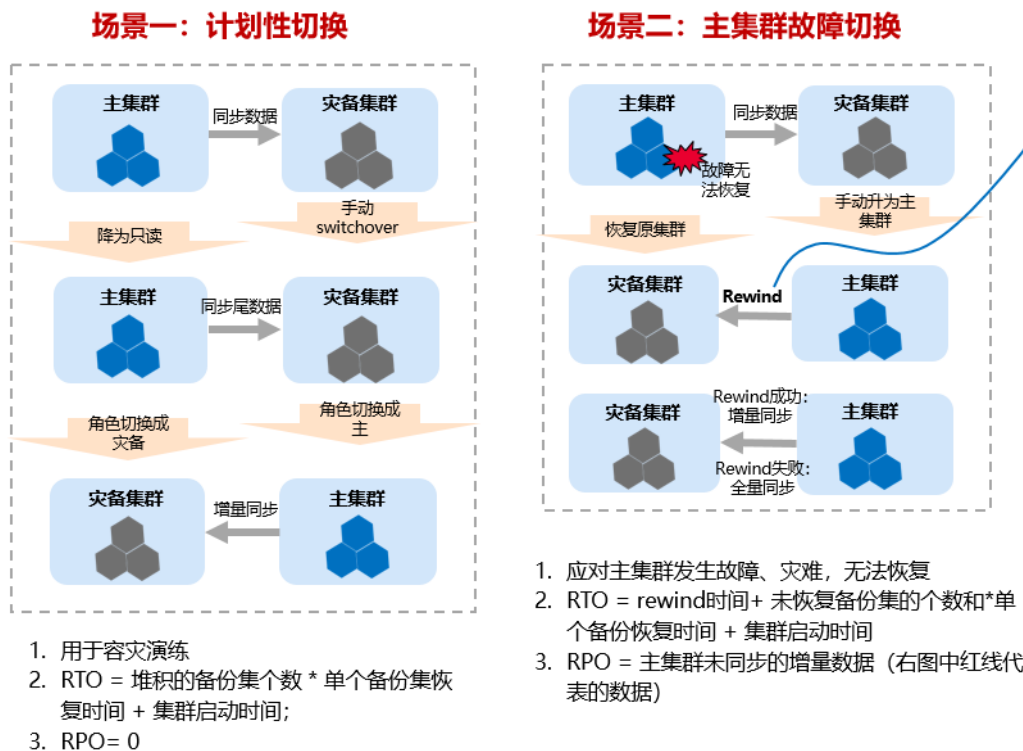
- 主备集群拓扑结构一致，DN数必须相同
- 主备集群间网络互通，可配互信

方案特点：

- 两套集群相对独立，主集群可读写，备集群非恢复期间可读
- 容灾架构不依赖底层存储，业务侧透明，可支撑PB级数据同步
- 支持容灾行存、列存数据

集群切换：

图 3-20 集群切换



容灾切换分两种，一种是计划性的切换，用于容灾演练，另一种是故障切换，在发生故障时进行切换。

对于计划性切换， $RPO = 0$ 的，主集群将数据同步到备集群后降备集群；故障切换时，备集群立即升主，此时 $RPO \neq 0$ 。

## 效果收益

两套集群的高可用容灾方案，耦合性比较松，是Oracle，DB2等厂商选择的方案。优点在于：

1. 两套集群是独立的，备集群不影响主集群。
2. 通过两套集群之间的切换，比较容易实现大版本的软件升级和应用升级。

有力支撑大规模MPP集群的数据安全保护能力。

了解更多请参见[集群容灾](#)。

# 4 DWS 工具

## 4.1 客户端工具

### 4.1.1 gsql

#### 技术背景

gsql是DWS提供在命令行下运行的数据库连接工具，可以通过此工具连接服务器并对其进行操作和维护，除了具备操作数据库的基本功能，gsql还提供了若干高级特性，便于用户使用。

#### 功能描述

- **连接数据库：**默认只支持从服务器本机连接，如果需要连接到远端的数据库，必须在服务端进行配置。
- **执行SQL命令：**支持交互式地键入并执行SQL命令，也可以执行一个文件中指定的SQL命令。
- **执行元命令：**元命令可以帮助管理员查看数据库对象的信息、查询缓存区信息、格式化SQL输出结果，以及连接到新的数据库等。

### 4.1.2 Data Studio

#### 功能描述

Data Studio是一款GUI工具，可以通过Data Studio连接数据库，执行并调试SQL语句和存储过程。Data Studio支持DWS的基本特性，为数据库开发人员提供图形界面，显著提高构建应用程序的效率，并简化数据库开发及应用开发任务。

