GreatDB
万里数据库

技术白皮书 2025

万里分布式数据库管理系统

稳定 · 性能 · 易用

北京万里开源软件有限公司
Beijing Great OpenSource Software Co., Ltd.

目录

1. 文档概述

1.1 文档使用范围

本文针对万里分布式数据库管理系统（产品 GreatDB Cluster V6 版本，以下简称“GreatDB Cluster”）的市场定位和特点，帮助用户从产品的体系架构、组件基本原理、产品功能和应用场景上全面了解本产品。

本文档适合初次接触本产品的用户，用于指导用户从宏观上对 GreatDB Cluster 建立初步的了解和认识。

1.2 术语

本文使用的专用术语、定义，包括通用词语在本文档中的专用解释。术语具体说明参见下表。

表 1-1 术语表

术语	说明
分布式数据库	分布式数据库系统通常使用较小的计算机系统，每台计算机可以单独放在一个地方，每台计算机中都可能有 DBMS 的一份完整拷贝副本，或者部分拷贝副本，并具有自己局部的数据库。位于不同地点的许多计算机通过网络互相连接，共同组成一个完整的、全局的逻辑上集中、物理上分布的大型数据库。
分布式事务	分布式事务，指事务的参与者、支持事务的服务器、资源服务器以及事务管理器分别位于不同的分布式系统的不同节点之上。
原子性	整个事务中的所有操作，要么全部完成，要么全部不完成，不可能停滞在中间某个环节。事务在执行过程中发生错误，会被回滚（Rollback）到事务开始前的状态，就像这个事务从来没有执行过一样。
一致性	在事务开始之前和事务结束以后，数据库的完整性约束没有被破坏。
隔离性	两个事务的执行是互不干扰的，一个事务不可能看到其他事务运行

	中间某一时刻的数据
持久性	在事务完成以后，该事务对数据库所做的更改便持久保存在数据库之中，不会被回滚。

1.3 缩略语

本文使用的专用缩略语以及说明参见下表。

表 1-2 缩略语表

缩略语	英文	中文
ACID	Atomicity、Consistency、Isolation、Durability	ACID，指数据库事务正确执行的四个基本要素的缩写。包含： 原子性（Atomicity）、一致性（Consistency）、隔离性（Isolation）、持久性（Durability）
OLTP	On-Line Transaction Processing	联机事务处理
OLAP	On-Line Analytical Processing	联机分析处理
HTAP	Hybrid Transactional/Analytical Processing	混合事务/分析处理，是一种新兴的数据处理方式，旨在通过在同一数据库系统中同时支持事务处理（OLTP）和分析处理（OLAP）来简化数据处理流程。
SN	ComputeNode，又称 SQLNode	计算节点，用于接收处理应用程序请求，维护集群元数据。一个 SQLNode 对应一个实例
DN	DataNode	存储节点，用于存放用户数据。一个

		存储节点对应一个实例
Shard	shard	分片，用于存放指定条件的用户数据，一个分片由多个存储节点组成一个复制组，同一个复制组内的存储节点包含的数据完全一致
Partition	Partition，又称 sharding	分区，决定数据分布的策略，常用的分布算法有 hash、range、list、columns 等
MVCC	Multiversion concurrency control	多版本并发控制（Multiversion concurrency control， MCC 或 MVCC），是数据库管理系统常用的一种并发控制。 MVCC 意图解决读写锁造成的多个、长时间的读操作饿死写操作问题。每个事务读到的数据项都是一个历史快照，并依赖于实现的隔离级别。写操作不覆盖已有数据项，而是创建一个新的版本，直至所在操作提交时才变为可见。快照隔离使得事务看到它启动时的数据状态。
VIP	Virtual IP	Virtual IP 虚拟 IP 地址。是一个逻辑上的 IP 地址，它不代表任何特定的网络接口或硬件设备，而是用于访问一个或多个服务器的虚拟地址，一般用于网络访问时的负载均衡、故障转移和屏蔽后端高可用切换场景。
Plugin	功能插件（Plugin）	计算机编程和软件工程中，插件（Plugin）或扩展（Extension）是一种特殊的类型软件，它可以被添加到一个已存在的软件应用中，以增加特定的功能或特性。
ADM	GreatADM	本文特指万里数据库一体化数据库管理平台 GreatADM。ADM 支持单机、集中式架构、分布式数据、多机房容灾

		方案的部署、扩缩管理、监控告警、日常巡检分析、和 SQL 性能分析、SQL 开发审核等功能的综合性数据库管理平台。
DTS	GreatDTS	本文特指万里数据库迁移同步工具 GreatDTS。主要支持 Oracle、MySQL、MariaDB、GreatDB Cluster 等数据库双向迁移，异构迁移的兼容性评估、数据全量和增量同步、数据比对校验等功能。
Paxos	Paxos 协议	基于消息传递且具有高度容错特性的一致性算法。
TPC-H	Transaction Processing Performance Council	TPC-H 是业界常用的一套基准，由 TPC 委员会制定发布，用于评测数据库的分析型查询能力。TPC-H 查询包含 8 张数据表、22 条复杂的 SQL 查询，大多数查询包含若干表 Join、子查询和 Group by 聚合等。 SF100，TPC- H 中使用 SF（Scale Factor）来表示数据量规模，SF 1 约对应 1 GB 数据量，SF100 约对应 100GB 数据量。

2. 产品简介

2.1 产品定位

GreatDB Cluster 是一款由万里数据库自主研发的安全、可控、企业级分布式关系型数据库产品，支持多种灵活稳定的金融级高可用方案；提供完备的事务支持，能适用于要求苛刻的在线事务处理（OLTP）应用场景，同时具备轻量级实时数据分析处理（OLAP）能力；完全兼容 SQL92、SQL99、2003SQL 标准，兼容 MySQL 8.0 ，同时支持大部分常用 Oracle 语法；自带丰富的生态组件，支持多种 CPU 芯片架构和 Linux 系统；支持众多金融级安全功能特性。

GreatDB Cluster 利用多个机器共同服务，可以有效解决数据存储容量和访问量

的瓶颈问题。并且可以动态扩容计算和存储节点数目，从而实现关系型数据库横向线性扩展。具备稳定可靠、高性能、高安全、高兼容等特性，目前已经成功应用在金融、运营商、政企、能源等众多行业。

