北京金融科技产业联盟
BEIJING FINTECH INDUSTRY ALLIANCE

openGauss 金融应用关键能力研究

北京金融科技产业联盟

2024 年 6 月

版权声明

编制委员会

主任

聂丽琴

编委会成员

黄本涛　李　斌　林毅贤

编写组成员

阮桂亮　娄贺展　黄凯耀　赵　蒙　高海涛　刘杨箐

何佳佳　杨艳明　贺承汉　熊小军　周　斌　窦　欣

张　益　李　凯　王　薇　李雨晴　朱宏军　叶晓光

薛兴荣　何振岩

统稿

张　蕾

参编单位：

北京金融科技产业联盟

兴业银行股份有限公司

中国邮政储蓄银行股份有限公司

华为技术有限公司

天津南大通用数据技术股份有限公司

超聚变数字技术有限公司

目录

一、发展情况及现状

（一）研究背景

数字基础设施键核心技术的深化应用，推动了我国数据库产业的进一步发展，数据库产品不断涌现，开源软件生态建设也日趋完善。openGauss 是一种基于开源技术的关系型数据库管理系统，它采用开源模式，支持大数据处理，具有较为完善的安全机制，并与 Oracle 数据库语法兼容。研究其关键能力，对金融行业已有的应用和数据迁移具有一定价值。截至 2023 年 11 月底，openGauss 社区理事会包含了华为、超聚变、交通银行、邮储银行、招商银行、民生银行、兴业银行等多家技术和金融领域头部企业，在技术资源、技术氛围和场景多样等方面具备一定优势。本报告对照金融行业数据库需求对 openGauss 进行关键能力的分析，为金融业务创新中使用该技术提供支撑。

（二）金融行业数据库应用情况

集中式数据库在我国金融行业的使用时间很久，得到了广泛的应用。整体来说集中式数据库的应用比例高达 89%，其中银行业应用比例接近 80%，证券和保险行业的比例超过了 90%。分布式数据库近年来在我国金融业不同领域也已逐步开展应用，现已涵盖不同类型的业务系统，总体占比达到 7%，其中银行业超过了 17%，证券业和保险业分别为 3.74%和 1.92%。

2023 年，国内金融行业数据库加速发展，金融信息化研究所发布的《金融业数据库供应链安全发展报告(2022)》指

出，超过 40%的金融机构在办公和一般系统中使用了国内数据库产品，银行业、证券业和保险业核心系统的应用进展如图 1 所示。

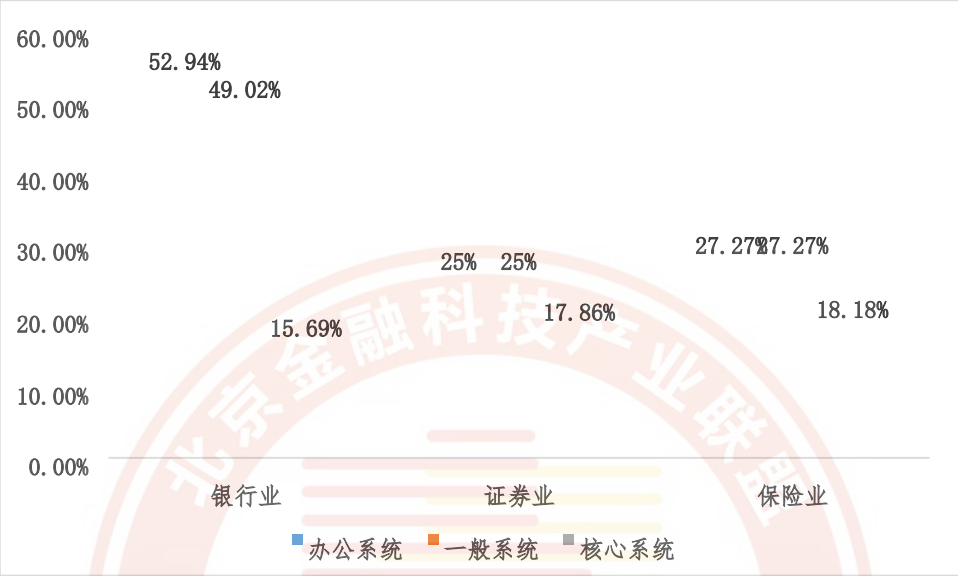


图 1 不同领域金融机构使用国内数据库产品示意图

openGauss 系数据库目前在金融行业有较多的实用案例。邮政储蓄银行通过鲲鹏、自主创新操作系统、openGauss 打造的 IT 基础设施，支持邮政储蓄银行全国 6.5 亿用户，日均 20 亿的交易，全天联机平均耗时降低 30%，系统负载峰值 TPS 提升 319%。兴业银行基于 openGauss 开源数据库，对现有业务进行优化，已在报表系统、支付系统等系统投产使用，共计应用超过 30 套系统。四川银行构建的新一代反洗钱系统，单个处理任务涉及 6 条上亿级大表的多表关联查询，运行耗时从单次 12 个小时降低至 1-1.5 小时。另外，openGauss 在保险行业和证券行业，如中华保险、上海期货交易所等企业单位，都得到了应用。

（三）数据库金融应用关键要求

数字化的浪潮给金融行业带来了业务创新，也使金融行业数据库面临着新挑战与新需求。

安全要求。金融行业的数据库在数字化时代面临着愈加严峻的数据安全挑战。金融行业是数据密集型、高安全标准和强监管的行业，数字化时代新技术的发展，也使数据库面临新的威胁手段，需要采用更加先进的技术手段保障数据安全。

高性能、高可用、可扩展与高稳定要求。金融行业传统中心化数据库架构在高频处理海量数据时面临着时延较高、扩展性能不足、一致性无法保证等问题。同时，金融行业核心业务涉及大量资金流动、客户信息、交易数据等敏感信息，高业务连续性以及数据监管合规要求也必须充分保证。此外，7*24 小时服务不间断也对金融数据库提出高稳定性要求。

易运维要求。金融行业数据库需具备业务线上化、便捷化的能力，支持在线变更，包括在线 DDL 变更、在线配置变更、在线数据变更、在线扩容与缩容，以及在线版本升级等，同时也需降低服务运维复杂度并提供问题诊断。

智能化要求。金融应用需要处理海量的结构化和非结构化数据，涉及复杂的计算和分析。金融行业数据库对利用 AI 技术实现数据库的自动优化、自动索引，提高数据处理效率和质量，降低运维成本和风险等有强烈需求。

（四） openGauss 技术特点

openGauss 总体技术架构，如图 2 所示：

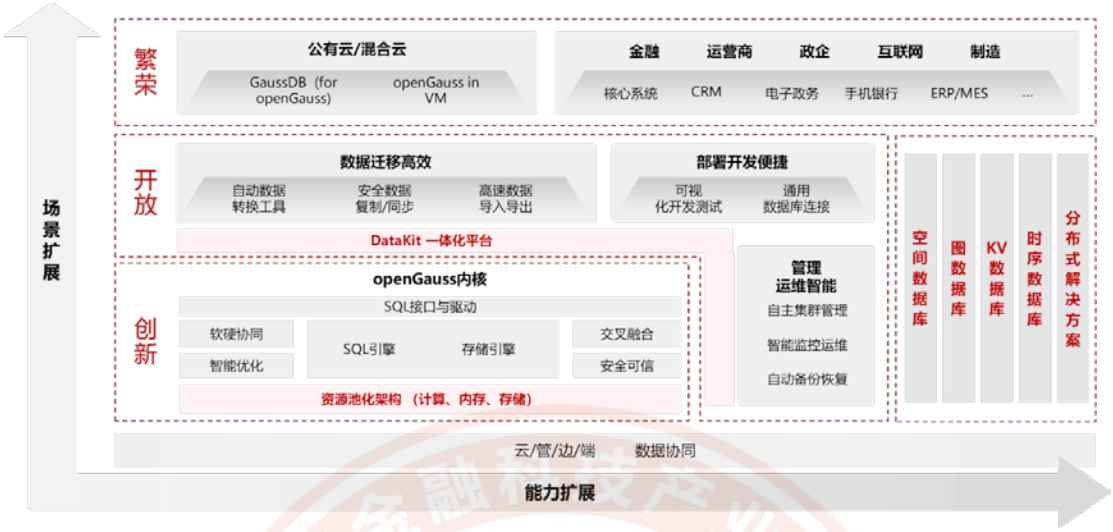


图 2：openGauss 总体技术架构

openGauss 金融版本在安全、高可靠、性能优化、智能运维方面具备如下能力：

安全：提供全密态计算、国密算法认证和加密、动态数据脱敏。

可靠：日志并行回放实现 RTO<10s，Paxos 架构，两地三中心流式容灾。

性能优化：Numa-Aware 改造，指令集优化，对应鲲鹏系列 tpmc 进行了调优。

智能化：数据库管理系统（AI4DB）提供智能索引推荐、慢 SQL 诊断等，数据库内机器学习（DB4AI）支持主流机器学习场景

资源池化：存储池化、内存池化和计算池化三层池化架构，支持应用横向扩展

二、安全研究

本章根据金融行业数据库数据安全要求，从数据安全架

构、全栈国密、全密态数据三方面展开 openGauss 研究。

（一）安全架构

金融领域数据库的核心任务是帮助用户安全的存储和管理数据，保证复杂环境下数据不丢失、隐私不泄露、数据不被篡改以及服务不中断。openGauss 数据库在安全审计、用户数据保护、用户识别和认证、安全管理、TSF 自保护、TOE 访问功能满足 CC EAL4+安全认证的要求，其安全架构的主要模块和采用的技术包括:

安全感知框架: 包括数据库防火墙的入侵防御、基于 AI 的攻击识别及智能防御。

安全认证: 数据库服务端的强认证机制。

访问控制: 具备权限管理模型、对象访问控制及校验机制。

数据脱敏与加密: 对关键数据采用数据加密存储机制或数据静态脱敏及动态脱敏机制保护。

数据防篡改: 采用多副本备份和区块链技术对数据进行一致性保护。

数据库审计: 通过系统内部细粒度审计机制，记录用户操作行为。

第三方安全测试: 引入第三方安全测试和认证，加速完善数据库安全能力的构建。

（二）全栈国密

《中华人民共和国密码法》要求关键信息基础设施应当使用商用密码进行保护，并开展商用密码应用安全性评估（密评）。中国人民银行发布《金融行业信息系统商用密码应用基本要求》及配套测评规范，推动相关测评机构开展金融信息系统密评工作。openGauss 已通过国密局数据库国密认证，全栈国密如图 3 所示：

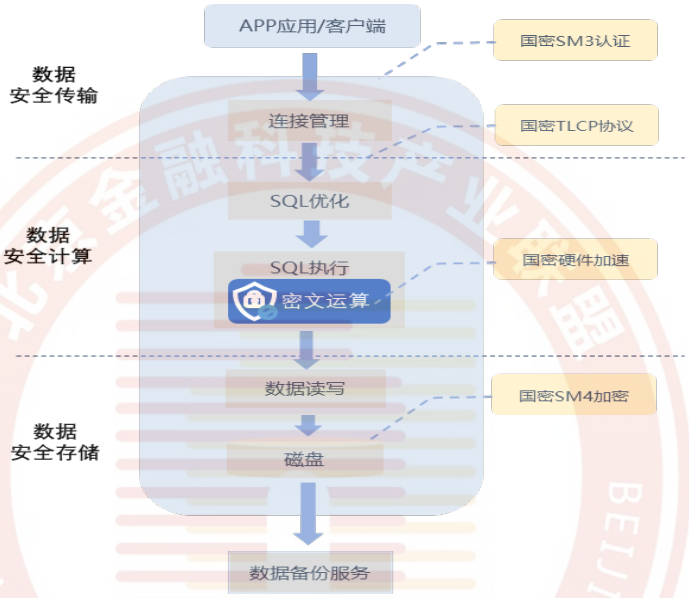


图 3 openGauss 全栈国密

openGauss 支持使用国密算法 SM3 进行用户登录认证，支持使用国密算法 SM4 进行数据加密。在鲲鹏 920 上，通过使用内置的 KAE 引擎实现 SM4 加密算法加速 4.6x、SM3 哈希算法加速 12.6x，如图 4 所示。

openGauss 支持国密 TLCP 协议，增加支持的国密加密套件，包括 ECDHE-SM4-SM3、ECDHE-SM4-GCM-SM3、ECC-SM4-SM3、ECC-SM4-GCM-SM3。其中，TLCP 是指符合《GB/T38636 2020 信息安全技术 传输层密码协议（TLCP）》的安全通信协议，其特点是采用加密证书/私钥和签名证书/私钥相分离的方

式。

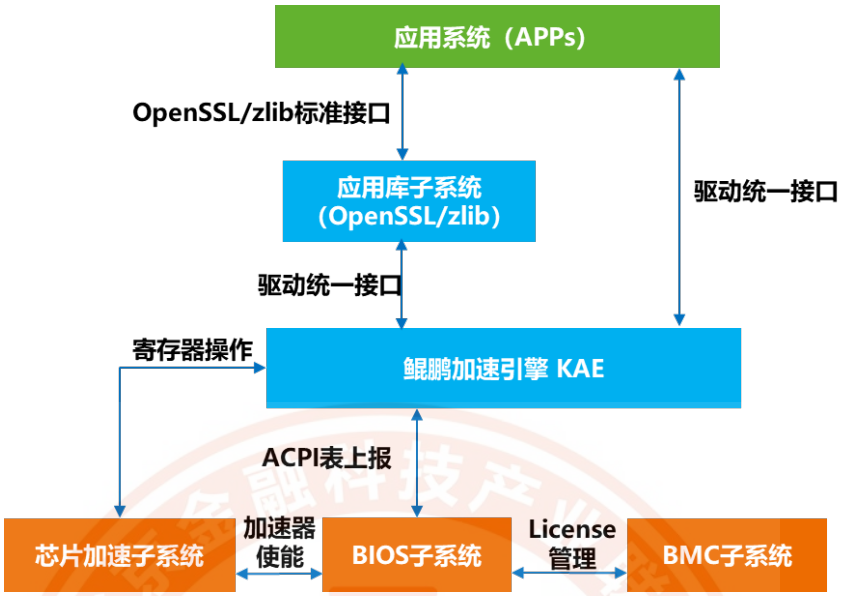


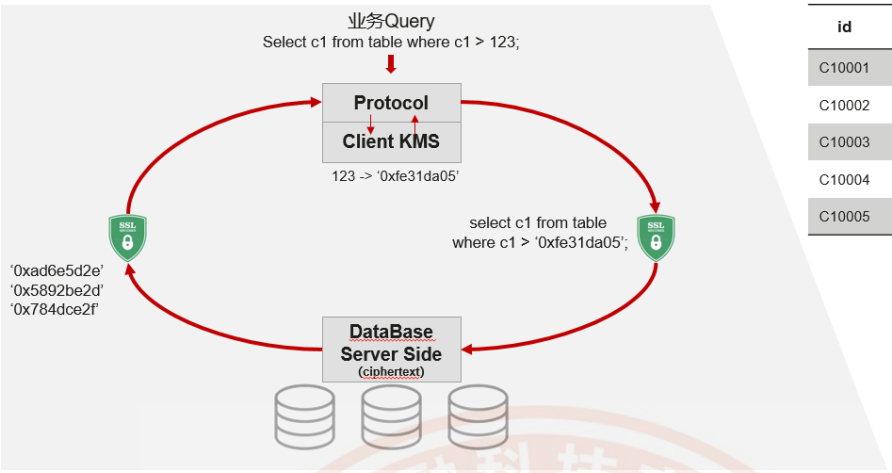
图 4 鲲鹏 920 KAE 加速引擎

（三）全密态数据库

密态数据库是专门处理密文数据的数据库系统，数据以加密形态存储在数据库服务器中，数据库支持对密文数据的检索与计算，而与查询任务相关的词法解析、语法解析、执行计划生成、事务 ACID、数据存储都继承原有数据库能力。

在密态数据库机制下，业务数据流图如下图 5 所示。假定数据列 c1 已以密文形态存放在数据库服务端，用户发起查询任务指令。用户发起的查询任务无需进行特殊化改造，对于查询中涉及的与敏感数据 c1 相关联的参数，在客户端按照与数据相同的加密策略(加密算法，加密密钥等)完成加密，如图 5 中关联参数“123”被加密成“0xfe31da05”。参数加密完成后整个查询任务被变更成一个加密的查询任务并通过安全传输通道发到数据库服务端，由数据库服务端完成基于密文的查询检索。检索得到的结果仍然为密文，并最

终返回客户端进行解密。



id	c1 (plaintext)	c1 (ciphertext)
C10001	256	0xad6e5d2e
C10002	157	0x5892be2d
C10003	97	0x124f4ed2
C10004	685	0x784dce2f
C10005	58	0x324f4ed2

图 5 密态数据库核心业务数据流

密态数据库的核心思想是：由用户持有数据加解密密钥且数据加解密过程仅在客户侧完成，数据以密文形态存在于数据库服务侧的整个生命周期过程中，并在数据库服务端完成查询运算。openGauss 全密态数据库支持密码学软件模式和基于 TEE(Trusted Execution Environment)的硬件模式。

在硬件模式下，openGauss 支持鲲鹏 ARM TrustZone 较高级别隔离，并且通过多层密钥管理体系、可信传输通道、会话级密钥管理机制等密钥安全保障机制实现硬件环境中的数据及密钥安全，降低因硬件安全问题而导致的用户数据及密钥泄露风险。

在软件模式下，openGauss 支持密态查询，通过密态查询引擎，实现数据等值查询、范围查询、保序查询、表达式计算等。并通过引入确定性加密机制，实现数据的增删改查、表字段关联、等值检索等操作。基于 GS-OPE 算法的密文索引技术，实现数据密态保序查询、表达式大小比较等。通过