Data Studio主要为数据库开发人员提供以下功能：

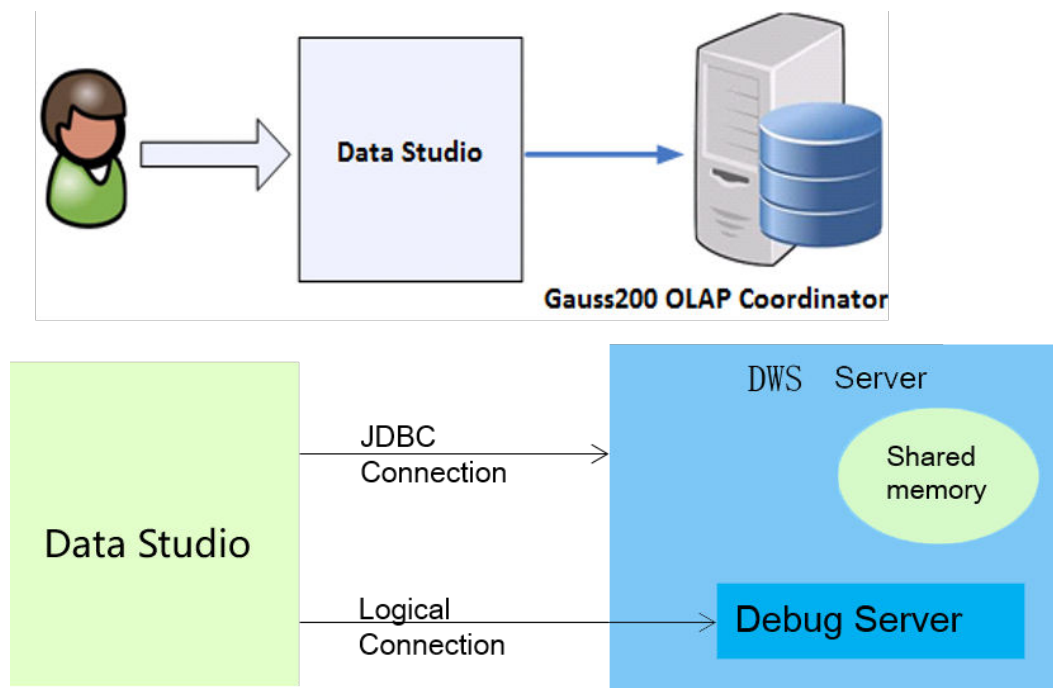
- 浏览数据库对象。
- 创建和管理数据库对象（例如：数据库、用户、表、索引）。
- 编辑和执行PL/SQL语句。



- 导入和导出表数据。
- 调试SQL语句和存储过程。

## 技术原理

图 4-1 Data Studio 组件交互图



如上图所示，Data Studio使用C/S结构，通过JDBC驱动与DWS数据库进行通信。

调试时，将使用两个连接：

- JDBC连接，用于执行查询。
- 逻辑连接，用于其它调试操作，例如断点和可变操作。

数据库服务和调试服务通过共享内存进行通信。

图 4-2 Data Studio 界面展示

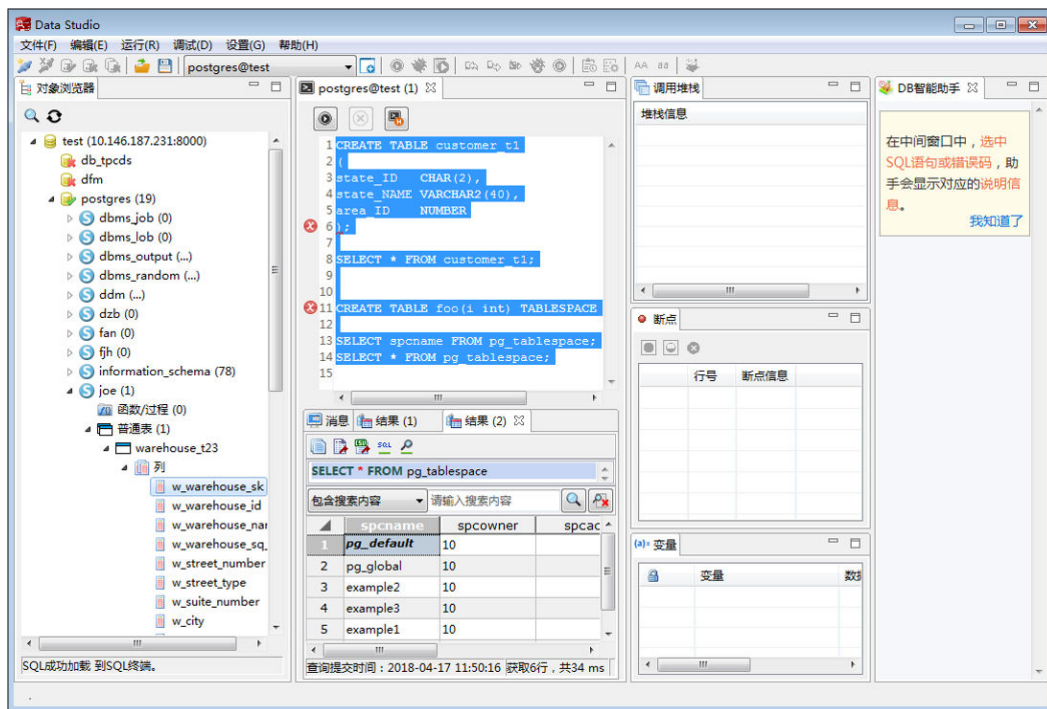


表 4-1 Data Studio 功能

序号	功能清单	描述
1	数据库对象管理	管理数据库、Schema、表、列、约束、索引、视图、索引、表空间、用户角色等数据库对象。
2	SQL编辑器	创建、编辑、运行、调试PL/SQL存储过程，查询语句格式化，SQL语句自动建议和模板化填充。
3	存储过程调试	创建调试连接，使用断点控制PL/SQL存储过程的执行，展示调用栈、变量等调试信息。
4	控制台	查看执行计划及成本，支持标准输入、输出。
5	查询结果窗	显示、复制、导出、编辑、搜索查询结果。

Data Studio的具体使用方法请参见[Data Studio](#)。

### 4.1.3 Database Schema Converter

#### 技术背景

当客户选择切换到华为数据库后可能会面临数据库的迁移任务，数据库迁移包括用户数据迁移和应用程序sql脚本迁移。

其中，应用程序sql脚本迁移是一个复杂、高风险且耗时的过程。

## 功能描述

Database Schema Convertor是一款运行在Linux或Windows操作系统上的命令行工具，致力于向客户提供简单、快速、可靠的应用程序sql脚本迁移服务，通过内置的语法迁移逻辑解析源数据库应用程序sql脚本，并迁移为适用于DWS数据库的应用程序sql脚本。

Database Schema Convertor不需要连接数据库，可在离线模式下实现零停机迁移，迁移过程中会显示迁移过程状态，并用日志记录操作过程中发生的错误，便于快速定位问题。