2.2 产品优势

- 高度可扩展性

  软件架构分层设计，各层组件（计算节点、存储节点）均可横向、纵向动态在线扩缩容。

- 灵活的数据拆分技术

  支持哈希、范围、列表、复制多种数据分片规则，可以根据业务数据特征，选择最适合的分片技术把数据分别存储在多个节点中；通过选择合理的数据分片规则，发挥分布式数据库的最佳性能。

- 在线数据重分布

  支持数据的扩容、缩容，能高效地将数据自动地均匀分布到数据库集群内各个分片上；也支持手动定义规则重新分布数据（比如有数据热点时）

- 高可靠性

  整个集群无单点故障，数据多副本，具备故障自动切换保证系统高可用性。完善的数据备份恢复机制，可以满足同城双活和异地灾备等使用场景。

- 强一致性分布式事务

  具备完善的分布式事务处理机制，可保证读写及数据恢复的强一致。

- SQL 兼容性

  高度兼容 MySQL 语法，也兼容常用的 Oracle 的语法。

- 运维和管理智能高效

  提供统一的管理门户和高效的运维工具，实现高效、智能、可视化运维管理。

- 分布式数据备份恢复

  支持在线备份，支持恢复到任意时刻点的数据，可灵活进行数据库备份和恢复。

- 云化支持

  支持 k8s 等云化部署方案，提供 GreatDB Cluster Operator 方案，支持自动化批量部署、自动化运维和在线的实时扩缩容管理。

- 完整的国产化支持

提供丰富的国产化 CPU 支持，包括飞腾、鲲鹏、海光等国产主流芯片；麒麟、统信、欧拉等国产操作系统等国产生态，同时涵盖不同行业上百家国产应用软件和中间件等进行适配认证支持。

- 良好的上下游生态

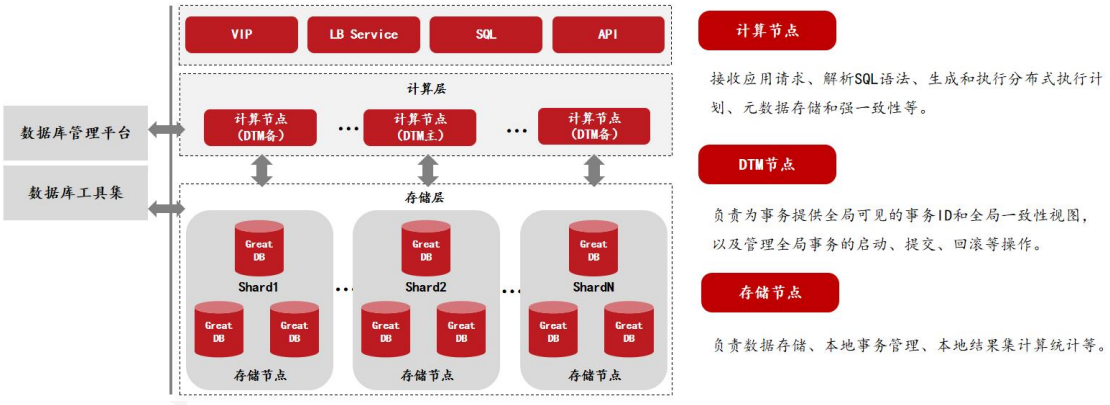
具备良好的生态链完整性，上下游产品能无缝衔接，比如 OGG、Informatica、Kettle。所有的第三方工具、编程语言、客户程序可以像使用 MySQL 数据库一样与 GreatDB Cluster 集群进行完美地交互。

- 自主、安全、可控

完全自主研发，源代码全掌握，安全可控，已在金融、电信、政企、能源行业成功商用，运行稳定、安全、高效。

3. 产品架构

产品按照极致稳定、极致性能、极致易用的理念进行设计，采用 Shared-Nothing 的计算、存储分离分布式架构，具有极强的可扩展性，满足用户按需动态扩缩容的需求；基于经典的 paxos 一致性协议实现数据库多副本的强一致性，满足金融级数据高可用；采用一致性哈希算法，保证数据均衡分布，兼容主流 SQL 标准，对应用开发完全透明，应用平滑迁移；采用了线程池、并行计算、计划下推等多种优化策略，深度适配多种平台，满足了高并发，低响应延迟的场景要求。



GreatDB Cluster 数据库架构包括两部分：

- 计算层：所有的计算节点（也称 SQLNode）组成了可横向扩展的计算层。计算节点用于接收应用请求，解析 SQL 语法，生成执行计划以及执行 SQL 计划。计算节点只存储集群的元数据，计算节点之间通过 paxos 进行元数据同步，保证整个集群元数据的强一致性。除了存储元数据外，还集成了 DTM（Distributed Transaction Manager）分布式事务管理器，负责维护全局事务 ID 的申请，全局事务活跃列表的获取等相关任务。

- 存储层：所有的存储节点（也称 DataNode）组成了可横向扩展的存储层，用于存储真实的业务数据。存储节点接收计算节点的 SQL 请求，给计算层提供数据存储和抽取接口。数据使用不同的分片算法进行分布，一个分片（shard）包含多个存储节点，组成 shard 的所有存储节点保存相同数据副本，副本通过 paxos 协议进行同步，从而保证数据的强一致，同时提升服务的可用性。

4. 产品功能与特性

GreatDB Cluster 是一款适用于 OLTP 场景，同时兼顾 OLAP 的分布式关系型数据库。

GreatDB Cluster 完全兼容 MySQL8.0 语法，兼容 SQL92、99、2003 、部分 2016 标准，同时兼容部分 Oracle 语法，包括 数据定义语言 (DDL）、数据操纵语言（DML）、数据查询语言（DQL）、数据库管理语句、事务控制语言（TCL）、内置函数、程序控制语言、批处理语法等。

GreatDB Cluster 对跨节点的复杂 SQL 操作支持全面，使得业务人员的开发工作量大幅降低，无需考虑大量的 SQL 改造。通过提供良好的 SQL 兼容性和可维护性，使用户可以像使用 MySQL 数据库一样使用分布式数据库，同时又能解除 MySQL 数据库不可扩展带来的担忧。

在分布式数据库集群中节点数目和种类很多，为了方便管理，GreatDB Cluster 也提供统一图形运维界面,方便用户对数据库集群进行操作和管理。同时也提供相关工具，实现数据顺利地迁移到 GreatDB Cluster。

4.1 基础功能

GreatDB Cluster 高度兼容 MySQL 语法，也提供标准的数据库接口 ODBC、JDBC、OLEDB、ADO.NET、PDO 等常用接口，同时支持各种开发语言和工具，如 JAVA、C/C++、

