

openGauss  
3.0.0

# 产品描述

文档版本	01
发布日期	2022-03-31



版权所有 © 华为技术有限公司 2022。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。

## 华为技术有限公司

地址： 深圳市龙岗区坂田华为总部办公楼 邮编： 518129

网址： <https://www.huawei.com>

客户服务邮箱： [support@huawei.com](mailto:support@huawei.com)

客户服务电话： 4008302118

# 目 录

1 产品定位.....	1
2 系统架构.....	2
3 应用场景.....	4
4 产品特点.....	5
5 运行环境.....	6
6 技术指标.....	7
7 基本功能和特性.....	8
8 企业级增强特性.....	9

# 1 产品定位

---

openGauss是一款支持SQL2003标准语法，支持主备部署的高可用关系型数据库。

- 多种存储模式支持复合业务场景，新引入提供原地更新存储引擎。
- NUMA化数据结构支持高性能。
- Paxos一致性日志复制协议，主备模式，CRC校验支持高可用。
- 支持全密态计算，账本数据库等安全特性，提供全方位端到端的数据安全保护。
- 通过Table Access Method接口层支持多存储引擎。

openGauss是一款提供面向多核的极致性能、全链路的业务和数据安全、基于AI的调优和高效运维的能力，全面友好开放，携手伙伴共同打造全球领先的企业级开源关系型数据库，采用木兰宽松许可证v2发行。openGauss深度融合华为在数据库领域多年的研发经验，结合企业级场景需求，持续构建竞争力特性。

# 2 系统架构

openGauss是单机系统，在这样的系统架构中，业务数据存储在单个物理节点上，数据访问任务被推送到服务节点执行，通过服务器的高并发，实现对数据处理的快速响应。同时通过日志复制可以把数据复制到备机，提供数据的高可靠和读扩展。

## 软件架构

openGauss是单机系统，支持主备部署。

openGauss逻辑架构如图2-1所示。

图 2-1 openGauss 逻辑架构图

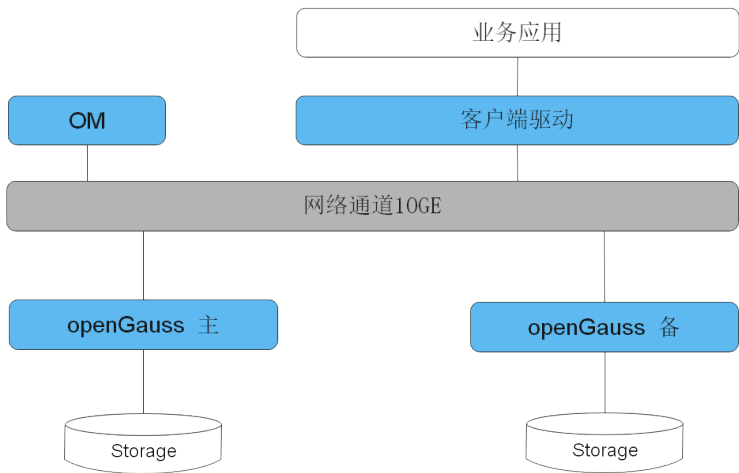


表 2-1 架构说明

名称	描述
OM	运维管理模块（Operation Manager）。提供数据库日常运维、配置管理的管理接口、工具。

名称	描述
CM	数据库管理模块（Cluster Manager）。管理和监控数据库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。
客户端驱动	客户端驱动（Client Driver）。负责接收来自应用的访问请求，并向应用返回执行结果。客户端驱动负责与openGauss实例通信，发送应用的SQL命令，接收openGauss实例的执行结果。
openGauss（主备）	openGauss主备（Datanode）。负责存储业务数据、执行数据查询任务以及向客户端返回执行结果。 openGauss实例包含主、备两种类型，支持一主多备。建议将主、备openGauss实例分散部署在不同的物理节点中。
Storage	服务器的本地存储资源，持久化存储数据。

# 3 应用场景

---

- 交易型应用  
大并发、大数据量、以联机事务处理为主的交易型应用，如电商、金融、O2O、电信CRM/计费等，应用可按需选择不同的主备部署模式。
- 物联网数据  
在工业监控和远程控制、智慧城市的延展、智能家居、车联网等物联网场景下，传感监控设备多，采样率高，数据存储为追加模型，操作和分析并重的场景。

# 4 产品特点

openGauss具有高性能、高可用、高安全性和可维护性好的特点。

- 高性能  
通过列存储、向量化执行引擎、融合引擎等关键技术，实现百亿数据量查询秒级响应。
- 高可用  
同城跨AZ（ Available Zone ）容灾，数据不丢失，分钟级恢复。
- 高安全性  
支持访问控制、加密认证、数据库审计、动态数据脱敏等安全特性，提供全方位端到端的数据安全保护。
- 高可靠
  - 闪回和回收站  
通常在表数据被错误的UPDATE、DELETE、TRUNCATE和DROP时数据难以恢复，即便恢复也仅能通过PITR（ Point-in-time recovery，基于时间点恢复 ）的方式恢复到错误操作前的时刻。这种修复方式会导致整个数据库不可用，并且一些用户不希望“撤销”的表操作也同样会被“撤销”。openGauss对TIMECAPSULE以及RECYCLEBIN的支持，使用户可以通过指定时间戳进行闪回查询，或者指定时间戳对表进行闪回，获取到错误DELETE、UPDATE前的历史数据。通过从RECYCLEBIN中闪回TRUNCATE、以及DROP的表对象，用户可以将表数据恢复到错误操作前，大大提高了用户数据的可靠性。
- 可维护性好  
支持WDR诊断、慢SQL诊断、Session诊断等多种维护手段，准确快速定位问题。具备AI4DB能力，能够通过AI算法实现数据库自调优、自监控、自诊断等。