Database Schema Convertor支持迁移Teradata和Oracle至DWS，目前支持迁移的对象有：

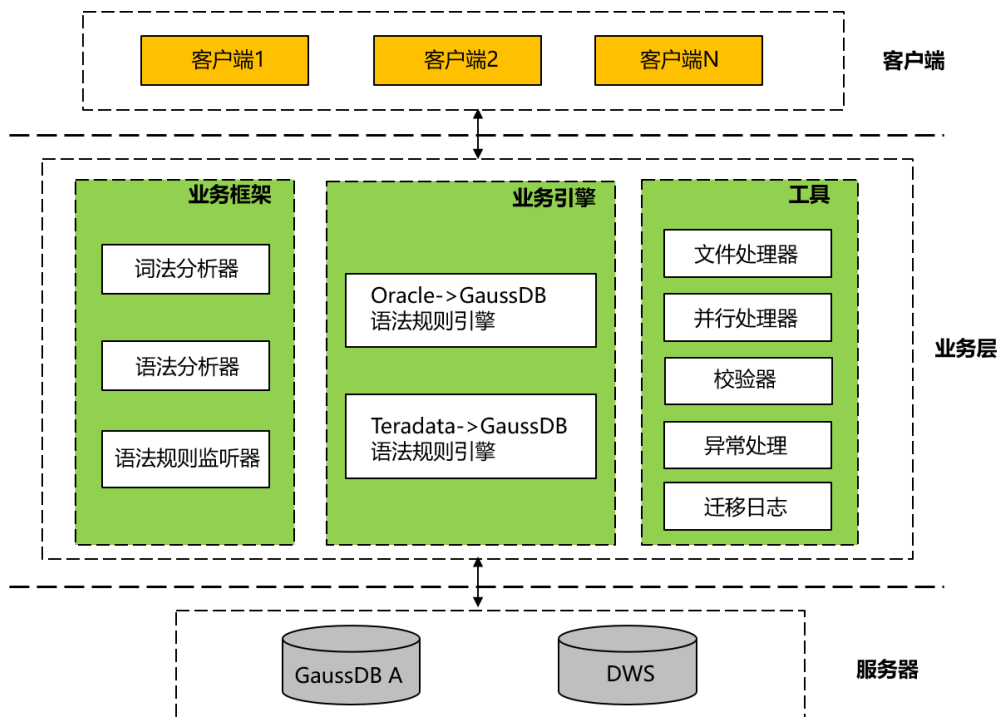
- Oracle和Teradata支持的通用对象：SQL模式，SQL查询
- 仅Oracle支持的对象：PL/SQL
- 仅Teradata支持的对象：包含BTEQ和SQL\_LANG脚本的Perl文件

目标用户群：

- 数据库管理员
- 数据库迁移工程师

## 技术原理

图 4-3 Database Schema Convertor 架构图



Database Schema Convertor支持语法迁移的源端及目标端数据库类型如下：

序号	功能清单	描述
----	------	----

1	Teradata Perl迁移	Database Schema Convertor支持将Teradata Perl文件迁移到GaussDB(DWS)、GaussDB 300和DWS
2	Teradata SQL迁移	Database Schema Convertor支持将Teradata SQL迁移到GaussDB(DWS)、GaussDB 300和DWS
3	Oracle SQL迁移	Database Schema Convertor支持将Oracle SQL迁移到GaussDB(DWS)、GaussDB 300和DWS
4	Oracle(beta) SQL迁移	Database Schema Convertor支持将Oracle SQL迁移到GaussDB(DWS)、GaussDB 300和DWS

Database Schema Convertor的具体使用方法请参见《[DSC工具指南](#)》。

## 4.1.4 数据管理服务(Data Admin Service, 简称 DAS)

### 功能描述

数据管理服务（Data Admin Service, 简称DAS），用来登录和操作云上数据库的Web服务，提供数据库开发、运维、智能诊断的一站式云上数据库管理平台，方便用户使用和运维华为云数据库。提供最好用的数据库客户端：无需安装本地客户端，所见即所得的可视化操作体验，提供数据和表结构的同步、在线编辑，SQL输入的智能提示等丰富的数据库开发功能。

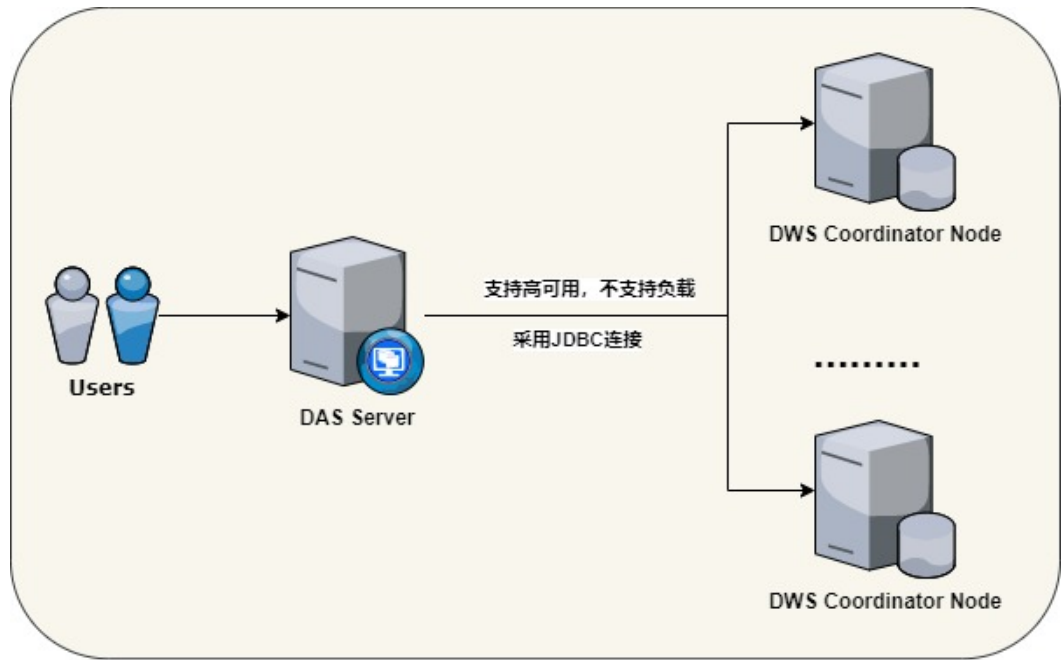
**版本约束：**DWS版本  $\geq 8.0.1$ ，且Agent版本  $\geq 8.1.3.101$ 。