Range-Identify 算法，实现数据密态范围查询。

三、多模多态分布式研究

本章针对金融行业数据库高性能、高可用、可扩展与高稳定需求，研究了 openGauss 的分存储模式、部署形态、分布式能力。

（一）多存储模式

openGauss 系统支持以下三种存储引擎：行存储引擎、列存储引擎、MOT 内存引擎。

1.行存引擎

行存储引擎的特点是支持高并发读写，时延小，适合 OLTP 交易类业务场景。openGauss 行存储引擎采用原位更新(in-place update)设计，支持 MVCC(Multi-Version Concurrency Control，多版本并发控制)，同时支持本地存储和存算分离的部署方式，支持存储层异步回放日志等。

行存储引擎 Ustore 将最新版本的“有效数据”和历史版本的“垃圾数据”分离存储。将最新版本的“有效数据”存储在数据页面上，并单独开辟一段 UNDO 空间，用于统一管理历史版本的“垃圾数据”，因此数据空间不会由于频繁更新而膨胀，“垃圾数据”集中回收效率更高。

2.列存引擎

传统行存储数据压缩率低，必须按行读取，即使读取一列也必须读取整行。在数据库遇到针对大量表的复杂查询，而这种复杂查询中仅涉及一个较宽(表列数较多)的表中个别列时，行存储以行作为操作单位，会引入与业务目标数据

无关的数据列的读取与缓存，造成了大量 IO 的浪费，性能较差。因此 openGauss 提供了列存储引擎的相关功能。创建表的时候，可以指定行存储还是列存储。

openGauss 行列混合存储是指将表按行存储到硬盘分区上，列存储是指将表按列存储到硬盘分区上。行、列存储模型各有优劣，建议根据实际情况选择。其通常用于 OLTP（联机事务处理）场景的数据库，默认使用行存储，仅对执行复杂查询且数据量大的 OLAP（联机分析处理）场景时，才使用列存储。默认情况下，创建的表为行存储。

3.内存引擎

内存引擎 MOT 作为在 openGauss 中与传统基于磁盘的行存储、列存储并存的一种高性能存储引擎，提供了高实时数据处理分析能力及低事务处理时延。内存引擎基于内存而非磁盘进行事务处理，包括且不限于数据及索引结构、数据管控、基于 NUMA 架构的内存管控、数据处理算法优化及事务管理机制完善。

全内存态存储并不代表着内存引擎中的处理数据会因为系统故障而丢失。内存引擎具有与 openGauss 的原有机制相兼容的并行持久化、检查点能力，使得内存引擎有着与其他存储引擎相同的容灾能力和高可靠能力。

（二）多部署形态

多部署形态分为主备式部署形态、分布式部署形态，如

图 6 所示。

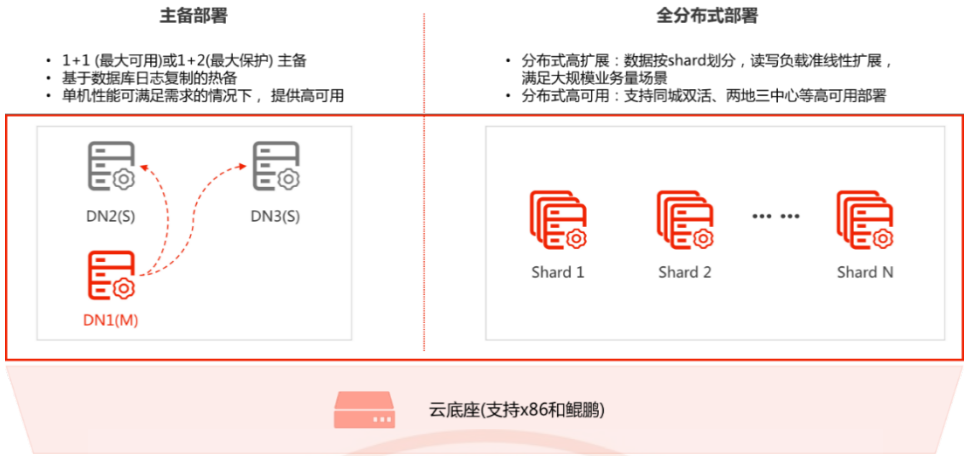


图 6 主备部署形态与分布式部署形态

1. 主备部署

主备模式相当于两个数据副本，主机和备机各一个数据副本，备机接受日志、执行日志回放。主备式部署形态支持一主多备模式。在一主多备模式下，所有的备机都需要重做日志，都可以升主。一主多备提供更高的容灾能力，更加适合于大批量事务处理的 OLTP 系统。主备之间可进行角色切换，主机故障后可对备机进行升主。主备模式提供了抵御实例级故障的能力，适用于不要求机房级别容灾。

openGauss 的主备部署形态可基于资源池化架构，该架构由三层池化、一个平台和一个标准组成，如图 7 所示：

图 7 openGauss 资源池化架构

openGauss 提供关系型 OLTP SQL 语法、OLAP SQL 语法、AI 推理等接口，并基于 X86、鲲鹏等算力，为应用提供 TP 行存加速、AP 列存加速、AI 训练推理、向量数据库等全方位的数据服务。可使用引擎满足不同业务处理诉求，实现计算池化。通过同步事务信息和数据库缓存，实现多节点下的多版本快照一致性读。结合 RoCE 和 SCM 等硬件，实现 Commit 加速和大容量内存访问，实现内存的池化。支持分布式存储、企业存储、对象存储等，通过高效裸设备访问，元数据共享，实现存储池化。

2.分布式部署

openGauss 社区版目前只支持主备部署形态。基于 openGauss 内核的 GaussDB 和 GBase 8c 支持分布式部署形态，分布式部署形态采用全组件的高可用冗余，即所有节点都支持分布式高可用部署。

GaussDB 的分布式架构和组件，如图 8 所示：

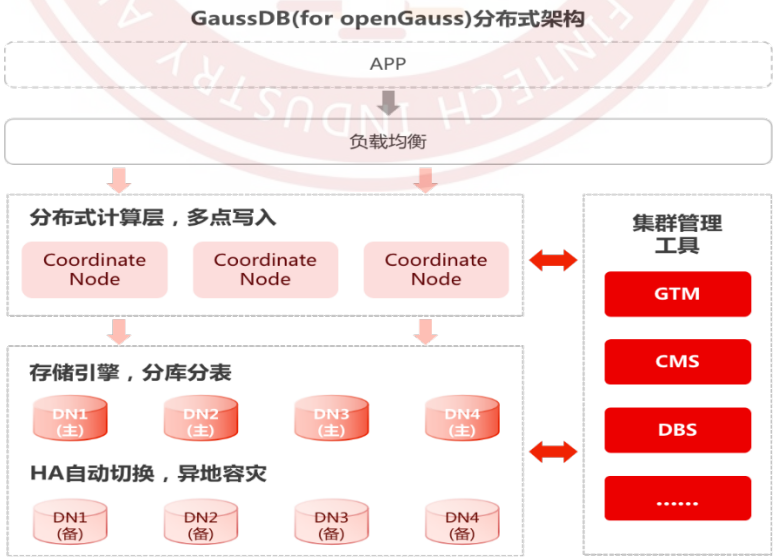


图 8 GaussDB 的分布式架构和组件

CM 集群管理模块（Cluster Manager）： 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。

GTM 全局事务控制器(Global Transaction Manager): 提供全局事务控制所需的信息,采用多版本并发控制 MVCC 机制。

WLM 工作负载管理器（Workload Manager）： 控制系统资源的分配，防止过量业务负载，对系统的冲击导致业务拥塞和系统崩溃。

CN（Coordinator Node）： 整个系统的业务入口和结果返回；接收来自业务应用的访问请求，分解任务并调度任务分片的并行执行。

DN（Data Node）： 执行查询任务分片的逻辑实体。

ETCD（Editable Text Configuration Daemon）: 分布式键值存储系统用于共享配置和服务发现（服务注册和查找）。

GBase 8c 的分布式架构和组件如图 9 所示:

图 9 GBase 8c 的分布式架构和组件

CN: 协调节点，采用完全对等的部署方式。对外提供接口，负责进行 SQL 解析和优化、生成执行计划，并协调数据

节点进行数据查询和写入。在功能上 CN 上只存储系统的全局元数据，并不存储实际的业务数据。

DN: 数据节点，采用主备的高可用架构，主备之间可以配置同步或异步方式。用于处理存储本节点相关的元数据，每个节点还存储它所在的业务数据的分片。在功能上，DN 节点负责完成执行协调器节点分发的执行请求，完成数据存储和本地数据查询和写入;

GTM: 全局事务管理器，采用主备的高可用架构，主备之间可以配置同步或异步方式。主要是做分布式事务，负责生成并维护全局时间戳，保证集群数据一致性。在部署上，GTM 的与数据节点的部署类似，也是一主多从的备份方式，节点间可以采用同步的备份方式也可以采用异步的备份方式。