# 5 运行环境

---

## 支持的硬件平台

openGauss支持运行在鲲鹏服务器和通用的x86服务器上：

- 支持鲲鹏服务器和基于x86\_64的通用PC服务器。
- 支持本地存储（SATA、SAS、SSD）。
- 支持千兆、万兆Ethernet网络。

## 支持的操作系统

- ARM：
  - openEuler 20.03LTS（推荐采用此操作系统）
  - 麒麟V10
- X86：
  - openEuler 20.03LTS
  - CentOS 7.6

# 6 技术指标

openGauss的技术指标如表6-1所示。

表 6-1 技术指标

技术指标	最大值
数据库容量	受限于操作系统与硬件
单表大小	32TB
单行数据大小	1GB
每条记录单个字段的大小	1GB
单表记录数	$2^{48}$
单表列数	250~1600（随字段类型不同会有变化）
单表中的索引个数	无限制
复合索引包含列数	32
单表约束个数	无限制
并发连接数	10000
分区表的分区个数	32768（范围分区）/64（哈希分区/列表分区）
分区表的单个分区大小	32TB
分区表的单个分区记录数	$2^{55}$

# 7 基本功能和特性

## 背景信息

openGauss是一个单机数据库，具备关系型数据库的基本功能，以及企业特性的增强功能。

## 基本功能

- 标准SQL支持  
支持标准的SQL92/SQL99/SQL2003/SQL2011规范，支持GBK和UTF-8字符集，支持SQL标准函数与分析函数，支持存储过程。
- 数据库存储管理功能  
支持表空间，可以把不同表规划到不同的存储位置。
- 提供主备双机  
事务支持ACID特性、单节点故障恢复、双机数据同步，双机故障切换等。
- 应用程序接口  
支持标准JDBC 4.0的特性、ODBC 3.5特性。
- 管理工具  
提供安装部署工具、实例启停工具、备份恢复工具。
- 安全管理  
支持SSL安全网络连接、用户权限管理、密码管理、安全审计等功能，保证数据库在管理层、应用层、系统层和网络层的安全性。

# 8 企业级增强特性

## 数据分区

数据分区是数据库产品普遍具备的功能。在openGauss中，数据分区是对数据按照用户指定的策略对数据做的水平分表，将表按照指定范围划分为多个数据互不重叠的部分（Partition）。

openGauss支持：

- 范围分区（Range Partitioning）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。
- 列表分区（List Partitioning）功能，即根据表的一列，将要插入表的记录中出现的键值分为若干个列表（这些列表在不同的分区里没有重叠），然后为每个列表创建一个分区，用来存储相应的数据。
- 哈希分区（Hash Partitioning）功能，即根据表的一列，通过内部哈希算法将要插入表的记录划分到对应的分区中。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。

例如，表8-1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 8-1 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openGauss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，目前不支持列表分区和哈希分区。

## 向量化执行和行列混合引擎

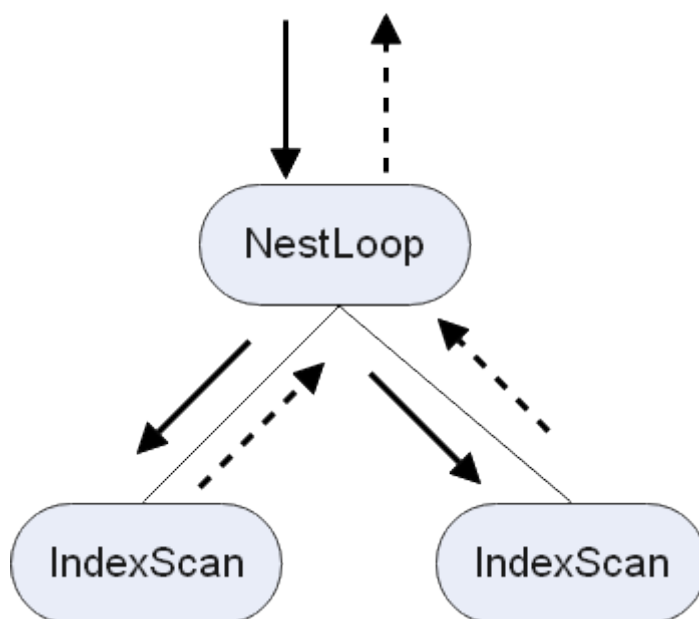
在大宽表，数据量比较大、查询经常关注某些列的场景中，行存储引擎查询性能比较差。例如气象局的场景，单表有200~800个列，查询经常访问10个列，在类似这样的场景下，向量化执行技术和列存储引擎可以极大的提升性能和减少存储空间。