DAS主要为DWS开发人员提供以下功能：

- 浏览及管理数据库对象（例如：数据库、表、视图、存储过程、序列、触发器）。
- 编辑和执行SQL语句。
- SQL执行记录管理。

## 技术原理

图 4-4 DAS 组件与 DWS 交互图



如上图所示，DAS使用B/S结构，通过JDBC驱动与DWS数据库进行通信。

图 4-5 DWS 端入口界面展示

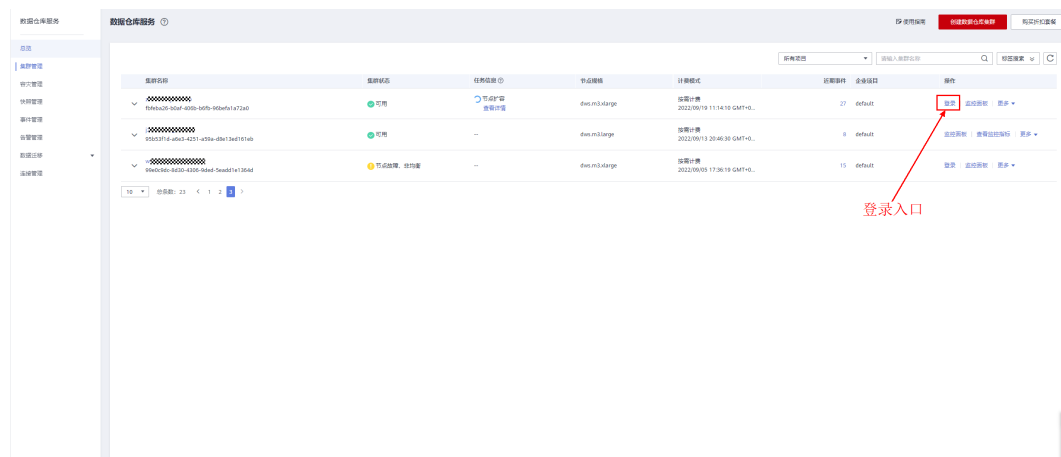


图 4-6 DAS 端口界面展示

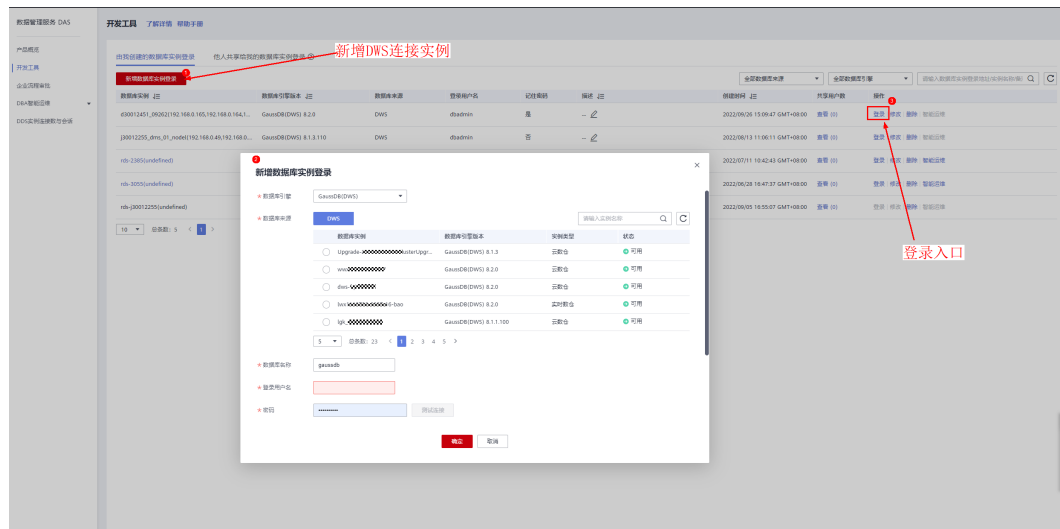


图 4-7 DAS-SQL 执行界面展示

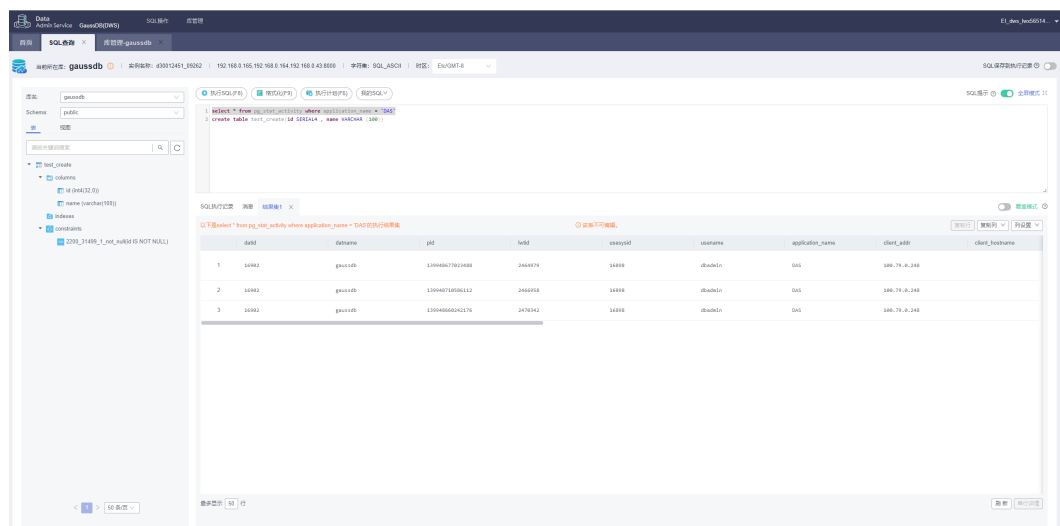


图 4-8 DAS-数据库对象界面展示

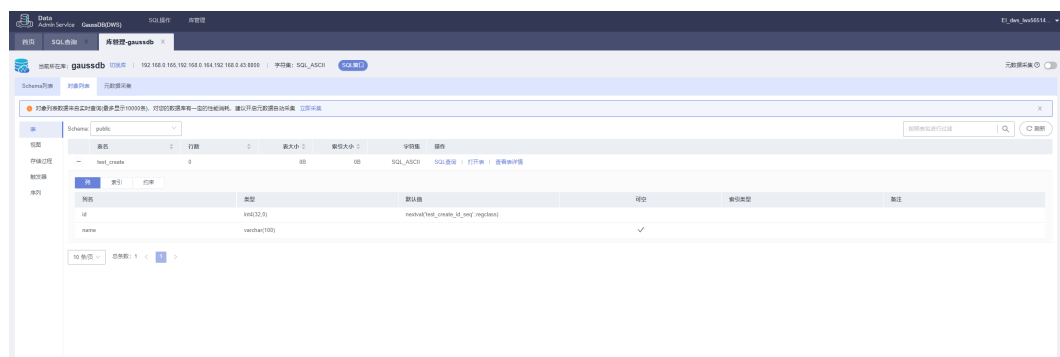


表 4-2 功能描述

序号	功能清单	描述
1	数据库对象管理	管理数据库、Schema、表、列、约束、索引、视图、存储过程、触发器、序列等数据库对象。
2	存储过程/函数新增、执行	创建及执行存储过程/函数
3	SQL编辑器	SQL语句执行、格式化、执行计划，SQL提示，常用SQL管理。
4	SQL执行记录管理	非敏感SQL执行记录管理。