C#、PHP、Python、GO 等。兼容 MySQL 工具生态圈，如 BenchmarkSQL、Sysbench、NaviCat、DBVisualizer 等。

除了通用语法类型外，GreatDB Cluster 还有扩展语法,主要包括两类：

- 第一类建表语法的扩展，如表分片信息定义；第二类基于原生语法增强，增加对常用 Oracle 语法支持，方便业务使用。如 Oracle 的 Sequence 序列语法，使得业务开发者的语法使用习惯得以保留，另一方面也使得基于 Oracle 数据库开发的业务迁移变得轻松。

4.1.1 数据类型

GreatDB Cluster 提供丰富的数据类型支持，参考如下：

数值类型

类型	描述
BIT	BIT[(M)].存储 1 至 64 个比特位。未指定 M 时默认长度为 1。 占用存储字节数随长度变化而不同。
BOOL | BOOLEAN	布尔类型，内部实现等同 TINY(1）。 注意：2 = TRUE 返回结果是 false
TINYINT	TINYINT[(M)].取值范围：有符号[-128, 127]，无符号[0, 255]。 占用存储字节数：1 Byte。可通过 M 限制位数。
SMALLINT	SMALLINT[(M)].取值范围：有符号[-32768, 32767]，无符号[0, 65535] 占用存储字节数：2 Bytes。可通过 M 限制位数。
MEDIUMINT	MEDIUMINT[(M)].取值范围：有符号[-8388608, 8388607]，无符号[0, 16777215] 占用存储字节数：3 Bytes。可通过 M 限制位数。
INT | INTEGER	INT[(M)].取值范围：有符号[-2147483648, 2147483647]，无符号[0, 4294967295] 占用存储字节数：4 Bytes。可通过 M 限制位数。
BIGINT	BIGINT[(M)].取值范围：有符号[-2^63, 2^63 - 1]，无符号[0, 2^64 - 1] 占用存储字节数：8 Bytes。可通过 M 限制位数。
DECIMAL | DEC	DECIMAL[(M[, D)]].定点数据类型，类似 Oracle 的 NUMBER。 M 为总位数，默认 10，最大 65。D 为小数点后位数，默认

类型	描述
	0，最大 30。 注意：两个定点数运算时基于 65 位计算，需考虑溢出问题。 占用存储字节数：
FLOAT 不建议使用	FLOAT[(M, D)] 该格式会在新版本取消。 FLOAT[(p)] 当 p 在 0-24 时，类型为 FLOAT；当 p 在 25—53 时，类型隐式变为 DOUBLE。 占用存储字节数：4 Bytes
DOUBLE 不建议使用	取值范围：0， [-1.7976931348623157E+308, -2.2250738585072014E-308]， [2.2250738585072014E-308, 1.7976931348623157E+308] 占用存储字节数：8 Bytes

- **日期和时间类型**

类型	描述
DATETIME	DATETIME[(fsp)].fsp 表示秒值的小数位数。 取值范围：['1000-01-01 00:00:00.000000', '9999-12-31 23:59:59.499999'] 可通过 DEFAULT，ON UPDATE CURRENT_TIMESTAMP 设置默认值或自动更新 占用存储字节数：5 - 8 Bytes，随 fsp 值变化。
TIMESTAMP	DATETIME[(fsp)].fsp 表示秒值的小数位数。 取值范围：UTC 时间['1970-01-01 00:00:01.000000', '2038-01-19 03:14:07.499999'] 可通过 DEFAULT，ON UPDATE CURRENT_TIMESTAMP 设置默认值或自动更新 占用存储字节数：4 - 7 Bytes，随 fsp 值变化。
TIME	TIME[(fsp)].fsp 表示秒值的小数位数。 取值范围：['-838:59:59.000000', '838:59:59.000000'] 占用存储字节数：3 - 6 Bytes，随 fsp 值变化。
DATE	取值范围：['1000-01-01', '9999-12-31'] 占用存储字节数：3 Bytes
YEAR	取值范围：['1901', '2155'] 占用存储字节数：1 Byte

- **字符串类型**

类型	描述
CHAR	CHAR[(M)].定长字符串，M 表示字段长度（字符数），末尾以空格填充。 M 取值范围[0, 255]，未指定 M 时默认为 1。 占用存储字节数：M*字符集单字符最大占用字节数
VARCHAR	VARCHAR[(M)].变长字符串，M 表示字段长度（字符数）。 M 取值范围[0, 64k]，同时受 innodb max row size=64k 限制。 占用存储字节数：长度标识+ 实际数据编码字节数（最大占用>255 字节时长度标识占 2 字节）
BINARY	BINARY[(M)].定长二进制字符串，M 表示字段长度（字节数），末尾以空格填充。 M 取值范围[0, 255]，未指定 M 时默认为 1。与 CHAR 类似 占用存储字节数：M
VARBINARY	VARBINARY[(M)].变长二进制字符串，M 表示字段长度（字节数）。 M 取值范围[0, 64k]，同时受 innodb max row size=64k 限制。 与 VARCHAR 类似 占用存储字节数：长度标识+ M（M>255 时长度标识占 2 字节）
TINYTEXT	文本类型。最大占用 256 字节空间。 最大可存储字符数受字符集单字符占用字节数影响。
TEXT	文本类型。最大占用 64k 字节空间。 最大可存储字符数受字符集单字符占用字节数影响。
MEDIUMTEXT	文本类型。最大占用 16M 字节空间。 最大可存储字符数受字符集单字符占用字节数影响。
LONGTEXT	文本类型。最大占用 4G 字节空间。 最大可存储字符数受字符集单字符占用字节数影响。
TINYBLOB	文本类型。最大占用 256 字节空间。类似 TINYTEXT。
BLOB	文本类型。最大占用 64k 字节空间。类似 TEXT。
MEDIUMBLOB	文本类型。最大占用 16M 字节空间。类似 MEDIUMTEXT。
LONGBLOB	文本类型。最大占用 4G 字节空间。类似 LONGTEXT。
ENUM	字符串枚举类型。内部以整数值存储，最多允许 65535 个枚举值定义。 单个枚举值最大字符数 255。可为 NULL，空字符串’’表示错误值。