- 向量化执行

标准的迭代器模型如图8-1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 8-1 向量化执行引擎



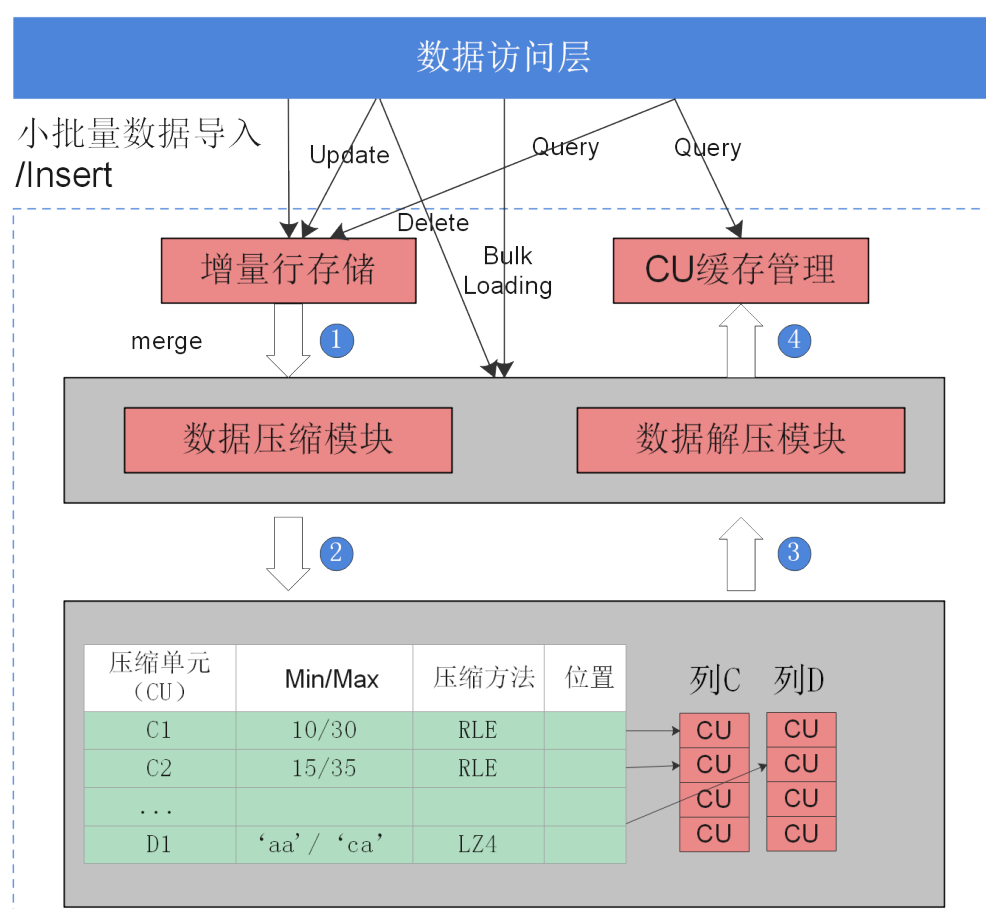
- 行列混合存储引擎

openGauss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，OLAP类业务场景（范围统计类查询和批量导入操作频繁，更新、删除、点查和点插操作不频繁，表的字段比较多，即大宽表，查询中涉及到的列不是很多）下，适合列存储，OLTP类业务场景（点查、点插、删除、更新频繁，范围统计类查询和批量导入操作不频繁，表的字段个数比较少，查询大部分字段）下，适合行存储。

如图8-2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 8-2 行列混存引擎



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。

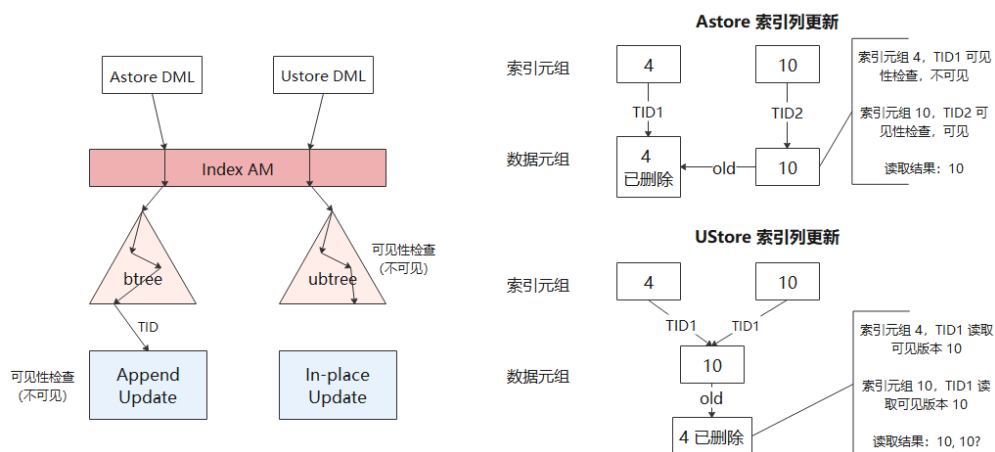
- 不支持触发器，不支持主外键。
- 支持Psort index、B-tree index和GIN index，具体约束参见《开发者指南》中“SQL参考 > SQL语法 > CREATE INDEX”章节。
- 列存下的数据压缩  
对于非活跃的早期数据可以通过压缩来减少空间占用，降低采购和运维成本。  
openGauss列存储压缩支持Delta Value Encoding、Dictionary、RLE、LZ4、ZLIB等压缩算法，且能够根据数据特征自适应的选择压缩算法，平均压缩比7:1。  
压缩数据可直接访问，对业务透明，极大缩短历史数据访问的准备时间。