DAS的具体使用方法请参见[使用DAS连接集群](#)。

## 4.2 数据库监控工具

### 技术背景

数据库作为IT基础设施，用户对数据库的要求是稳定、高吞吐、低延时。为了实现上述目标，确保数据库集群能够持续为用户提供稳定快速的服务。我们需要完善的数据库监控工具来帮助数据库运维人员，实时监控数据库运行的方方面面，做到异常问题早发现，快定位，及时处理。

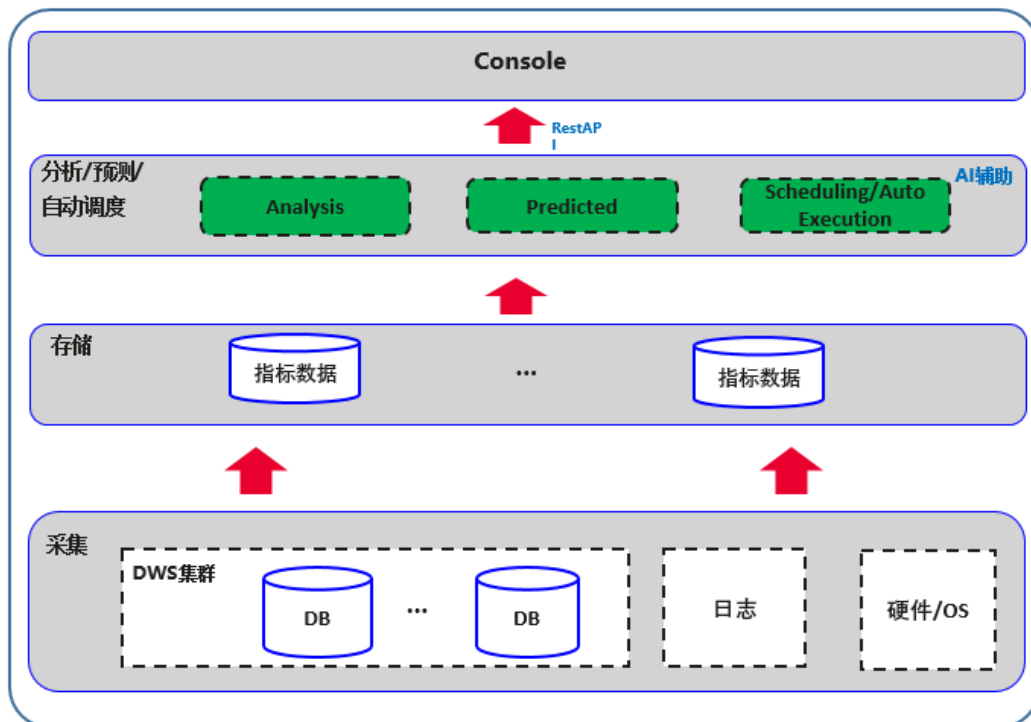
DMS ( Database Monitor Service ) 作为GaussDB(DWS)原生的数据库监控服务，可以提供全面的数据库资源消耗指标与数据库业务执行指标。DMS以可视化工具的形式监控数据库集群的实时与历史运行状态，帮助用户发现问题，定位问题，解决问题。

### 数据库监控原理

DMS采用三层结构实现GaussDB(DWS)数据库集群的监控：

- 采集 ( dms-agent )：agent嵌入数据库集群的每个节点，采集集群运行状态，收集集群运行的原始数据并上报。
- 存储 ( dms-collection )：接收agent上报的数据，并存入指标数据库。
- 分析 ( dms-monitoring )：利用监控数据库中的大量数据，并通过聚合计算将监控视图展现给用户。