类型	描述
SET	字符串集合。内部以整数值存储（以 bit 位标识），最多允许 64 个成员定义。 单个成员最大字符数 255。

- **空间类型**

类型	描述
GEOMETRY	所有空间类型的基类，可以存储点、线、面、几何图形任何一种类型的对象，通用类型。当一列需要存储多种不确定的几何图形时使用（不推荐使用，不利于优化）。
POINT	代表二维空间中的一个点，由一对坐标 (X, Y) 或 （经度，纬度） 定义
LINESTRING	由两个或更多点连接而成的序列，代表一条线段
POLYGON	一个多边形区域类型
MULTIPOINT	点集合，包含多个 POINT 对象
MULTILINESTRING	多条线集合，包含多个 LINESTRING 对象
MULTIPOLYGON	多边形集合，包含多个 POLYGON 对象
GEOMETRYCOLLECTION	几何对象集合，可以包含任意类型和数量的其他几何对象
JSON	JSON 类型，存储 JSON（JavaScript Object Notation）格式的文档数据

- **Oracle 数据类型兼容**

类型	描述	与 Oracle 原生差异
CLOB	文本类型，为 TEXT 类型同名。	Oracle 最大可容纳 2G-1 个字符
NUMBER	定点数值类型，为 DECIMAL 同名。	Oracle 默认长度、最大精度不同， 转换字符串时小数点后的 0 被

类型	描述	与 Oracle 原生差异
		舍弃
PLS_INTEGER	INT 类型别名	/
VARCHAR2	VARCHAR 类型别名	最大长度 4000

字段属性

属性名	说明	使用限制
AUTO_INCREMENT	自增列，插入 NULL 时自增长	1. 表只允许有一个自增列 2. 在整值和 DECIMAL 上使用
CHARACTER SET	字符集	用于字符列类型：CHAR,VARCHAR,TEXT,ENUM，SET
COLLATE	字符集比较规则	同上
COMMENT 'str'	列注释	
DEFAULT {literal | (expr)}	默认值	表达式外部需要有括号
[GENERATED ALWAYS] AS (expr)	用于指定 generated 列	表达式外部需要有括号
VIRTUAL | STORED	是存储 generated 列的值	generated 列创建索引需 STORED
NULL | NOT NULL	未指定时默认为 NULL	仅 innodb、myisam、memory 表支持 在 NULL 属性的列上建索引
VISIBLE | INVISIBLE	字段可见性，默认 VISIBLE	表至少包含一个 VISIBLE 列

- **索引属性**

属性名	说明	使用限制

属性名	说明	使用限制
FOREIGN KEY	外建	Innodb 分区表不支持
FULLTEXT	全文索引	仅 innodb 和 myisam 支持
KEY | INDEX	普通索引	
PRIMARY KEY	主键	Innodb 有最大长度限制
SPATIAL	空间索引	仅 innodb 和 myisam 支持
UNIQUE	唯一约束	

- **运算符支持**

类别	包含运算符
算术运算	+, -, *, /, %, MOD
位运算	&, ^, |, ~, >>, <<
比较运算	>, >=, <, <=, <>, !=, <=>, =, IS NULL, IS NOT NULL BETWEEN AND, IN, NOT IN, LIKE, REGEXP 等
赋值	:=, =
逻辑运算	AND, && NOT, ! OR, || XOR

- **运算符优先级(由高到低)**

优先级	运算符	备注
1	!	逻辑非
2	-, ~	负号和位反转

优先级	运算符	备注
3	^	位异或
4	*, /, %	乘、除、取模运算
5	+, -	加、减法运算
6	>>, <<	移位运算
7	&	位与
8	|	位或
9	>, >=, <, <=, <>, !=, <=>, = IS, LIKE, REGEXP, IN 等	比较运算
10	BETWEEN AND	比较运算
11	NOT	逻辑非
12	AND, &&	逻辑与
13	XOR	逻辑异或
14	OR, ||	逻辑或
15	:=, =	赋值运算

4.1.2 内置函数

内置函数包括但不限于：

类别	函数

类别	函数
控制流函数	CASE , IF() , IFNULL() , NULLOF()
数学函数	ABS() , ACOS() ASIN() , ATAN() , ATAN2() , CEIL() , CEILING() , CONV() ,COS() , COT() , CRC32() , DEGREES() , EXP() , FLOOR() , LN() , LOG() ,LOG10() , LOG2() , MOD() , PI() , POW() , POWER() , RADIANS() , RAND() ,ROUND() , SIGN() , SIN() , SQRT() , TAN() , TRUNCATE()
时间日期函数	ADDDATE() , ADDTIME() , CONVERT_TZ() , CURDATE() , CURRENT_DATE() ,CURRENT_TIME() , CURRENT_TIMESTAMP() , CURTIME() ,DATE(),DATE_ADD(),DATE_FORMAT() , DATE_SUB() , DATEDIFF() , DAY() , DAYNAME() , DAYOFMONTH() ,DAYOFWEEK() ,DAYOFYEAR() , EXTRACT() , FROM_DAYS() , FROM_UNIXTIME() ,GET_FORMAT() , HOUR() , LAST_DAY Re , LOCALTIME() , LOCALTIMESTAMP, MAKEDATE() , MAKETIME() ,MICROSECOND() , MINUTE() , MONTH() , MONTHNAME() ,NOW() , PERIOD_ADD() , PERIOD_DIFF() , QUARTER() , SEC_TO_TIME() , SECOND() ,STR_TO_DATE() , SUBDATE() , SUBTIME() , SYSDATE() , TIME() , TIME_FORMAT() ,TIME_TO_SEC() , TIMEDIFF() , TIMESTAMP() , TIMESTAMPADD() , TIMESTAMPDIFF() ,TO_DAYS() , TO_SECONDS() , UNIX_TIMESTAMP() , UTC_DATE() , UTC_TIME() ,UTC_TIMESTAMP() , WEEK() , WEEKDAY() , WEEKOFYEAR() , YEAR() , YEARWEEK()
字符串函数	ASCII() , BIN() , BIT_LENGTH() , CHAR() , CHAR_LENGTH() , CHARACTER_LENGTH() ,CONCAT() , CONCAT_WS() , ELT() , EXPORT_SET() , FIELD() , FIND_IN_SET() ,FORMAT() , FROM_BASE64() , HEX() , INSERT() , INSTR() , LCASE() , LEFT() ,LENGTH() , LIKE , LOAD_FILE() , LOCATE() , LOWER() , LPAD() , LTRIM() ,MAKE_SET() , MATCH , MID() , NOT LIKE , NOT REGEXP , OCT() , OCTET_LENGTH() ,ORD() , POSITION() , QUOTE() , REGEXP , REGEXP_INSTR() , REGEXP_LIKE() ,REGEXP_REPLACE() , REGEXP_SUBSTR() , REPEAT() , REPLACE() , REVERSE() ,RIGHT() , RLIKE , RPAD() , RTRIM() , SOUNDEX() , SOUNDS LIKE , SPACE() ,STRCMP() , SUBSTR() , SUBSTRING() , SUBSTRING_INDEX() , TO_BASE64() , TRIM() ,UCASE() , UNHEX() , UPPER() , WEIGHT_STRING()
类型转换函	BIANRY , CAST() , CONVERT，兼容 Oracle 类型转换函数