## 融合存储引擎

融合引擎架构支持了可插拔存储引擎的架构，新增了In-place update存储引擎，索引多版本为索引增加了事务信息，Xlog无锁刷新极大提升了Xlog写入效率，并行Page回放提升了备机回放效率，企业级特性闪回为用户提供了一个稳定的查询状态。

- In-place update存储引擎  
新增的In-place update存储引擎很好的解决了Append update存储引擎空间膨胀和元组较大的问题，高效回滚段的设计是In-place update存储引擎的基础。
- 索引多版本

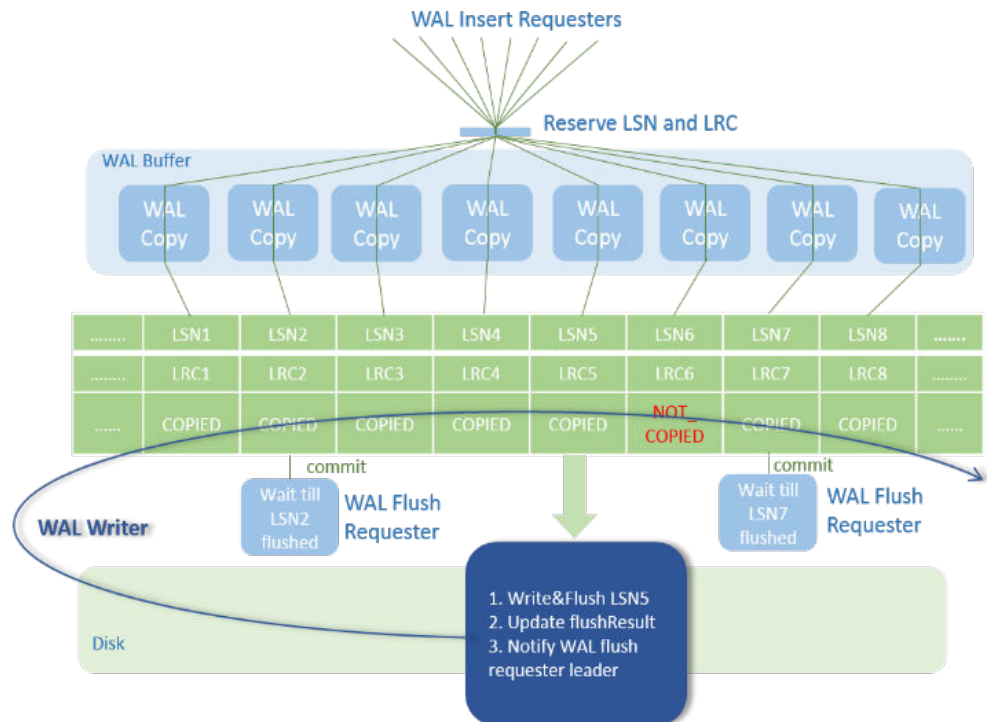
图 8-3 UBTree 与 BTree 查找、更新比较示意图



通过在索引页面元组上维护版本信息，UBtree能够在索引层进行 MVCC 可见性检查。同时 UBtree 也能通过版本信息独立判断索引元组是否已经无效（Dead），进而使得 in-place update引擎能实现数据表以及索引表上页级的空间清理，并在此基础上构建不依赖 AutoVacuum 的独立垃圾回收机制。

- Xlog无锁刷新

图 8-4 Xlog lock less Design



本特性对WalInsertLock进行优化，利用LSN（Log Sequence Number）及LRC（Log Record Count）记录了每个backend的拷贝进度，取消WalInsertLock机制。在backend将日志拷贝至WalBuffer时，不用对WalInsertLock进行争抢，可直接进行日志拷贝操作。并利用专用的WalWriter写日志线程，不需要backend线程自身来保证XLog的Flush。通过以上优化，取消WalInsertLock争抢及WalWriter专用磁盘写入线程，在保持原有XLog功能不变的基础上，可进一步提升系统性能。

- 并行Page回放

本特性针对Ustore Inplace update WAL log写入，Ustore DML operation并行回放分发进行优化。通过利用Prefix和suffix来减少update WAL log的写入。通过把回放线程分多个类型来解决Ustore DML WAL大多都是多页面回放问题。同时把Ustore的数据页面回放按照blkno去分发，提高并行回放的并行程度。

- 企业级特性闪回

闪回是数据库恢复技术的一环，能够使得DBA有选择性的高效撤销一个已提交事务的影响，将数据从人为的不正确的操作中进行恢复。在采用闪回技术之前，只能通过备份恢复、PITR等手段找回已提交的数据库修改，恢复时长需要数分钟甚至数小时。采用闪回技术后，恢复已提交的数据库修改前的数据，只需要秒级，而且恢复时间和数据库大小无关。

本特性支持四种闪回实现：

- 闪回查询：可以查询过去某个时间点表的某个snapshot数据，这一特性可用于查看和逻辑重建意外删除或更改的受损数据。闪回查询基于MVCC多版本机制，通过检索查询旧版本，获取指定老版本数据。
- 闪回表：可以将表恢复至特定时间点，当逻辑损坏仅限于一个或一组表，而不是整个数据库时，此特性可以快速恢复表的数据。闪回表基于MVCC多版本机制，通过删除指定时间点和该时间点之后的增量数据，并找回指定时间点和当前时间点删除的数据，实现表级数据还原。