图 4-9 DMS 架构



## 数据库监控指标

数据库监控指标参见表4-3，了解更多请参见[数据库监控](#)。

表 4-3 GaussDB(DWS)监控列表

监控对象	指标名称	含义	取值范围	监控周期 (原始指标)
集群概览	集群状态	集群的状态信息。	正常/异常/降级	30s
	节点数量	集群中可用节点的数量和节点的总数（可用/总数）。	≥ 0	60s
	CN数量	集群中CN节点的数量。	≥ 0	60s
	数据库数量	集群中创建的数据库数量。	≥ 0	90s
资源消耗	CPU使用率	集群中所有节点的实时平均CPU使用率。	0% ~ 100%	30s
	内存使用率	集群中所有节点的实时平均内存使用率。	0% ~ 100%	30s
	磁盘使用率	集群中所有节点的实时平均磁盘使用率。	0% ~ 100%	30s



监控对象	指标名称	含义	取值范围	监控周期 (原始指标)
	磁盘I/O	集群中所有磁盘的实时平均磁盘I/O。	≥ 0KB/s	30s
	网络I/O	集群中所有网卡的实时平均网络I/O。	≥ 0KB/s	30s
Top 5 耗时查询	查询ID	用户查询的ID号码, 由数据库自动生成。	≥ 0	180s
	SQL语句	用户执行的查询语句。	字符串	180s
	执行时间	用户查询语句的执行时间, 单位: ms。	≥ 0ms	180s
Top 5 下盘量查询	查询ID	用户查询的ID号码, 由数据库自动生成。	≥ 0	180s
	SQL语句	用户执行的查询语句。	字符串	180s
	下盘量	用户执行查询语句的下盘量, 单位: MB。	≥ 0MB	180s
集群资源消耗指标	CPU使用率	集群中所有节点的平均CPU使用率和倾斜率, 倾斜率计算公式: $(\max-\text{avg})/\max$ 。	0% ~ 100%	30s
	内存使用率	集群中所有节点的平均内存使用率和倾斜率, 倾斜率计算公式: $(\max-\text{avg})/\max$ 。	0% ~ 100%	30s
	磁盘使用率	集群中所有磁盘的平均使用率和倾斜率, 倾斜率计算公式: $(\max-\text{avg})/\max$ 。	0% ~ 100%	30s
	磁盘I/O使用率	集群中所有磁盘的平均I/O使用率和倾斜率, 倾斜率计算公式: $(\max-\text{avg})/\max$ 。	0% ~ 100%	30s
	网络I/O使用率	集群中所有网卡的平均I/O使用率和倾斜率, 倾斜率计算公式: $(\max-\text{avg})/\max$ 。	0% ~ 100%	30s
数据库主要指标	集群状态	集群运行状态。	正常/降级/异常	30s
	集群CN异常数量	集群中处于异常状态的CN的数量。	≥ 0	60s
	集群只读	集群是否处于只读状态。	是/否	30s
	并发会话数	集群中单位时间内并发的会话数量。	≥ 0	30s

监控对象	指标名称	含义	取值范围	监控周期 (原始指标)
	并发查询数	集群中单位时间内并发查询语句的数量。	≥ 0	30s
节点监控-概览	节点名称	集群中节点的名称。	字符串	30s
	CPU使用率	主机的CPU使用率。	0%~100%	30s
	内存使用率	主机的内存使用率。	0%~100%	30s
	平均磁盘使用率	主机磁盘使用率。	0%~100%	30s
	IP地址	主机的业务IP地址。	字符串	30s
	磁盘I/O	主机的磁盘I/O, 单位: KB/s。	≥ 0KB/s	30s
	TCP协议栈重传率	单位时间内TCP报的重发率。	0%~100%	30s
	状态	主机的运行状态。	在线/离线	30s
节点监控-磁盘	节点名称	集群中的节点名称。	字符串	30s
	磁盘名	主机上磁盘名称。	字符串	30s
	磁盘容量	主机上磁盘的容量, 单位: GB。	≥ 0GB	30s
	磁盘使用率	主机上磁盘的使用率。	0%~100%	30s
	磁盘读速率	主机上磁盘读速率, 单位: KB/s。	≥ 0KB/S	30s
	磁盘写速率	主机上磁盘写速率, 单位: KB/s。	≥ 0KB/S	30s
	I/O等待时间-await	平均每次I/O请求的等待时间, 单位: ms。	≥ 0ms	30s
	I/O服务时间-svctm	平均每次I/O请求的处理时间, 单位: ms。	≥ 0ms	30s
	I/O使用率-util	主机上I/O的使用率。	0%~100%	30s
节点监控-网络	节点名称	集群中的节点名称。	字符串	30s
	网卡名称	主机上的网卡名称。	字符串	30s
	网卡状态	网卡状态。	up/down	30s

监控对象	指标名称	含义	取值范围	监控周期 (原始指标)
	网卡速度	网卡工作速率, 单位: Mbps。	$\geq 0$	30s
	接收包数	网卡的接收包数。	$\geq 0$	30s
	发送包数	网卡的发送包数。	$\geq 0$	30s
	接收丢包数	网卡的接收丢包数。	$\geq 0$	30s
	接收速率	网卡单位时间内接收到的字节数, 单位: KB/s。	$\geq 0$ KB/s	30s
	发送速率	网卡单位时间内发送出的字节数, 单位: KB/s。	$\geq 0$ KB/s	30s
数据库监控	数据库名称	集群中用户创建的数据库名称。	字符串	60s
	使用容量	当前数据库的已使用容量, 单位: GB。	$\geq 0$ GB	86400s
	用户数量	当前数据库上的用户数量。	$\geq 0$	30s
	会话数量	当前数据库上的会话数量。	$\geq 0$	30s
	应用数量	当前数据库上的应用数量。	$\geq 0$	30s
	查询数量	当前数据库上的活动查询数量。	$\geq 0$	30s
	全表扫描返回行数	当前数据库上全表扫描查询的返回行数。	$\geq 0$	60s
	索引查询返回行数	当前数据库上使用索引查询的返回行数。	$\geq 0$	60s
	插入行数	当前数据库上的插入行数。	$\geq 0$	60s
	更新行数	当前数据库上的更新行数。	$\geq 0$	60s
	删除行数	当前数据库上的删除行数。	$\geq 0$	60s
	事务执行数	当前数据库上事务的执行次数。	$\geq 0$	60s
	事务回滚数	当前数据库中已经回滚的事务数。	$\geq 0$	60s
	死锁数	当前数据库中检索到的死锁数。	$\geq 0$	60s
临时文件数量	当前数据库创建的临时文件数量。	$\geq 0$	60s	