类别	函数
数	TO_CHAR(),TO_NUMBER(),TO_DATE(),TO_CLOB(),TO_TIMESTAMP()
聚集函数	AVG() , BIT_AND() , BIT_OR() , BIT_XOR() , COUNT() , COUNT(DISTINCT) , GROUP_CONCAT() ,JSON_ARRAYAGG() , JSON_OBJECTAGG() , MAX() , MIN() , STD() , STDDEV() ,STDDEV_POP() , STDDEV_SAMP() , SUM() , VAR_POP() , VAR_SAMP() , VARIANCE()
窗口函数	CUME_DIST() , DENSE_RANK() , FIRST_VALUE() , LAG() , LAST_VALUE() , LEAD() ,NTH_VALUE() , NTILE() , PERCENT_RANK() , RANK() , ROW_NUMBER(),LISTAGG()
加解密函数	AES_DECRYPT() , AES_ENCRYPT() , COMPRESS() , MD5() ,RANDOM_BYTES() , SHA1() ,SHA2() , STATEMENT_DIGEST() , STATEMENT_DIGEST_TEXT() , UNCOMPRESS() ,UNCOMPRESSED_LENGTH() , VALIDATE_PASSWORD_STRENGTH()
位运算函数	BIT_COUNT
辅助类函数	ANY_VALUE() , BIN_TO_UUID() , DEFAULT() , GROUPING() , INET_ATON() ,NET_NTOA() ,INET6_ATON(),INET6_NTOA(),IS_IPV4(),IS_IPV4_COMPAT(),IS_IPV4_MAPPED() , IS_IPV6() , IS_UUID() , MASTER_POS_WAIT() , NAME_CONST() ,SLEEP(),UUID(),UUID_SHORT() , UUID_TO_BIN() , VALUES()
其他	支持用户自定义函数
兼容 Oracle 函数	ADD_MONTHS() ,CHR(),DECODE(),DUMP(),INSTR(),INSTRB() ,LENGTH() ,LENGTHB() ,LTRIM() ,LPAD() ,RPAD() ,MONTHS_BETWEEN() , NVL() ,NVL2() ,REGEXP_COUNT() ,TRIM(),LTRIM() ,RTRIM() ,SUBSTR() ,SUBSTRB() ,SYSDATE() ,SYSTIMESTAMP() ,TO_CHAR() ,TO_DATE() ,TO_NUMBER() ,TO_TIMESTAMP() ,TRANSLATE() ,TRIM() ,TRUNC(date) ,TRUNC(nunber) ,VSIZE() ,WM_CONCAT() （兼容业务

类别	函数
	中常用 Oracle 函数、对象等较多，不在此逐一罗列）

4.1.3 索引

支持 B+树索引；拥有多种索引类型包括：唯一索引、复合索引、函数索引。可以通过使用索引快速定位数据，提高查询性能。

4.1.4 过程

支持存储过程功能，严格地说包含过程（PROCEDURE）和函数（FUNCTION），分别使用 CREATE PROCEDURE 和 CREATE FUNCTION 语句来创建。

4.1.5 触发器

提供完备的触发器功能，支持插入触发器、更新触发器、删除触发器等触发器类型。触发器支持存储过程所支持的一切特性；

触发时间支持操作前触发（BEFORE）与操作后(AFTER)触发；

触发事件支持插入触发（INSERT），更新触发（UPDATE）与删除（DELETE）触发。

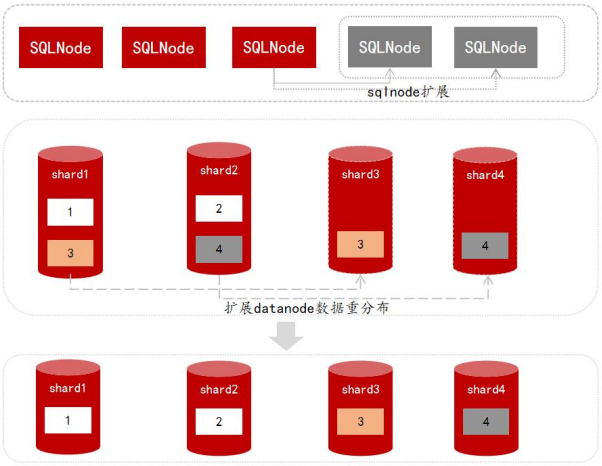
4.1.6 视图

支持视图功能。通过视图保存复杂查询，简化用户操作；提供单表和多表视图的支持。支持创建视图、删除视图、修改视图等视图基本操作。

4.2 高级功能

4.2.1 可扩展性

GreatDB Cluster 支持在线按需横向扩展功能



- 计算节点不存储业务数据，是轻量节点，可快速进行横向扩展以增强集群计算能力。新扩增的计算节点会通过 paxos 自动同步集群元数据，待完全同步完成后，可对外提供服务。
- 存储节点主要负责存储业务数据，在线扩容存储节点会触发数据自动重分布，实现集群存储能力的扩容。存储节点扩容以分片为单位，并保证足够的高可用。

性能可以随计算节点和存储节点的增加线性扩展，性能损耗控制在 10%内。若业务为 CPU 密集型，则计算节点数目要适当增加；若业务为 IO 密集型，存储节点数目随着数据量按比例增加。

4.2.2 高可用

GreatDB Cluster 采用全组件冗余架构，任意组件故障不会影响集群的可用性，高可用达到 99.99%。

集群所有的组件都基于 paxos 保证元数据与业务数据的强一致，任一组件都至少部署 3 个节点，在某台机器出现故障的时候，自动进行故障转移，任意故障场景 RPO=0，确保数据零丢失。