- 闪回drop：可以恢复意外删除的表，从回收站（recycle bin）中恢复被删除的表及其附属结构如索引、表约束等。闪回drop是基于回收站机制，通过还原回收站中记录的表的物理文件，实现已drop表的恢复。
- 闪回truncate：可以恢复误操作或意外被进行truncate的表，从回收站中恢复被truncate的表及索引的物理数据。闪回truncate基于回收站机制，通过还原回收站中记录的表的物理文件，实现已truncate表的恢复。

## 高可靠事务处理

openGauss提供事务管理功能，保证事务的ACID特性。

为了在主节点出现故障时尽可能地不中断服务，openGauss提供了主备双机高可靠机制。通过保护关键用户程序对外不间断提供服务，把因为硬件、软件和人造成的故障对业务的影响程度降到最低，以保证业务的持续性。

### 故障恢复

支持节点故障可恢复及恢复后满足ACID特性。节点故障、停止后重启等情况下，openGauss能够保证故障之前的数据无丢失，满足ACID特性。

### 事务管理

- 支持事务块，用户可以通过start transaction命令显式启动一个事务块。
- 支持单语句事务，用户不显式启动事务，则单条语句就是一个事务。

## 高并发&高性能

openGauss通过服务器端的线程池，可以支持1W并发连接。通过NUMA化内核数据结构，支持线程亲和性处理，可以支持百万级tpmC。通过页面的高效冷热淘汰，支持T级别大内存缓冲区管理。通过CSN快照，去除快照瓶颈，实现多版本访问，读写互不阻塞。通过增量检查点，避免全页写导致的性能波动，实现业务性能平稳运行。

## SQL 自诊断

通过执行查询对应的explain performance，获得对应执行计划，是一种十分有效的定位查询性能问题的方法。但是这种方法需要修改业务逻辑，同时输出的日志量大，问题定位的效率依赖于人员的经验。SQL自诊断为用户提供了另一种更为高效易用的性能问题定位方法。

在执行作业之前，配置GUC参数resource\_track\_level和resource\_track\_cost，然后运行用户作业，就可以通过查看相关系统视图，获得执行完成的相关查询作业可能存在的性能问题。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《开发者指南》中“性能调优 > SQL调优指南 > 典型SQL调优点 > SQL自诊断”章节，就可以对存在性能问题的作业进行调优。

SQL自诊断可以在不影响用户作业，不修改业务逻辑的情况下，诊断出相对准确的性能问题，为用户提供更为易用的性能调优参考。

## 全密态数据库等值查询

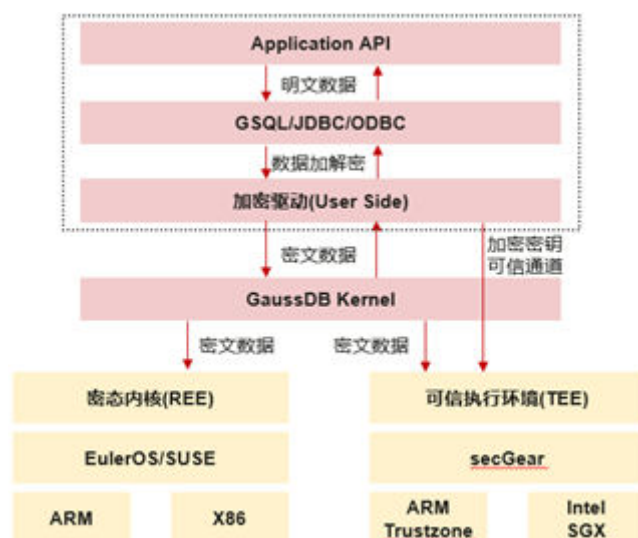
伴随着云基础设施的快速增长和成熟，与之对应的云数据库服务也层出不穷。云数据库俨然已成为数据库业务未来重要的增长点，绝大多数的传统数据库服务厂商正在加速提供更优质的云数据库服务。但无论是传统的线下数据库服务，还是日益增长的云数据库服务，数据库的核心任务都是帮助用户存储和管理数据，在复杂多样的环境下，保证数据不丢失、隐私不泄露、数据不被篡改，同时服务不中断。这就要求数据

库具体多层次的安全防御机制，用来抵抗来自多方面的恶意攻击行为。通过成熟的安全技术手段，构建数据库多层级安全防御体系，保障数据库在应用中的安全。因此，为了更好的保护敏感和隐私数据，特别是针对云数据库服务，急需一种能在服务器端彻底解决数据全生命周期隐私保护的系统性解决方案，该方案被称为密态数据库解决方案。

- 密态数据库总体方案

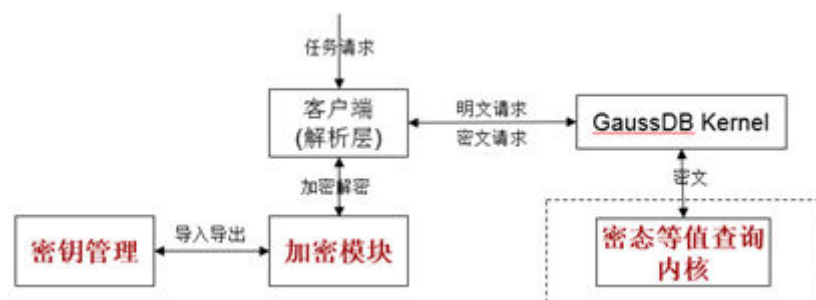
密态等值查询属于密态数据库第一阶段方案，但是遵从密态数据库总体架构。密态数据库的总体架构示意图如图8-5所示。密态数据库的完整形态包括密码学方案和软硬结合方案。