HA Center: 集群状态管理器，采用 Raft 的复制协议。存储各个节点的高可用状态，负责在故障情况下判断集群各个节点状态。

GHA Server: 集群管理器，采用主备的高可用架构，主备之间可以配置同步或异步方式。用以管理整个集群各个节点的高可用状态（主备、死活等）。

（三）关键能力

1.分布式查询能力

分布式架构需要最大化利用架构下的整体资源，进行分布式查询的性能提升与优化，且随着节点规模的扩大，线性增长整体性能。具体来说，主要采用以下几种方式提高性能。

分布式路径搜索: openGauss 优化引擎生成分布式路径，

将同一个表的数据分布到不同的数据节点上。创建表时可选择将数据在每个表上做哈希分布或随机分布。为了正确执行两表连接操作，可能需要将两个表的数据重新分布，因此 openGauss 的分布式执行计划中增加了对数据进行重分布的 **Redistribute** 和 **Broadcast** 两个算子。**Redistribute** 算子将一个表的数据按照执行的哈希值在所有的数据节点上做重分布，**Broadcast** 算子通过广播的方式重新分布一个表的数据，保证广播之后每个数据节点上都有这个表的数据副本。分布式路径生成时，会考虑两表及连接条件上的数据是否处于同一个数据节点，如果不是，会添加相应的数据分发算子。根据分发算子需处理的数据量以及网络通信消耗，可计算路径代价，优化引擎根据代价选出最优路径。

分布式执行计划：openGauss 为提高分布式架构下资源利用率，提供四种执行计划，分别为 LightProxy（CN 轻量化）计划、FQS（Fast Query Shipping）计划、Stream 计划以及 Remote-Query 计划，其中 FQS 和 Stream 是可下推计划。也就是说，集群中的所有数据节点都参与 SQL 执行，如图 10。

图 10: 分布式执行计划

FQS 计划是协调节点直接将原语句下发到各个或者部分

数据节点上，各节点单独执行，相互之间不进行数据交互，而 Stream 计划是原语句在协调节点上生成执行计划，然后协调节点将执行计划下发到各个数据节点上，各数据节点在执行过程中使用 Stream 算子进行数据交互。

分布式数据存储与管理：openGaus 采用 Hash、Replicate 等分片策略把数据尽可能均匀分布到各数据节点，实现不同查询场景的性能优势或采用复制表实现更快速的本地连接查询，如图 11 所示。

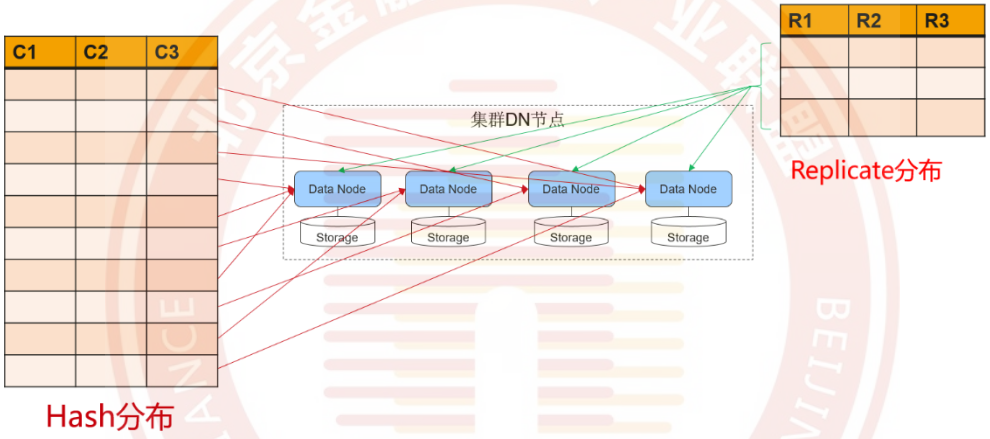


图 11：分布式数据存储与管理

2. 组件高可用能力

openGauss 组件高可用部署方式包括：

同城灾备方式：抵御硬件级别、机房级别灾难，两机房之间距离小于 50 千米；异地灾备、两地三中心方式：抵御硬件级别、机房级别、城市级别灾难，两地之间距离可以大于 1000 千米。异地多活方式：支持多集群部署，集群间数据双向同步。

组件高可用同机房/同城部署方案：同机房部署，顾名思义就是将整个集群全部部署在同一个机房内，如图所示就是一个同机房的典型部署，集群内 3 个数据节点分别部署在三个数据中心里；2 个协调节点高可用组，每组一主两备，分别部署在 3 个数据中心里；GTM 一主一备，部署在任意 2 个数据中心；HA Center 是负责记录各个节点高可用状态和必要时候的主备倒换的节点，它是采用 Raft 的复制协议，分别部署在 3 个数据中心，如图 12 所示。

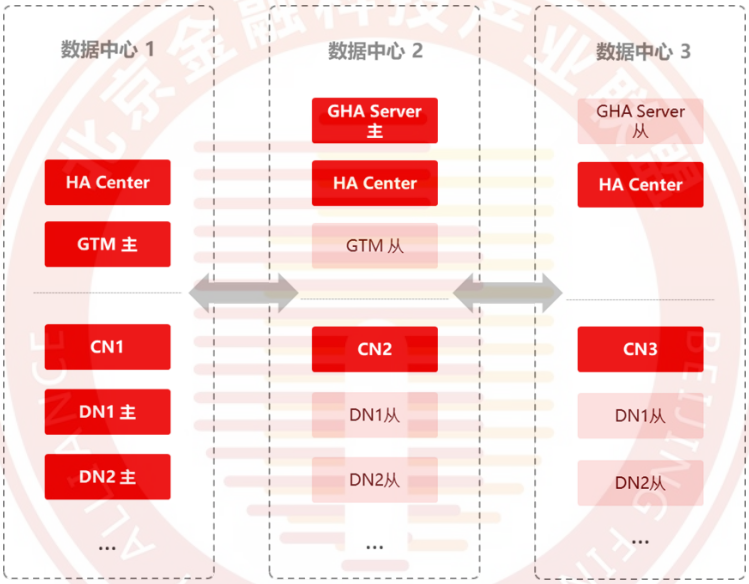


图 12 全组件高可用同机房/同城部署方案

集群的每个组件数量和部署方式可根据具体情况进行调整，以保证任意数据中心宕机时，其他两个数据中心组成的集群仍可对外提供服务。在同机房的部署方式下，集群可抵御硬件级别的故障，不能抵御城市级别和机房级别的灾难。

组件高可用单一集群异地灾备部署方案：该部署方式可支持单一集群异地灾备，适用于对时延要求不太高但是对异地高可用级别要求比较高的系统，其架构方案如图 13 所示。

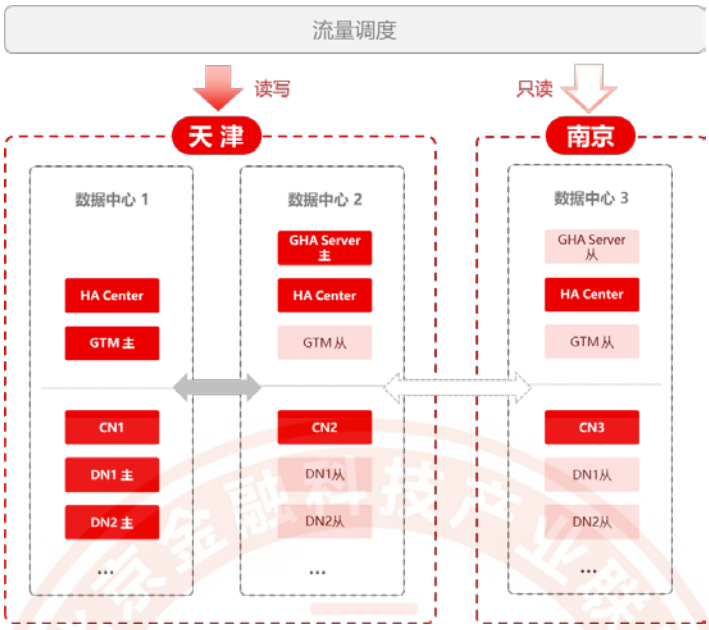


图 13：组件高可用单一集群异地灾备部署方案

图中部署方式可在天津配置读写，在南京做只读的业务。天津部署两个数据中心，协调节点和 GTM 节点主备之间采用同步备份方式；南京部署第三个数据中心，协调节点和 GTM 节点与位于天津的主节点之间采用异步备份方式。此类情况要保证异地每个集群组件最少有一个节点或备机，整个集群可抵御硬件级别故障和机房级别以及城市级别的灾难，两地间距离可大于 1000 千米。