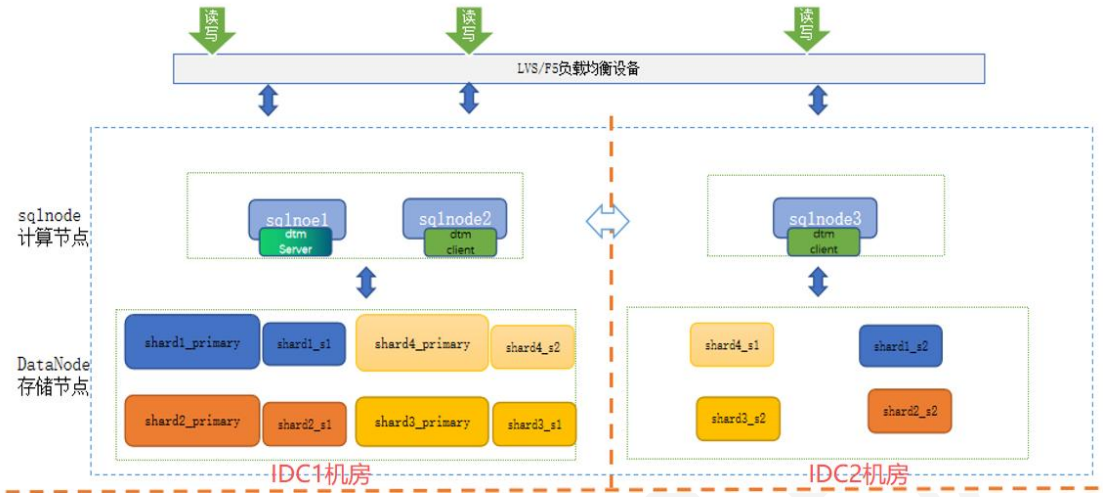
GreatDB Cluster 支持两地三中心等部署架构，拥有同城双活、异地双活和灾备的高可用能力，可以构建去中心化的跨机房对等部署集群，实现四个九级别的容灾能力。

同城双中心部署：依靠自身的强大集群功能中的地理标签功能，可实现实时数据同步，保证同城主备机房间数据一致性。主备机房通常需要保证网络的稳定性和速度，一般来说网络延迟<3 毫秒。

单个节点故障：任何单个节点故障，由于 paxos 多数派协议，都不会造成数据丢失，即 RPO 等于 0。如果 shard 从节点发生故障，不影响集群使用；如果 shard 主节点发生故障，paxos 会通过多数派选举协议，自动选择新的主节点，并实现 RTO＜60s。

机房 2 故障：机房 1 中包含多数节点，因此，机房 2 整体故障的情况下，依旧可以保证数据不丢失，并且不影响集群正常使用。

机房 1 故障：由于机房 1 故障，机房 2 少数副本，不满足多数派协议，集群服务受影响。在机房 1 恢复前需要重置集群状态，确保数据库集群可用。

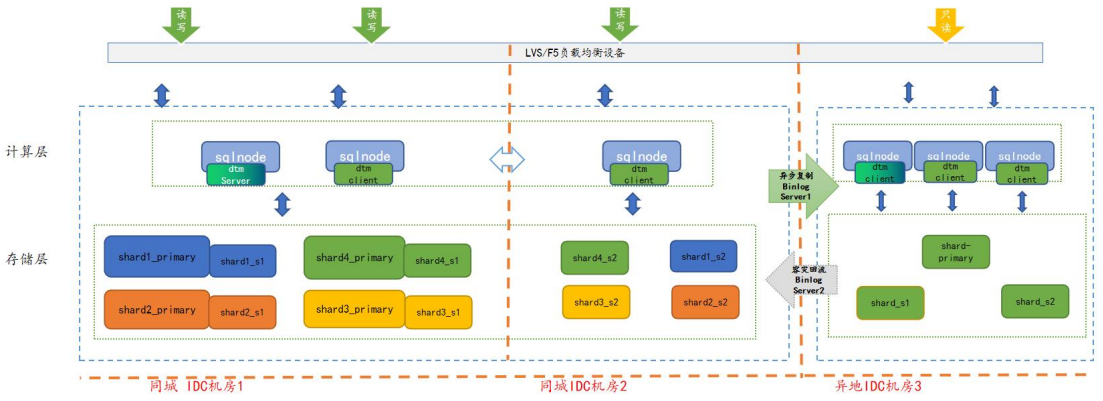


GreatDB Cluster 同城双中心部署

两地三中心部署：部署方案在机房 1 和机房 2 部署单一大集群，在灾备机房 3 部署小集群，并通过 binlog server1 实现主集群到机房 3 的增量数据同步。

故障转移方案：机房 2 故障转移方案与同城 2 机房部署的转移原理一致；当机房 1 和机房 2 都发生故障后，需要业务切换到灾备机房并访问数据库，从而实现灾备机房 3 能接管数据库服务。

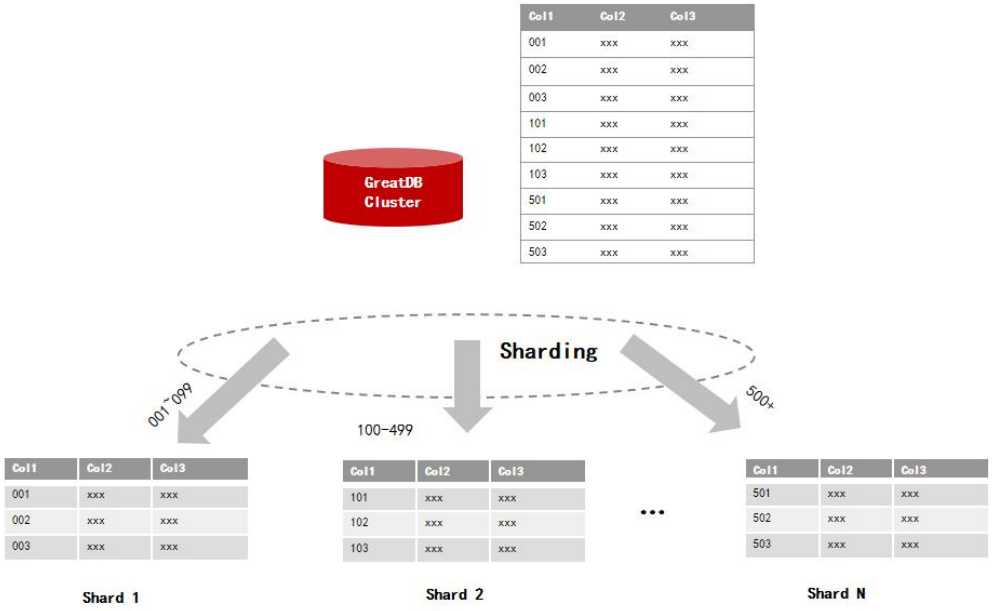
恢复方案：当机房 1 和机房 2 恢复后，启动备用 binlog server2，并把故障期间机房 3 的增量数据同步到本地大集群。等数据验证完成后，把业务在切回到本地大集群，从而恢复正常运行。