图 8-5 密态数据库总体架构



由于密态等值查询仅涉及到软件部分，仅需集成密态数据库总体架构的软件部分，其总体实现方案如图8-6所示。

图 8-6 密态等值查询总体方案



从总体流程上来看，数据在客户端完成加密，以密文形式发送到openGauss数据库服务侧，即需要在客户端构建加解密模块。加解密模块依赖密钥管理模块，密钥管理模块生成根密钥（RK, Root Key）和客户端主密钥（CMK, Client Master Key）。有了CMK，可以通过SQL语法定义列加密密钥（CEK, Column Encryption Key），CMK由RK加密后保存在密钥存储文件（KSF, Key Store File）中，CMK和RK由KeyTool统一管理；CEK则由CMK加密后存储在服务端(加密算法使用对称加密算法AES256)和国密算法SM2。

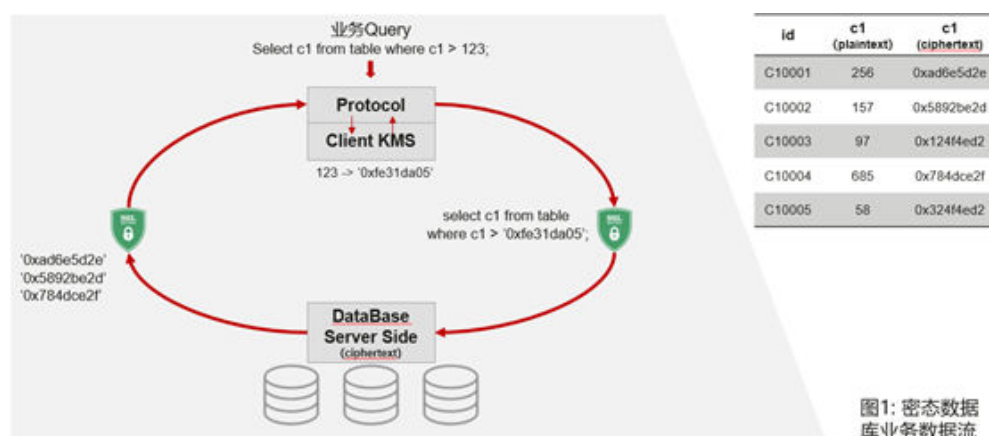
客户端依据生成的CEK来对数据进行加密，数据加密算法主要使用对称加密算法AES算法(包括AES128和AES256)和国密算法SM4。加密后的数据会存放在数据库服务端，经过密文运算后服务端返回密文结果集，并在客户端完成最后的解密，获取最终结果。

用户根据业务需要对数据定义加密属性信息（被加密的列被称之为加密列），对于不需要加密的数据则按照原有明文格式发送至服务端。当查询任务发起后，客户端需要对当前的Query进行解析，如果查询语句中涉及加密列，则对对应的列参数（加密列关联参数）也要进行加密（这里说的加密均需要为确定性加密，否则无法支持对应的查询）；如果查询语句中不涉及加密列，则直接发送至服务端，无需额外的操作。

在数据库服务侧，加密列的数据始终以密文形态存在，整个查询也在密文形态下实现。对于第一阶段密态等值查询解决方案，需要采用确定性加密，使得相同的明文数据获得相同的密文，从而支持等值计算。

- 密态数据库数据流图

图 8-7 密态数据库数据流图



从该数据流图中可以看出，密态数据库允许客户端对客户端应用程序内的敏感数据进行加密。在查询期间，整个业务数据流在数据处理过程中都是以密文形态存在。优势如下：

- 保护数据在云上全生命周期的隐私安全；
- 通过将密钥掌握在用户自己手上，实现公有云、消费者云以及开发用户的用户信任问题；
- 让云数据库借助全密态能力更好的遵守个人隐私保护方面的法律法规。

- 使用场景

混合云场景：数据库客户端和数据库服务端均部署于用户私有网络，客户端采用华为管控界面。

公有云场景：数据库客户端在用户本地，数据库服务端在华为云。

公有云服务：数据库客户端和数据库服务端均在华为云。

## 内存表

内存表把数据全部缓存在内存中，所有数据访问实现免锁并发，实现数据处理的极致性能，满足实时性严苛要求场景。

## 支持多存储引擎

openGauss基于统一的事务机制，统一的日志系统，统一的并发控制系统，统一的元信息信息，统一缓存管理提供Table Access Method接口，支持不同的存储引擎。

目前支持Astore和Ustore存储引擎。

## 主备双机

主备双机支持同步和异步复制，应用可以根据业务场景选择合适的部署方式。同步复制保证数据的高可靠，一般需要一主两备部署，同时对性能有一定影响。异步复制一主一备部署即可，对性能影响小，但异常时可能存在数据丢失。openGauss支持页面损坏的自动修复，在主机页面发生损坏时，能够自动从备机修复损坏页面。

openGauss支持备机并行日志恢复，尽量降低主机故障时业务不可用的时间。

同时，如果按照主备模式部署，并打开备机可读功能后，备机将能够提供读操作，但不支持写操作（如建表、插入数据、删除数据等），从而缓解主机上的压力。

## 具备 AI 能力

- 参数自动调优

在数据库场景中，不同类型的作业任务对于数据库的最优参数数值组合存在偏差。为了获得更好的运行性能，用户希望快速将数据库的参数调整到最优状态。人工调参的学习成本高且不具有实时性和广泛可用。通过机器学习方法自动调整数据库参数，有助于提高调参效率，降低正确调参成本。