组件高可用两个集群异地多活部署方案：此方案可支持通过集群间的数据同步配合业务层面进行异地多活，方案架构如图 14 所示。

图中方式下，交易按照分区调度，比方说天津和南京两地，天津处理某种交易，南京处理另一种交易，这个分类可以是地理上的，也可以是业务上的。然后两地之间再进行数

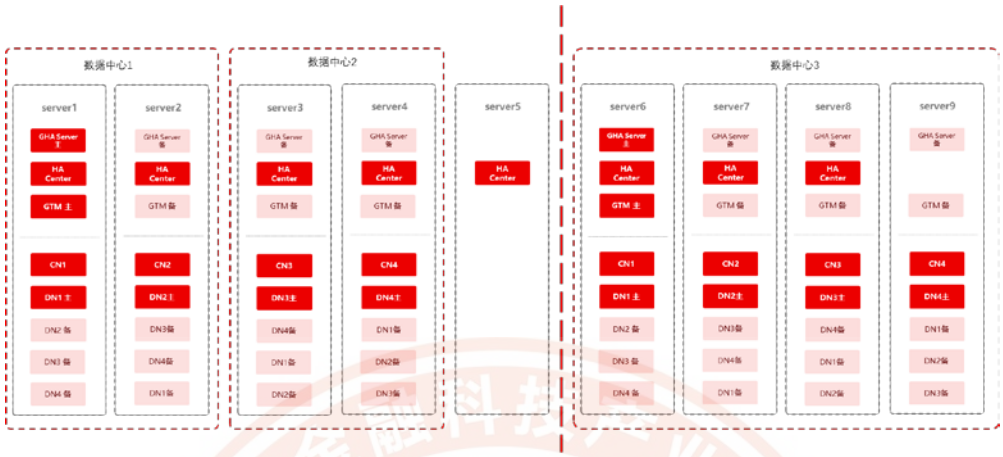
据同步。这种方式可支持多集群部署，集群间数据双向同步，数据以某一维度进行分区调度，可抵御硬件级别故障和机房级别、城市级别灾难，两地之间距离可以大于 1000 千米。

图 14 组件高可用两个集群异地多活部署方案

组件高可用两地三中心部署方案：此方案需要异地数据中心 3 的 DN 高可用组数量和配置与本地集群的 DN 高可用组的数量和配置完全相同。例如：本地集群 DN 共 4 分片，每个分片 1 主 4 备。要求异地集群 DN 共 4 分片，每个分片 1 主 4 备，方案架构如图 15 所示。

在图中 server5 服务器上部署的 ETCD 为仲裁节点，保证在数据中心 1 或数据中心 2 任意中心单独下线情况下，进行多数派仲裁，自动切换至剩余的数据中心继续对外服务，同时防止脑裂。建议同一城市单独部署。节点对服务器配置要求低，可与其他应用/产品部署在同一服务器，仅做节点状态仲裁，不涉及业务。异地容灾建议配置手工切换，防止异

地数据中心 3 因网络原因导致自动拉起的情况。组件高可用两地三中心部署方案中的数据同步以及数据一致性的保障



方式是采用集群管理机制实现的。

图 15 组件高可用两地三中心部署方案

集群管理 CM(Cluster Manager)机制支持 VIP(Virtual IP)管理。业务通过配置 VIP 连接数据库，当主机故障发生主备切换时，业务连接可自动重连到新的主机(毫秒级别)；当数据库出现双主时，通过 VIP 连接数据库可确保连接唯一的主机，降低双主丢数据的风险。

集群管理 CM 也支持两节点部署，通过引入第三方网关 IP，有效解决 CM 集群两节点部署模式下自仲裁问题，支持 CMS 和 DN；同时支持动态配置 CM 集群故障切换策略和数据库集群脑裂故障恢复策略，从而能够尽可能确保集群数据的完整性和一致性。

集群管理模块的磁盘只读检测能力增强，实现只读状态从数据库获取，保证准确性；只读仲裁只仲裁当前超过阈值的实例，其他节点不受影响；主机只读保护后自动主备切换，选可用备机升主保证集群能正常提供工作。

集群管理机制支持按事件调用用户自定义脚本，支持 CM 组件单独升级，增强数据库集群可靠性；CM 可以根据配置信息，支持用户自定义组件的监控和管理。

集群管理机制实现 DCF（Distributed Consensus Framework，分布式共识框架，基于 Paxos 算法实现数据同步强一致）支持策略化多数派能力，以多数派为前提，同时根据用户配置的 AZ，保证 AZ 内至少有一个节点同步复制日志。从而实现两地三中心部署方案中集群节点的数据同步以及集群所有节点的数据一致性。

3. HTAP 能力

基于资源池化架构，openGauss 具备对 OLTP 和 OLAP 及其他数据模型混合负载能力，包括 SMP 并行查询、RSMP 多机并行、NDP 近数处理、分布式 OLTP 中间件、分布式 OLAP 中间件等。

SMP: openGauss 的 SMP 是一种利用多核 CPU 架构实现多线程并行计算以提高查询性能的技术。思路是对于能够并行的查询算子，将数据分片，启动若干个工作线程分别计算，最后将结果汇总，返回前端。在复杂查询场景中，SMP 比较低并发度的单个查询能够减少查询执行时间，提升查询性能及资源利用率。

多机并行： openGauss 资源池化架构下，当一个节点发起 SQL 查询请求，且通过代价判断应启动多机查询，则生成分布式计划，再根据设定的 Node-Group，将此查询在 Node-Group 内所有节点多机并行执行，执行后结果从发起的节点

返回。

近数处理：在 openGauss 资源池化架构下，近数处理指把某些算子卸载到存储节点上运行，以减少数据网络的传输，提升查询的性能。近数处理的总体方案架构如图 16 所示。

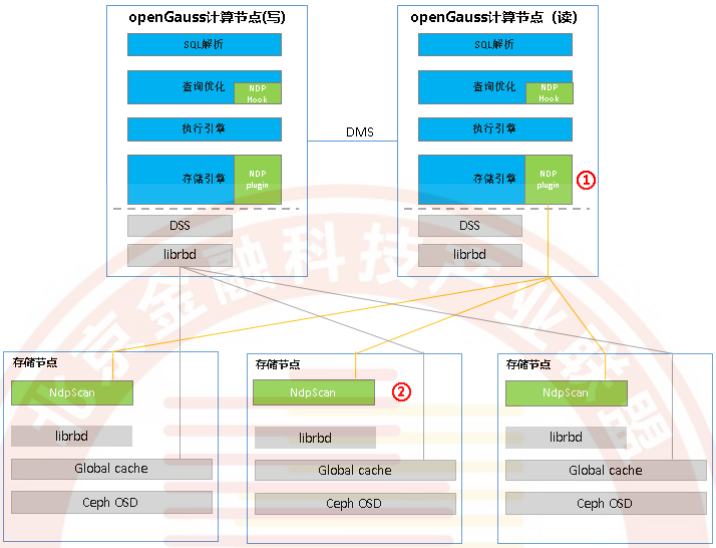


图 16 近数处理总体方案架构

计算侧数据库内核中以插件的方式提供算子卸载功能。在存储侧，部署 NdpScan 进程，提供近存储数据计算的功能。计算侧算子卸载并不会改变原有的执行计划和算子的相关信息，通过存储引擎接口 hook 实现 NDP 数据处理逻辑，屏蔽了执行引擎感知卸载，只存在少量必须的 Hook 点。存储节点上运行分布式存储软件，Global Cache，以及 NdpScan 进程。主要资源还是分配给分布式存储软件，因此 NdpScan 只能运行必要的功能，并且 NdpScan 不感知文件，只是按照存储的语义(image，object)方式读取数据然后按照 PAGE_SIZE 粒度进行解析，之后对这些数据进行过滤或者聚合操作。

分布式 HTAP 解决方案：openGauss 社区通过协同分布式

OLTP 中间件 ShardingSphere 和融合分析引擎 openLooKeng，实现 HTAP 解决方案。总体架构如图 17 所示。

图 17 分布式 HTAP 解决方案总体架构

ShardingSphere 是一套开源的分布式数据库中间件，提供标准化的数据分片、分布式事务和数据库治理功能。openLooKeng 是一款开源的高性能数据虚拟化引擎，提供统一 SQL 接口，具备跨数据源/数据中心分析能力以及面向交互式、批、流等融合查询场景。openGauss 分布式 HTAP 解决方案通过并行扫描、算子卸载等，发挥多节点算力，进行复杂 SQL 处理。分布式 OLTP 组件和分布式 OLAP 组件共享注册中心，实现对用户数据表元数据的管理，可支持超过 1024 个数据分片。数据分片的类型既可以是传统的主备模式，也可以是资源池化模式。

四、内核可观测研究

数据库在运行过程中，应提供数据库的运行指标、当前运行状态、当前资源使用情况等信息的访问接口以便数据库使用者能清晰了解数据库当前状态，在出现故障时准确定位

故障原因。本章从内核可观测性维度和可观测性接口两方面研究 openGauss 数据库可观测性能力。

（一）观测维度

数据库内核可观测性能力可从会话、事务、语句 3 个维度评估。

1.会话可观测，包括:

每个会话的连接数、执行时间、锁等待情况以及查询的执行计划和效率。这样可以快速识别出可能导致性能下降或故障的会话，并采取相应措施进行优化或修复。

会话跟踪和日志记录功能，以便详细了解每个会话的操作、语句执行和错误情况，可通过分析会话级别的日志和跟踪数据，并进行故障排查。

会话之间的相互影响和资源竞争情况。通过监测会话级别的资源利用率和锁等待情况，发现潜在的瓶颈或资源争夺，并进行调整以提高整体数据库性能。

2.数据库事务可观测包括:

事务执行时间：记录每个事务的开始时间和结束时间，并计算事务的执行时间。这有助于评估数据库系统的响应性能和吞吐量。

事务并发度：跟踪并发执行的事务数量，了解数据库系统的并发负载状况，以便进行性能分析和调优。

锁和资源争用：监控数据库事务对锁的请求、获取和释放过程，检测和量化数据库中的锁争用情况和资源瓶颈，有助于解决并发性能问题。

事务隔离级别和一致性：追踪和度量不同事务隔离级别下的一致性读写操作，以验证数据库系统是否符合事务隔离规范，并辅助调优事务的一致性需求。

异常和回滚：记录事务执行过程中的异常情况和回滚操作，例如死锁、超时等，以及事务的回滚次数和原因，有助于故障排查和改进事务处理机制。

3.语句可观测，包括：

SQL 执行时间：记录每个 SQL 语句的开始时间和结束时间，并计算 SQL 执行时间。这有助于评估数据库系统的响应性能和吞吐量。

SQL 执行计划：捕获 SQL 查询执行计划，并提供一系列指标和统计信息，如访问路径、扫描方式、索引使用等，以便进行性能分析和调优。

索引和表扫描:监控 SQL 查询使用的索引和表扫描方式，检测和量化索引的效率和查询优化的效果，有助于解决慢查询和高 CPU 占用问题。

缓存命中率：跟踪 SQL 查询的缓存命中率和缓存大小，以评估缓存使用效果和可用性，提高查询的效率和性能。

查询分布和监控：收集和分析 SQL 查询的分布情况和趋势，检测和诊断慢查询的原因和来源，辅助调优数据库的设计和配置。

（二）观测接口

openGauss 提供 DBE_PERF 内视图、WDR、ASP 等接口支持数据库内核的观测。

1.DBE_PERF 内视图

DBE_PERF Schema 内视图提供主要的资源管理、性能监控方面的相关视图，支持实时 TOP SQL、历史 TOP SQL、内存、操作系统运行等性能监控数据，用来诊断性能问题。

如表 1 所示，openGauss 提供 query 级别和算子级别的资源监控实时视图来查询实时 TOP SQL。

表 1 实时 TOP SQL 接口-视图级别表

视图级别	查询视图
Query 级别	DBE_PERF.STATEMENT_COMPLEX_RUNTIME
Operator 级别	DBE_PERF.OPERATOR_RUNTIME

资源监控实时视图记录了查询作业运行时的资源使用情况(包括内存、CPU 时间、IO 等)以及性能告警信息。

如表 2 所示，openGauss 提供 query 级别和算子级别的资源监控历史视图来查询历史 TopSQL。

表 2 历史 TOP SQL 接口-视图级别表

视图级别	查询视图
Query 级别	DBE_PERF.STATEMENT_COMPLEX_HISTORY
Operator 级别	DBE_PERF.OPERATOR_HISTORY

资源监控历史视图记录了查询作业运行结束时的资源使用情况(包括内存、下盘、CPU 时间、IO 等)、运行状态信息(包括报错、终止、异常等)以及性能告警信息。但对于

FATAL、PANIC 错误导致的查询异常结束，状态信息列只显示 aborted，无法记录详细异常信息。

DBE_PERF.STATEMENT_COMPLEX_HISTORY 视图显示在数据库主节点上执行作业结束后的负载管理记录。

STATEMENT_COMPLEX_HISTORY_TABLE 视图显示数据库主节点执行作业结束后的负载管理记录，所显示数据为内核中归档转储到系统表中的数据。

2.WDR 报告

openGauss 提供 WDR(Workload Diagnosis Report)负载诊断报告，记录整个数据库在运行期间的现状或者说真实状态，常用于判断长期性能问题。WDR 报告在打开参数 enable_wdr_snapshot 后生成，其相关参数如表 3 所示。

表 3 WDR 报告参数表

参数	参数说明
enable wdr_snapshot	是否开启数据库监控快照功能
wdr_snapshot interval	后台线程 Snapshot 自动对数据库监控数据执行快照操作的时间间隔
wdr_snapshot query_timeout	系统执行数据库监控快照操作时，设置快照操作相关的 sql 语句的执行超时时间。如果语句超过设置的时间没有执行完并返回结果，则本次快照操作失败

参数	参数说明
wdr snapshot retention days	系统中数据库监控快照数据的保留天数，超过设置的值之后，系统每隔 wdr_snapshot interval 时间间隔，清理 snap_shot_id 最小的快照数据

3. ASP

openGauss 的 ASP（Active Session Profile）类似于 Oracle 的 ASH 用于监控和分析数据库中的活跃会话。openGauss 的 ASP 支持三种持久化模式：数据表保存是将数据定期持久化到 GS_ASP 表中；文件保存是将内存中的 ASP 数据输出到磁盘文件中；还有表+文件保存模式。

4. 内核 eBPF

eBPF（Extended Berkeley Packet Filter）是一种内核级的性能监控工具。eBPF 可收集各种系统信息，如 CPU 使用率、内存使用情况、网络流量等。通过执行 eBPF 程序，开发者可以实现对系统的深度监控和数据分析。

五、AI 能力研究

AI 与数据库结合是近些年的行业研究热点，openGauss 较早地参与了该领域的探索，并取得了阶段性的成果。

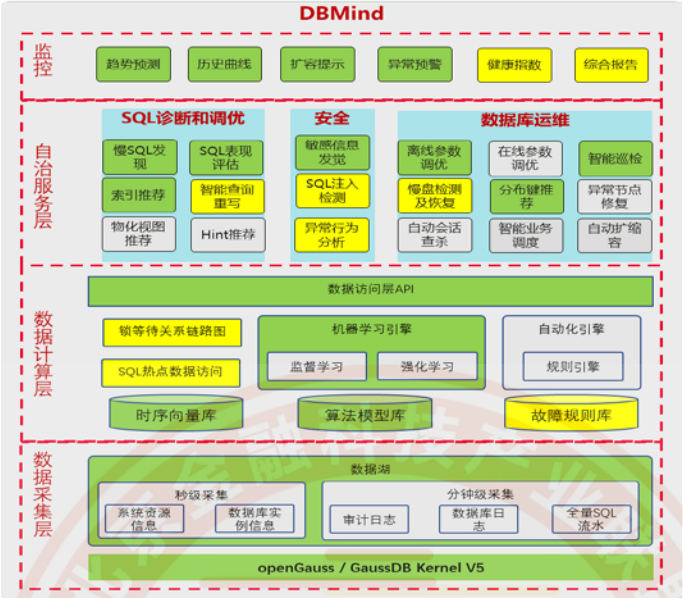
本章从 AI4DB 和 DB4AI 两个方面研究金融行业数据库智能化需求下，openGauss 数据库 AI 能力。

（一）AI4DB：智能运维能力

AI4DB 是指用人工智能技术优化数据库的性能，也可以通过人工智能实现自治、免运维等。AI4DB 主要包括自调优、

自诊断、自安全、自运维、自愈等。

openGauss 的 AI4DB 特性子模块 DBMind 系统架构如图



18 所示:

图 18 DBMind 系统架构

DBMind 的数据采集层负责采集必要的数据供上层调用;数据计算层包含大量模型和故障规则库等,为自治服务层提供算法支撑;自治服务层负责实现例如故障诊断、索引推荐、慢 SQL 诊断等具体功能;监控层在自治服务层的基础上提供优化建议、输出预警信息等。

DBMind 主要包含四个模块:数据库指标采集和预测与异常监控、慢 SQL 发现与根因分析、智能索引推荐、智能调优与诊断。

数据库指标采集、预测与异常监控: DBMind 提供 openGauss-exporter(采集数据库指标)和 reprocessing-exporter(采集到的指标进行二次加工)与 Prometheus 对接,对采集到的指标进行预测分析,如通过预测内存使用率

发现内存泄漏、预测磁盘使用情况、预测合适扩容时机等。AI 异常检测算法可根据指标走势，帮助用户及时发现问题。用户也可通过修改配置文件来指定需要进行预测的关键系统指标（KPI），及时进行运维操作。

慢 SQL 发现与分析： SQLdiag 是一个 SQL 语句执行时间预测工具，通过模板化方法，实现在不获取 SQL 语句执行计划的前提下，依据语句逻辑相似度与历史执行记录，预测 SQL 语句的执行时间。慢 SQL 发现分为两个主要过程。在训练阶段，通过历史 SQL 数据对自编码模型，聚类模型及执行时间序列模型进行 AI 模型训练，训练后的模型用于后续慢 SQL 发现的推理过程。推理阶段，用户输入待预测负载，系统根据训练阶段生成的自编码模型对待预测负载进行编码，根据训练阶段生成的聚类模型进行分类，进而根据每类的历史信息预测 SQL 语句的执行时长，并判断和发现执行效率低、性能不佳的 SQL 语句。openGausss 慢 SQL 分析工具结合 openGauss 自身特点，融合 DBA 慢 SQL 诊断经验，能同时按照可能性大小输出多个根因并提供针对性的建议，简化运维人员的工作，