监控对象	指标名称	含义	取值范围	监控周期 (原始指标)
	临时文件容量	当前数据库写入临时文件的大小, 单位: GB。	≥ 0	60s
性能监控	集群CPU使用率	集群中所有节点的平均CPU使用率和倾斜率的历史趋势, 倾斜率计算公式: $(\max-\text{avg})/\max$ 。	0% ~ 100%	30s
	集群内存使用率	集群中所有节点的平均内存使用率和倾斜率的历史趋势, 倾斜率计算公式: $(\max-\text{avg})/\max$ 。	0% ~ 100%	30s
	集群磁盘使用率	集群中所有节点的平均磁盘使用率和倾斜率的历史趋势, 倾斜率计算公式: $(\max-\text{avg})/\max$ 。	0% ~ 100%	30s
	集群磁盘I/O	集群中所有磁盘的平均磁盘I/O数值和倾斜率的历史趋势, 倾斜率计算公式: $(\max-\text{avg})/\max$ 。	0% ~ 100%	30s
	集群网络I/O	集群中所有网卡的平均网络I/O数值和倾斜率的历史趋势, 倾斜率计算公式: $(\max-\text{avg})/\max$ 。	0% ~ 100%	30s
	集群状态	集群的状态信息的历史趋势。	正常/异常/降级	30s
	集群只读	集群只读状态变化趋势的历史趋势。	是/否	30s
	集群CN异常数量	集群中CN异常数量变化的历史趋势。	≥ 0	60s
	集群DN异常数量	集群中DN异常数量变化的历史趋势。	≥ 0	60s
	集群DN实例CPU使用率	集群中所有DN的平均CPU使用率和倾斜率变化的历史趋势, 倾斜率计算公式: $(\max-\text{avg})/\max$ 。	0% ~ 100%	60s
	集群会话数量	集群中会话数量变化的历史趋势。	≥ 0	30s
集群查询数量	集群中查询数量变化的历史趋势。	≥ 0	30s	

监控对象	指标名称	含义	取值范围	监控周期 (原始指标)
	集群死锁数量	集群中死锁数量变化的历史趋势。	$\geq 0$	60s
	集群平均每秒事务数	集群所有数据库每秒钟平均事务数。计算公式： (delta_xact_commit + delta_xact_rollback) / current_collect_rate。	$\geq 0$	60s
	集群平均每秒查询数	集群所有数据库每秒钟平均并发数。计算公式： delta_query_count / current_collect_rate。	$\geq 0$	60s
	数据库会话数量	集群中单个数据库上会话数量变化的历史趋势。	$\geq 0$	30s
	数据库查询数量	集群中单个数据库上查询数量变化的历史趋势。	$\geq 0$	30s
	数据库插入行数	集群中单个数据库上插入行数变化的历史趋势。	$\geq 0$	60s
	数据库更新行数	集群中单个数据库上更新行数变化的历史趋势。	$\geq 0$	60s
	数据库删除行数	集群中单个数据库上删除行数变化的历史趋势。	$\geq 0$	60s
	数据库容量	集群中单个数据库上数据库容量变化的历史趋势。	$\geq 0$	86400s
实时会话	会话ID	当前会话的会话ID（查询线程ID）。	字符串	30s
	用户名称	执行当前会话的用户名。	字符串	30s
	数据库名称	当前会话连接的数据库名称。	字符串	30s
	会话用时	当前会话的存在时间，单位： ms。	$\geq 0$ ms	30s
	应用名称	创建当前会话的应用名称。	字符串	30s
	查询数量	当前会话执行过的SQL语句的数量。	$\geq 0$	30s
	最近查询用时	当前会话中前一个SQL语句的执行用时。	$\geq 0$ ms	30s
	客户端IP	当前会话的发起客户端IP地址。	字符串	30s
	接入CN	当前会话的连接CN。	字符串	30s

监控对象	指标名称	含义	取值范围	监控周期 (原始指标)
	会话状态	当前会话的执行状态。	运行/空闲/重试	30s
实时查询	查询ID	当前查询语句的QueryID, 内核给每一个查询语句分配的唯一标识。	字符串	30s
	用户名称	提交当前查询语句的用户名称。	字符串	30s
	数据库名称	当前查询语句所对应的数据库名称。	字符串	30s
	应用名称	当前查询语句所对应的应用名称。	字符串	30s
	工作负载队列	承载当前查询语句的工作负载队列名称。	字符串	30s
	提交时间	当前查询语句提交的时间戳。	字符串	30s
	阻塞时间	当前查询语句执行前的等待时间, 单位: ms。	≥ 0	30s
	执行时间	当前查询语句到目前为止的执行时间, 单位: ms。	≥ 0	30s
	CPU时间	当前查询语句在所有DN上执行的CPU总时间, 单位: ms。	≥ 0	30s
	CPU时间倾斜	当前查询语句在所有DN上执行的CPU时间倾斜率。	0% ~ 100%	30s
	查询语句	当前正在执行的查询语句。	字符串	30s
	接入CN	提交当前查询语句的CN名称。	字符串	30s
	客户端IP	提交当前查询语句的客户端IP地址。	字符串	30s
	快慢车道	当前查询语句所在的车道。	快车道/慢车道	30s
	查询状态	当前正在执行查询的语句的查询状态。	字符串	30s
会话ID	当前查询语句的SessionID, 内核给每一个客户端连接分配的唯一标识。	字符串	30s	
排队状态	当前查询在数据库中执行的状态, 是否在工作负载队列上排队。	是/否	30s	