参数自动调优：参数调优服务支持离线与在线两种服务形式，支持多种算法，包括强化学习、全局搜索算法等。

当模型处于训练阶段中，根据输入的数据库参数数值（其中包括数据库当前参数数值以及数据库当前性能参数数值）通过强化学习和启发式算法得到新的参数数值组合，数据库的参数调整由强化学习和启发式算法两个组成部分的输出结果混合得到，将模型的输出经过反归一化得到新的参数数值，将新的数值植入数据库并运行测试作业得到当前数值组合下数据库的性能表现，如执行时长、吞吐量等指标。最后将表现作为反馈给学习模型，往复迭代。

当模型处于调优阶段中，将当前数据库的参数数值作为输入，其中包括数据库当前参数数值以及数据库当前性能参数数值。通过模型得到当前情况下的所能得到的最优参数调整方案。

当模型处于推荐模式下，会直接根据用户当前Workload的特征进行秒级参数推荐。

- 慢SQL发现

在实际生产环境中，用户通常希望作业能够以最快的方式执行成功。然而实际由于语句的复杂度不同，各个语句的执行时间并不相同。用户希望能在语句执行之前识别这些执行时间长的语句，将它们单独执行以免过长时间的锁影响其他语句的执行效果。与此同时，这种提前识别的功能又不能占用用户数据库本来的资源或影响数据库本来的响应时间。

慢SQL诊断特性利用机器学习算法构建AI框架，对语句进行语义结构的解析，并以此为基础，定性或定量地预测未知语句的执行时间，满足用户提前识别慢SQL语句的要求。

慢SQL诊断有两个主要阶段：训练阶段和预测阶段。

- 训练阶段：用户准备好自己的典型业务SQL历史记录，通过用户输入的日志地址导入历史SQL数据，训练智能模型。

- 预测阶段：用户输入待预测负载，模型根据每类的历史信息预测执行时间，从而发现潜在慢SQL。
- 索引推荐

支持单Query索引推荐与Workload级别索引推荐。在进行Workload级别索引推荐时，先根据AI算法，筛选出具有代表性的SQL语句。然后针对代表性的SQL语句，基于语句的语义信息和数据库的统计信息，生成最优的推荐索引。将所有语句的推荐索引作为候选索引集合，然后计算每条候选索引对workload的收益，推荐出收益最大的索引组合方案。
- 时序预测与异常检测

支持采集部署数据库宿主机上的特征信息，并将上述时序特征数据收集并存储起来。利用上述存储的数据进行时序预测，例如存储空间预测等。还支持根据上述数据进行潜在异常问题的发现，能够先知先觉地发现可能出现的问题，并作出提前部署。
- 支持DB4AI功能

兼容MADlib生态，支持70+算法，性能比原生MADlib有数倍提升，支持XGBoost、GBDT等高级算法。利用数据库实现SQL语句驱动AI任务。
- SQL执行时间预测

在查询性能调优、业务负载分析等场景，用户经常需要对SQL的执行时间进行预测，目前数据库优化器主要基于代价模型，无法准确预测执行时间。该特性通过AI模型，能够对历史执行过的查询或相似查询进行时间预测，满足SQL执行时间预测要求。

SQL执行时间预测：根据收集的历史性能数据进行编码和基于深度学习的训练及预测。

历史数据收集由数据库内核进程完成，内核进程通过curl向python端AI引擎发送https请求来（1）配置机器学习模型、（2）发送训练数据、（3）触发模型训练、（4）请求训练过程监测服务端口、（5）加载用于训练的模型、（6）使用加载好的模型进行预测。数据的编码阶段在数据库中完成，保证出库数据已脱敏。模型的预测阶段需要在查询计划生成后对整个计划进行编码并写文件发给python端，python端tensorflow计算图只需要加载一次，可以进行高度并行化的批量预测。
- 数据库监控

在日常运维过程中，用户需要持续对数据库运行状态进行监控，但是数据库内部的复杂性导致用户难以高效的提取关键数据。实现数据库自监控能够提升用户运维效率，用户只需要关心核心指标和异常数据。

  - a. 事务概要信息包括提交（commit\_counter）和回滚（rollback\_counter）的事务数，事务响应时间。提交和回滚的事务数、事务的响应时间都是自上次重启后的累计值。
  - b. Workload SQL 概要信息包括一个workload内DDL，DCL，DML分布，还包含DML类型中SELECT、UPDATE、INSERT、DELETE的个数。SQL类型分布是自上次重启后的累计值。
  - c. Workload SUID时间概要信息包括一个workload内Select、Update、Insert、Delete的耗时（累计，平均，最大，最小）。
  - d. SQL response time percentile概要信息包括过去一段时间系统中80%和95%的SQL响应时间。
  - e. Waitevents概要信息只包括单独节点上的事件等待信息，不包含全局汇聚信息。主要包括等待状态STATUS，IO事件IO\_EVENT，Lock事件LOCK\_EVENT，Lwlock事件LWLOCK\_EVENT的成功等待次数，失败等待次