智能索引推荐： openGauss 智能索引推荐，可对单 Query 查询或包含多条 DML 语句的 workload 级别查询，通过 AI 算法自动学习与智能推荐索引优化方案。虚拟索引功能可通过优化器建立模拟索引，用以评估该索引对指定查询语句的代价影响，为用户提供优化、可靠的索引建议。

智能参数调优与诊断： openGauss 集成基于 AI 的参数调优与诊断工具，通过深度强化学习和全局搜索算法等 AI 技

术，自动智能的提供最佳数据库参数配置方案。该工具可以在多场景下，快速给出当前负载的参数配置，减少 DBA 工作量，提升运维效果。参数调优与诊断工具包含三种运行模式：Recommend 模式、Train 模式、Tune 模式。**Recommend 模式，**用户在数据库系统中直接获取当前正在运行的 workload 特征信息，根据上述特征信息生成参数推荐报告：提示当前数据库中不合理的参数配置和潜在风险、输出根据当前正在运行的 workload 行为和特征、输出推荐的参数配置。**Train 模式，**通过用户提供的 benchmark 信息，不断地进行参数校正和 benchmark 执行。通过反复的迭代训练强化学习模型，以便用户后期通过 tune 模式加载该模型进行调优。**Tune 模式，**使用优化算法进行数据库参数调优，支持深度强化学习和全局搜索算法（全局优化算法）。深度强化学习模式要求先运行 train 模式，生成训练后的调优模型，而全局搜索算法则不需要提前进行训练，可直接进行搜索调优。

（二）DB4AI：数据库原生 AI 计算

DB4AI 通过在数据库内集成 AI 算法，令 openGauss 具备数据库原生 AI 计算引擎、模型管理、AI 算子、AI 原生执行计划的能力，为用户提供普惠 AI 技术。不同于传统的 AI 建模流程，DB4AI 建模可以解决数据在各平台的反复流转问题，DB4AI 通过数据库规划最优执行路径，简化开发流程，让开发者更专注于具体业务和模型调优。DB4AI 整体结构如图 19

所示。

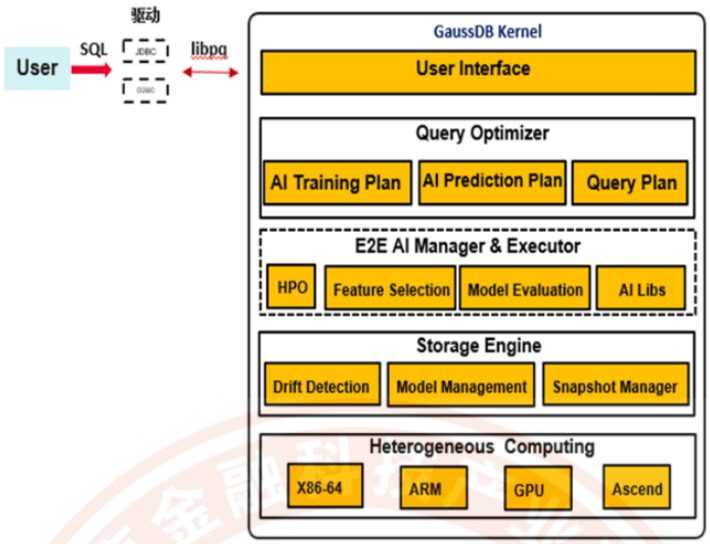


图 19 openGauss DB4AI 整体结构

openGauss 除支持插件式向量数据库引擎，支持原生 DB4AI，还具有以下的特点：

语法：当前主流的计算框架：Tensorflow、pytorch、keras 等大多依托于 python 语言作为构建的脚本语言。openGaus DB4AI 提供了 CREATE MODEL 和 PREDICT BY 两种语法用于完成 AI 的训练和推断任务。该语法相比较 python 更加趋近于自然语言，符合开发者的使用习惯。

数据版本管理：openGauss DB4AI 特性中添加了 snapshot 功能。数据库通过快照的形式将数据集中的数据固定在某个时刻，同时支持保存经过处理过滤的数据。功能支持全量保存和增量保存，其中增量保存每次仅存储数据变化，快照的空间占用大大的降低了。用户可以直接通过不同版本名称的快照直接获取相对应的数据。

性能：openGauss DB4AI 特性通过添加 AI 算子的方式将

模型计算内置到数据库中。以算法训练为例，数据读取、模型的计算更新、最终的模型保存将在数据库的执行器中完成。这种方式更充分地利用和释放数据库的计算能力。基于内核技术的 DB4AI 在计算速度上优于其他更高层级调用的方法，对比数据如图 20 所示。

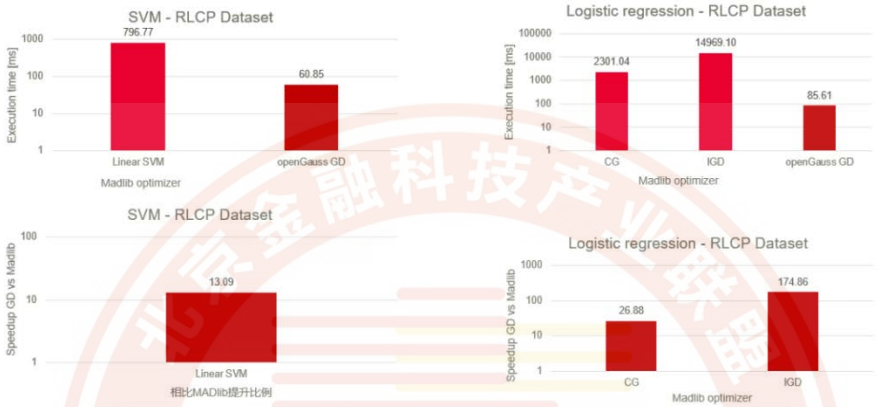


图 20 DB4AI 计算速度对比

六、异构数据库工具研究

MySQL 开源数据库在金融行业应用广泛，openGauss 一方面通过插件化架构，在内核层面提供了 MySQL 引擎插件，覆盖 MySQL 常见语法，另一方面推出了端到端的 MySQL 迁移方案，支持 MySQL 向 openGauss 数据库的平滑迁移。

本章从语法兼容和数据迁移两方面介绍金融行业数据库从 MySQL 向 openGauss 数据库平滑迁移的能力。

（一）语法兼容

openGauss 的 MySQL 兼容性主要通过 Dolphin 内核插件实现。总体架构如图 21 所示。

图 21 MySQL 兼容性实现架构

MySQL 插件在 SQL 引擎上定义数据库扩展点，从关键字、数据类型、常量与宏、函数和操作符、表达式、类型转换、DDL/DML/DCL 语法、存储过程/自定义函数、系统视图等方面兼容 MySQL 数据库。

为了更全面地支持 MySQL 生态，openGauss 实现了对 MySQL 通信协议的兼容，MySQL 协议兼容插件的架构如图 22

所示。

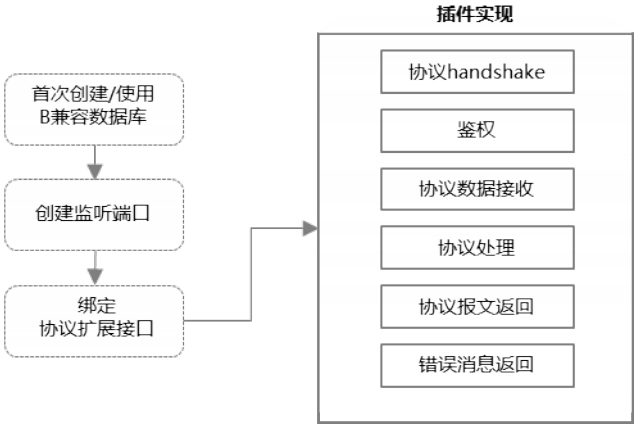


图 22 MySQL 协议兼容插件架构

内核层抽象多数据库协议兼容接口，并以插件方式进行 MySQL 协议开发，实现了 MySQL 协议兼容功能和内核代码的解耦。

（二）数据迁移

1.openGauss 数据迁移

具体包括对象迁移、全量数据迁移、增量数据迁移、全量数据校验、增量数据校验和反向迁移。

对象迁移支持将 MySQL 的数据库对象，包括表、索引、约束、视图、存储过程、函数、触发器等，迁移到 openGauss。

全量数据迁移支持将 MySQL 全量数据并发迁移到 openGauss。

增量数据迁移支持读取 MySQL 侧的 BinLog 并转化成 SQL 语句在 openGauss 侧回放，支持事务级并行回放。

全量数据校验对源 DB 和目标 DB 的全量数据，以表为单位进行校验，找出差异记录，给出修复建议，输出校验报告。

增量数据校验：定期从源数据库日志中提取增量数据，

并对源 DB 和目标 DB 的完整记录进行校验，找出差异记录，给出修复建议，输出校验报告。

反向迁移：将 openGauss 上所有的变更数据实时反向同步至 MySQL，实现在异常情况下业务快速回退。

2.MySQL 迁移

MySQL 迁移工具部署架构如图 23 所示：

图 23 MySQL 迁移工具部署架构

gs_rep_portal 集成了全量迁移、增量迁移、反向迁移、数据校验工具。gs_rep_portal 可设定迁移任务，根据用户设定的执行计划调用相应工具完成迁移步骤，实时展示每个步骤的状态、进度、异常原因等。