GreatDB Cluster 两地三中心容灾部署

4.2.3 灵活的数据分布策略

GreatDB Cluster 支持如下多种类型的表分布（Sharding）策略：



GreatDB Cluster 数据分片示例

普通表（Normal）：建表时会根据内部路由算法，自动表随机分布在任一 shard 分片中，也可根据业务诉求将指定 shard 分片。

全局表（Global）：也叫广播表，默认分布在所有的 shard 上，每个 shard 上拥有全量的表数据，建表时需在建表语句中使用 Create table tab ... disttype=global，一般用作数据量较少的配置表，或高频 Join 的驱动小表 。

分区表（Partition）：支持常见分区表的创建和子分区，语法完全兼容 MySQL8.0 分区表建表语法，无自定义分片关键词（注意：创建分区表时分区字段须包含主键或唯一约束中的所有字段）支持类型如下：

- 哈希分区（Hash）： 数据按照分区字段的 hash 值，分布在不同的 shard 上，每个 shard 上存储部分表数据，不同分片数据之间没有交集。
- 范围分区（Range）：数据按照分区字段的范围，分布在不同的 shard 上，每个 shard 上存储部分分片数据，不同分片数据之间没有交集。
- 列表分区（List）：数据按照分区字段的列表，分布在不同的 shard 上，每个 shard 上存储部分分片数据，不同分片数据之间没有交集。当确定了数据分布策略，数据会自动均衡的分布到存储节点上，支持手动调整分布策略。
- 其他分区：如 key 分区、Columns 分区，支持复合分区的创建和按需组合如 Range+hash/key、Range+list 等。

表分布可遵循以下原则进行设计：

- 全局表

适用场景为数据基本不变的属性表或者维度表，它经常与分区表进行关联查询。

- 分区表

用户通过定义分区键（指生成拆分规则的表字段）可以选择贴合业务数据特点的拆分策略，让在线事务型数据库操作尽可能在高并发的场景下保持低延迟。所以使用如何选择拆分字段成为数据库表结构设计的重要步骤，选择分区类型可以采取如下原则：

前台数据业务：这类业务大部分操作围绕某一个业务主体展开，例如互联网业务典型的是按照用户展开业务操作，物联网是围绕设备、车辆等展开业务操作，银行政府机构柜面类业务围绕客户展开业务操作，电商 ISV、餐饮 ISV 围绕商家展开业务操作等。这类业务数据可以围绕的业务主体进行拆分，能够很好地解决业务超大数据量、高并发、低延迟数据库使用需求。

后台类业务： 按条件组合过滤出一批数据分页展示，并且处理数据后写回数据库。这时可能存在大量单表和多表关联，多种过滤条件组合删选，多表事务处理等。这类场景首要拆分方式还是推荐数据主体拆分。如果数据库处理和时间紧密相关，也可以按时间拆分。

4.2.4 数据重分布

业务的增长不可避免地需要对资源进行扩容，由于使用了分片技术，数据被切分成细小的分片分布在存储节点集群中。集群扩容后，原有的数据分片就面临着被打散重新分配的过程，这个过程就是数据重分布（ReSharding）。

数据重分布功能采用后台数据迁移和日志回放相结合的方法，通过智能过滤，减少数据迁移量，并通过多表并发提升迁移效率，通过极限逼近和循环回放技术，减少新旧数据切换时间，实现不停机的在线数据重分布，达到数据库扩容、缩容对业务无感知。

数据重分布对在线业务影响小且可操作性强，具备如下特点：

- 执行时间可控制，对业务影响秒级。

从前面的原理介绍中可以看出，重分布过程中数据的迁出迁入是个循序渐进的过程，仅在原分片和目标分片数据存在少量差距时才会禁写，只读锁的切换时间〈=1s，因此对在线写业务影响小，不影响业务数据查询。

- 手动重分布

针对某些业务场景,比如存在数据热点问题，可以指定某个或某些数据分配到指

定的存储节点上。

- 可操作性好

提供了存储过程和视图，可以通过 SQL 语句的方式完成操作，全程状态可观测和可控制，包括状态监控、执行、取消等。也可用图形操作界面完成上述功能。

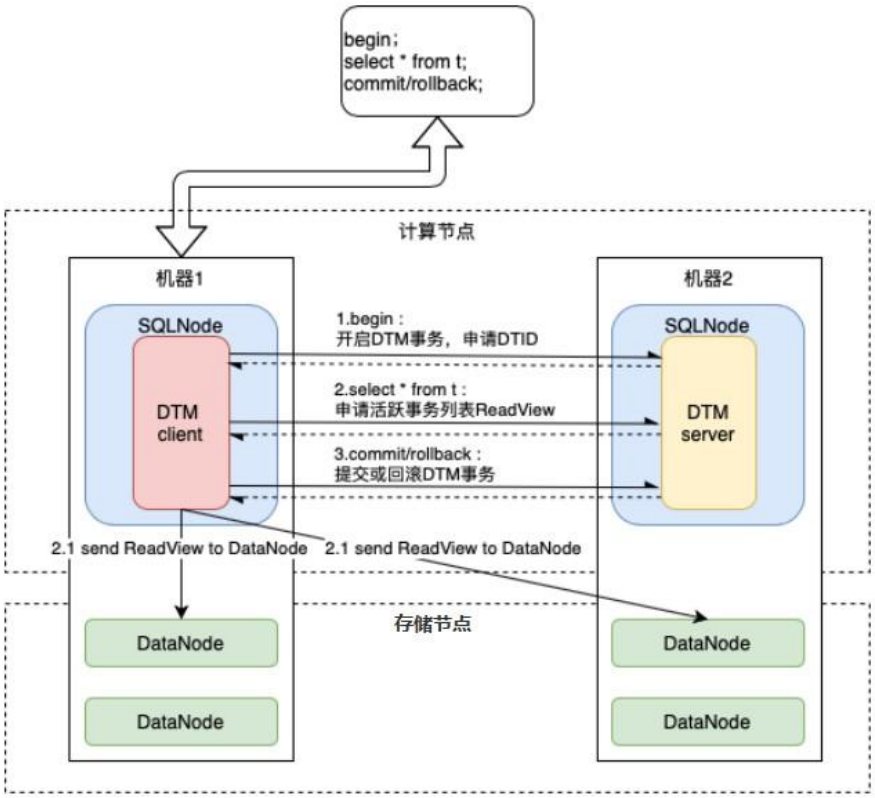
- 产品通用性好

对比需要预先做好分片规划的预 Sharding 方案，重分布方案对设计人员的要求相对较低，支持任意分片策略间的重分布，彻底与业务模型解耦，无需在系统设计初期就要精确规划好未来的数据分布情况。