数, 事件在节点上等待的总时间, 事件最小等待时间, 事件最大等待时间, 事件平均等待时间。

- f. 对于进入到Parser的SQL, 生成归一化的Unique SQL ID以及相应的SQL文本串。统计Unique SQL在各个执行阶段的时间消耗, 从而依据时间分布进行SQL性能分析和调优; 统计实例、会话在各个阶段的时间消耗, 辅助系统整体性能调优。查询SQL执行次数, SQL内核响应时间, IO时间、CPU时间、网络传输时间等, 物理、逻辑读, Select返回结果集数量、扫描元组数量、更新行、删除行、插入行等, 新生成(硬)复用(软)计划次数。

## 逻辑日志复制

在逻辑复制中把主库称为源端库, 备库称为目标端数据库, 源端数据库根据预先指定好的逻辑解析规则对WAL文件进行解析, 把DML操作解析成一定的逻辑变化信息(标准SQL语句), 源端数据库把标准SQL语句发给目标端数据库, 目标端数据库收到后进行应用, 从而实现数据同步。逻辑复制只有DML操作。逻辑复制可以实现跨版本复制, 异构数据库复制, 双写数据库复制, 表级别复制。

## 支持 WDR 自动性能分析报告

定时主动分析run日志和WDR报告(自动后台生成, 可由关键指标阈值如CPU占用率、内存占用率、长SQL比例等触发), 并生成html、pdf等格式的报表。能自动生成性能报告。WDR(Workload Diagnosis Report)基于两次不同时间点系统的性能快照数据, 生成这两个时间点之间的性能表现报表, 用于诊断数据库内核的性能故障。

WDR主要有两个组件:

- SNAPSHOT性能快照: 性能快照可以配置成按一定时间间隔从内核采集一定量的性能数据, 持久化在用户表空间。任何一个SNAPSHOT可以作为一个性能基线, 其他SNAPSHOT与之比较的结果, 可以分析出与基线的性能表现。
- WDR Reporter: 报表生成工具基于两个SNAPSHOT, 分析系统总体性能表现, 并能计算出更多项具体的性能指标在这两个时间段之间的变化量, 生成SUMMARY和DETAIL两个不同级别的性能数据。

## 增量备份/恢复 (beta)

支持对数据库进行全量备份和增量备份, 支持对备份数据进行管理, 查看备份状态。支持增量备份的合并, 过期备份的删除。数据库服务器动态跟踪页面更改, 每当一个关系页被更新时, 这个页就会被标记为需要备份。增量备份功能需要打开GUC参数enable\_cbm\_tracking, 以便允许服务器跟踪修改页。

## 恢复到指定时间点 (PITR)

时间点恢复(Point In Time Recovery)基本原理是通过基础热备 + WAL预写日志 + WAL归档日志进行备份恢复。重放WAL记录的时候可以在任意点停止重放, 这样就有在一个在任意时间的数据库一致的快照。即可以把数据库恢复到自开始备份以来的任意时刻的状态。在恢复时可以指定恢复的停止点位置为TID, 时间和LSN。

## 生成列

生成列是指由表中其他字段计算得到的列。生成列是一个SQL标准特性, SQL生成列在插入或者更新数据时自动计算, 像普通列一样需要占用存储空间。

## hash 索引

openGauss支持HASH索引，长索引列等值查询场景下性能优于BTREE；优化锁管理提供更高的并发度；提供XLOG日志防护，保障数据不丢失。

## 支持国密算法

用户认证方式（gsqsl、JDBC、ODBC）支持国密SM3算法；提供国密SM4算法接口，用于对数据进行加解密，增强数据库的安全能力。

## 插件化架构

通过对SQL引擎中语法、语义的解耦，实现openGauss的语法、语义层插件化，实现异构数据库语法模块与openGauss内核的解耦；支持算子插件化，实现特定算子从计划创建、优化到执行的插件化。

## 其他

- upsert支持子查询  
支持在upsert的更新语句之中使用子查询表达式进行赋值，且子查询表达式中支持通过excluded进行冲突行的引用。
- 列存表支持唯一索引  
支持在列存表上创建基于cbtree的唯一索引、创建主键唯一键约束，防止表中出现重复数据，扩展了列存表使用场景。
- 支持jsonb数据类型  
支持jsonb数据类型用来高效的操作JSON数据，支持对于json、jsonb的各种丰富的操作符、操作函数，支持在jsonb上创建索引，满足广泛的JSON使用与搜索场景。
- UCE故障感知与响应  
通过感知内存UCE错误产生时系统发出的SIGBUS信号，并根据所携带的物理地址，实现对应日志打印与openGauss数据库状态变化，进入对应的数据库进程退出流程。
- 支持unique sql语句监控自动淘汰  
openGauss支持unique SQL自动淘汰，使用LRU算法依据uniqueSQL的更新时间自动淘汰旧的uniqueSQL信息，保障最新的统计信息能够持续记录，提高数据库的易运维性。
- 支持gs\_cgroup负载管理  
gs\_cgroup 负载管理工具，通过在数据库内核创建与管理控制组，设置系统资源配额和资源限额，来管理用户与业务对资源的使用率与优先级，充分高效的利用机器资源。