全量迁移 gs_mysync 是 MySQL 迁移至 openGauss 的复制工具，支持初始全量数据的复制功能。gs_mysync 通过一次初始化配置，使用只读模式，将 MySQL 的数据全量拉取到 openGauss。支持在同一快照下，表间数据并行迁移。gs_mysync 支持表及表数据、视图、触发器、自定义函数、

存储过程的全量迁移。

增量迁移是指将MySQL数据迁移期间产生的增量数据迁移至openGauss端。Debezium MySQL connector的source端，监控MySQL数据库的binlog日志，并将数据(DDL和DML操作)以Avro格式写入到Kafka;Debezium MySQL connector的sink端,从Kafka读取Avro格式数据(DDL和DML操作)，并组装为事务，在openGauss端按照事务粒度并行回放，从而完成数据(DDL和DML操作)从MySQL在线迁移至openGauss端。由于该方案严格保证事务的顺序性，因此将DDL和DML路由在Kafka的一个topic下，且该topic的分区数只能为1(参数 num.partitions=1)，从而保证 source 端推送到Kafka，和sink端从Kafka拉取数据都是严格保序的。

反向迁移是指将 openGauss 端产生的增量数据迁移至mysql端。Debezium MySQL connector的source端，监控openGauss的xlog日志，并将数据的DML操作以Avro格式写入到Kafka; Debezium MySQL connector的sink端，从Kafka读取Avro格式的数据，在MySQL端按表并行回放，从而完成数据的DML操作从openGauss在线迁移至MySQL。反向迁移可满足用户业务迁移逃生的诉求，保持源端、目标端两个库并行运行，在目标端数据库出问题后应用能及时切回源端数据库。

数据校验工具 gs_datacheck，分为 check 服务和extract服务。check服务用于数据校验，extract服务用于数据抽取和规整。数据校验包括全量校验和增量校验。全量

校验是在全量数据迁移完成后，由 extract 服务对 MySQL 源端和 openGauss 目标端数据通过 JDBC 方式进行数据抽取然后规整计算，并将计算后的中间数据推送到 Kafka 中。最后由 check 服务提取 Kafka 中的中间数据，构建默克尔树，通过默克尔树比对实现表数据校验且输出校验结果。增量校验是由 debezium 服务侦听源端 MySQL 数据库的增量数据，到指定 topic。再由源端 extract 服务处理该 topic 增量数据，触发 check 增量校验。

七、典型案例

中国邮政储蓄银行在全国有近 4 万个营业网点，覆盖全国 99%的县(市)，深度下沉、覆盖广泛、布局均衡，客户数高达 6.5 亿户，客户资产（AUM）超过 10 万亿元。庞大的网络优势和个人客户资源为邮储银行零售业务发展提供了强劲的动力。

邮储银行上一代核心系统于 2014 年投产运行，建成了小型机集群上账户数和交易量最大的银行核心系统，运行期间取得多项成效，但也面临业务和技术上的一系列挑战。随着业务的迅猛扩展，原有核心系统逐步显现出系统容量扩展性有限，针对多元化的业务场景需求响应能力有待提升，关键设备技术与生态迭代缓慢、运行保障形势严峻等问题，核心系统的升级改造迫在眉睫。

中国邮政储蓄银行于 2019 年 3 月启动新一代分布式个

人业务核心系统建设项目。

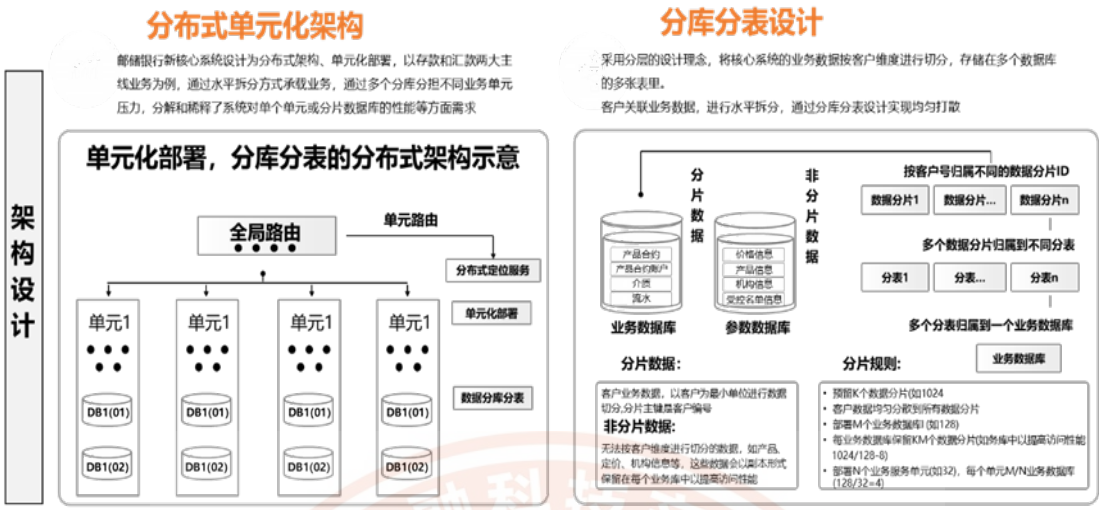


图 24 邮储银行新核心数据库系统架构

邮储银行新一代分布式个人业务核心系统建设历程如下:

- 2021 年 4 月：分布式技术平台投产，新核心主体结构建成。通过旁路验证，持续调优，在性能、稳定性、持续服务、运行管理等方面均达到要求。
- 2021 年 7 月：分布式运维平台投产，新核心的中控室建成。通过支持技术平台运行验证，对新核心运行监测指标进行持续调优，建立并完善了新核心运行维护机制。
- 2021 年 11 月：国际汇款功能投产，核心业务功能投产并营业，通过小量投产，提前演练对客运营服务，类似样板间启用，首批业务功能正式对客服务。
- 2022 年 4 月：核心功能整体投产上线，类似精装完成，落成入驻，个人客户开始逐步在线迁移。
- 2022 年 11 月：全量客户迁移完成，通过在线迁移、无

感切换的新模式，全量个人客户全面迁入并运行于个人新核心系统。

邮储银行新核心建设突破了传统技术架构框架，全面基于分布式、单元化架构进行设计重构，并引入鲲鹏+openGauss 构筑金融核心级分布式架构与技术栈。基于大型银行核心级技术要求，深度打磨国产数据库产品能力，已向社区贡献了近百个关键领域需求与实现，提升了内核性能与稳定性，实现了核心技术自主创新，支持起海量交易处理的大型银行分布式核心系统。

新核心系统业务能力和体验得以大幅增强，具备高峰期每秒 6.7 万笔的交易处理能力，能满足邮储银行超 6.5 亿客户、18 亿账户的服务需要。投产近两年以来运行平稳，各项指标表现优异，全天联机交易处理时间仅需 65 毫秒，比原系统减少 30%；季度结息总时长从 140 分钟减至 35 分钟，降低 75%，可支持邮储未来 10 年业务增长。

八、总结与展望

本报告结合金融行业数据库的关键需求介绍了 openGauss 的能力，包括 openGauss 在数据安全能力、多模多态分布式数据库、内核可观测、数据库 AI 能力和异构数据库迁移工具等。

随着数字化转型的不断深入，金融数据库技术将迎来新的发展浪潮。在这个过程中，数据安全将被置于前所未有的重要位置。金融机构将投入更多资源，研发更为先进的加密技术和访问控制机制，以确保客户数据的绝对安全。同时，

分布式数据库技术也将得到广泛应用，它通过在多个物理节点上存储数据，不仅提高了数据的可访问性和容错性，还显著提升了处理大规模数据集的能力。内核优化作为提升数据库性能的关键，将吸引更多研究者的关注。他们将致力于优化内存管理、减少 I/O 操作、改进查询算法，以实现更高效的数据处理。此外，人工智能技术的融入将使数据库管理更加智能化，AI 不仅可以自动优化查询计划，还能进行预测性维护和智能数据清洗，极大地提高了数据库的运行效率和准确性。异构数据库迁移工具的完善，为 openGauss 等数据库系统带来更大的灵活性和可扩展性。这些工具能够简化不同数据库系统之间的数据迁移过程，降低迁移成本，提高数据整合的效率。随着这些工具的不断进步，openGauss 等数据库将能够更好地适应多样化的业务需求，支持更复杂的数据处理任务，为金融行业提供更加强大和灵活的数据管理解决方案。