监控对象	指标名称	含义	取值范围	监控周期 (原始指标)
历史查询	查询ID	查询语句的QueryID, 内核给每一个查询语句分配的唯一标识。	字符串	180s
	用户名称	提交查询语句的用户名称。	字符串	180s
	应用名称	查询语句对应的应用名称。	字符串	180s
	数据库名称	查询语句所对应的数据库名称。	字符串	180s
	工作负载队列	承载当前查询语句的工作负载队列名称。	字符串	180s
	提交时间	查询语句提交的时间戳。	字符串	180s
	阻塞时间	查询语句执行前的等待时间, 单位: ms。	≥ 0	180s
	执行时间	查询语句到目前为止的执行时间, 单位: ms。	≥ 0	180s
	CPU时间	查询语句在所有DN上执行的CPU总时间, 单位: ms。	≥ 0	180s
	CPU时间倾斜	查询语句在所有DN上执行的CPU时间倾斜率。	0%~100%	180s
	查询语句	执行的查询语句。	字符串	180s
慢实例监控	慢实例数量	当前时间点上检测出慢实例的数量。	≥ 0	240s
	检测时间	慢实例首次检测出来的时间。	字符串	240s
	节点名称	慢实例部署的节点名称。	字符串	240s
	实例名称	慢实例的名称。	字符串	240s
	慢节点检测次数 (24小时内)	慢实例在24小时内被检测出的次数。	≥ 0	240s
工作负载队列监控	工作负载队列	集群中工作负载队列的名称。	字符串	120s
	CPU使用率	工作负载队列的实时CPU使用率。	0%~100%	120s
	CPU资源	工作负载队列的CPU使用率配额。	0%~100%	120s
	实时短查询并发数	工作负载队列的实时简单并发数。	≥ 0	120s

监控对象	指标名称	含义	取值范围	监控周期 (原始指标)
	短查询并发	工作负载队列的简单并发数配额。	$\geq 0$	120s
	实时查询并发	工作负载队列的实时复杂并发数。	$\geq 0$	120s
	查询并发	工作负载队列的复杂并发数配额。	$\geq 0$	120s
	存储资源	工作负载队列的存储资源配额。	$\geq 0$	120s
	磁盘使用率	工作负载队列的磁盘使用率。	0% ~ 100%	120s
	内存资源	工作负载队列的内存配额。	$\geq 0$	120s
	内存使用率	工作负载队列的内存使用率	0% ~ 100%	120s
工作负载队列上的等待查询	用户名	处于等待状态查询的用户名称。	字符串	120s
	应用名称	处于等待状态查询的应用名称。	字符串	120s
	数据库	处于等待状态查询的数据库名称。	字符串	120s
	排队状态	查询在数据库中执行的状态 (CCN/CN/DN)。	字符串	120s
	等待时间	处于等待状态查询的等待时间长度, 单位: ms。	$\geq 0$ ms	120s
	所属工作负载队列	处于等待状态查询所属的工作负载队列。	字符串	120s
	查询语句	处于等待状态查询语句的内容。	字符串	120s
熔断查询	查询ID	熔断查询语句的QueryID。	字符串	120s
	查询语句	熔断状态查询语句的内容。	字符串	120s
	阻塞时间	查询语句触发熔断之前的阻塞时间, 单位: ms。	$\geq 0$	120s
	执行时间	查询语句触发熔断之前的执行时间, 单位: ms。	$\geq 0$	120s
	CPU时间	查询语句触发熔断之前各个DN消耗的平均CPU时间, 单位: ms。	$\geq 0$	120s



监控对象	指标名称	含义	取值范围	监控周期 (原始指标)
	CPU倾斜率	查询语句触发熔断之前各个DN间消耗CPU时间的倾斜率。	0%~100%	120s
	异常处理方式	查询语句触发熔断后的处理方式。	取消/降级	120s
	处理状态	查询语句熔断处理状态。	执行中/已完成	120s
SQL诊断	查询ID	当前查询的查询ID (查询逻辑ID)。	字符串	180s
	数据库	当前查询执行的数据库名称。	字符串	180s
	模式名	当前查询的模式名称。	字符串	180s
	用户名称	当前查询的用户名称。	字符串	180s
	客户端	当前查询的发起客户端名称。	字符串	180s
	客户端IP地址	当前查询的发起客户端IP地址。	字符串	180s
	运行时间	当前查询的执行时间, 单位: ms。	≥ 0	180s
	CPU时间	当前查询执行的CPU时间, 单位: ms。	≥ 0	180s
	开始时间	当前查询的开始时间。	时间戳	180s
	完成时间	当前查询的结束时间。	时间戳	180s
	详情	当前查询的详细介绍。	字符串	180s
INODE	INODE使用率	磁盘inode使用率。	0%~100%	30s
SCHEMA	SCHEMA使用率	数据库schema使用率。	0%~100%	3600s

### 4.3 三方工具兼容

为了让用户使用数据库更便捷, DWS一方面在努力自主研发工具, 另一方面也在逐步构建与国内外业界主流数据库工具的对接。数据库外围工具种类丰富、标准繁多, 目前DWS兼容的工具类别主要包括: BI工具、ETL工具、报表工具、数据挖掘、数据复制、数据集成、备份恢复、安全、运维监控、云平台、Web中间件等, 详细工具清单参照GaussDB生态对接相关文档。

# 5 DWS 对外接口

DWS各组件提供的对外接口说明如表5-1所示。

表 5-1 组件对外接口

组件名	支持的接口类型
DWS	JDBC、ODBC
DWS管控面	REST、SNMP