4.2.5 分布式事务

GreatDB Cluster 提供分布式事务支持，通过全局事务管理器(DTM)、分布式事务、MVCC 多版本控制、行级锁等技术实现完整的分布式事务的能力。确保集群事务操作 ACID 完整性。支持读已提交(RC)、可重复读(RR)两种隔离级别，实现高效的读写访问。同时 DTM 具备高可用特性，可以在计算层中进行快速故障转移。

一个简单的 DTM 事务中 DTM 相关的交互如下图所示



4.2.6 语法扩展和增强

Oracle 有些非标语法的确会方便业务开发，同时目前大量现存业务是基于 Oracle 开发的，如果将这部分业务迁移至 GreatDB Cluster，业务或多或少会涉及改造工作，GreatDB Cluster 将常用的 Oracle 语法支持起来可以降低业务迁移和应用修改的工作量。

进行相应的语法扩展和增强，包括但不限于：

类别	对象
扩展对象类型	BLOB ,CLOB ,NUMBER ,PLS_INTEGER ,VARCHAR2 ,SEQUENCE ,DBLINK
扩展语法	START WITH ...CONNECT BY ,ROWNUM ,CAST(expr AS VARCHAR(n)) ,DATETIME INTERVAL ,DATETIME 运算 ,EXECUTE IMMEDIATE ,EXPLAIN PLAN FOR ,INSERT ALL INTO ,MERGE INTO ,MINUS , Oracle hint ,Oracle (+) 外连接
扩展存储对象语法	CREATE OR REPLACE PROCEDURE 创建或替换存储过程 ,CURSOR ,EXIT/EXIT WHEN ,FOR LOOP ,异常处理 EXCEPTION HANDLER ,IF .. ELSIF 支持 ,WHILE...LOOP... END LOOP ,创建无参数存储过程/函数时支持不带括号 ,存储过程/函数支持默认参数(DEFAULT) ,存储过程支持使用 RETURN ,匿名存储块 ,命名标记法传递参数
PACKAGE 支持	CREATE PACKAGE / CREATE PACKAGE BODY
内置 DBMS 包	DBMS_OUTPUT ,DBMS_UTILITY ,DBMS_LOB ,DBMS_METADATA ,DBMS_RANDOM ,DBMS_SQL ,DBMS_CRYPTO ,DBMS_LOCK
OCI 接口类	OCIEnvCreate, OCIServerAttach, OCIStmtExecute, OCIDefineByPos, OCIAttrSet, OCIAttrGet, OCISessionBegin, OCISessionEndOCIStmtPrepare, OCIStmtFetch, OCILogon, OCITransStartOCITransDetach, OCITransCommit, OCITransRollback, OCIErrorGet, OCILogoff, OCIHandleAlloc, OCIHandleFree, OCIBindByPos, OCIBindByName, OCIDescriptorAlloc, OCIDescriptorFree，OCILobLocatorIsInit, OCILobWrite, OCILobRead, OCILobCreateTemporary，

类别	对象
	OCIDateTimeGetTimeZoneOffset, OCIDateTimeConstruct, OCIDateTimeGetDate, OCIDateTimeGetTime, OCIPing

4.2.7 数据迁移

数据迁移包括迁入和迁出，迁入是指把一种数据源的数据(比如 Oracle/MySQL 等)迁移到 GreatDB Cluster 中，迁出是指把 GreatDB Cluster 系统的数据同步到另外的系统中。

- GreatDTS 异构迁移同步工具

GreatDTS 是万里开源自主研发的数据库迁移工具，方便用户迁移其他数据库到 GreatDB Cluster 数据库，实现应用评估、对象迁移和转换、数据迁移和同步、对象和数据校验等一站式数据迁移服务，也支持并行执行、断点续传等功能。GreatDTS 具有如下优势与特点。

1. 快速部署

   可以使用 Docker 方式快速部署，降低软件部署复杂度，保障资源隔离，便于维护。

2. 稳定高效

   支持断点续传，可有效解决大数据量迁移网络、系统等异常导致的传输中断。

3. 安全精准

支持全量数据迁移，自动对迁移数据进行完整性校验，支持加密数据传输，保障数据安全。

4. 简单易用

快捷安装，简单易用的引导模式，高迁移效率。迁移过程中，可以控制迁移任务暂停、继续执行和取消。

5. 功能全面

支持源数据库对象分析、应用评估、兼容性验证、高并行迁移和数据校验，有效协助业务系统实施平滑迁移。

- 原始文件并行导入数据库集群

原始文件可以是 sql 文件，csv 文件等。GreatDB Cluster 能自动根据导入表的分片规则，自动对原始数据文件记录分拣，将数据下发至对应存储节点，最终完成导入。其中的各环节均支持并发处理，提升整体导入效率。

- MySQL 数据实时同步到 GreatDB Cluster

GreatDB Cluster支持用户将 MySQL 数据库的实时同步到 GreatDB Cluster，方便用户随时切换到 GreatDB Cluster 中。

- 全局一致性数据导出

GreatDB Cluster支持在分布式数据库中以全局一致的状态把当前时间点的数据全量或局部导出为 SQL 文件。这对于需要数据强一致的业务有重要意义，如金融行业年终的账期数据。

- 分布式 BinlogServer(CDC 变化数据捕获技术)

分布式数据 CDC (变化数据捕获)是以独立部署服务器实现，CDC Server 用来收集分布式集群每个 Shard 的 Binlog 日志，把零散的分布式日志再次整合为一个 Binlog，并转换为兼容 MySQL Binlog 格式。这样可以把 GreatDB Cluster 数据同步到其他系统中，比如大数据系统。

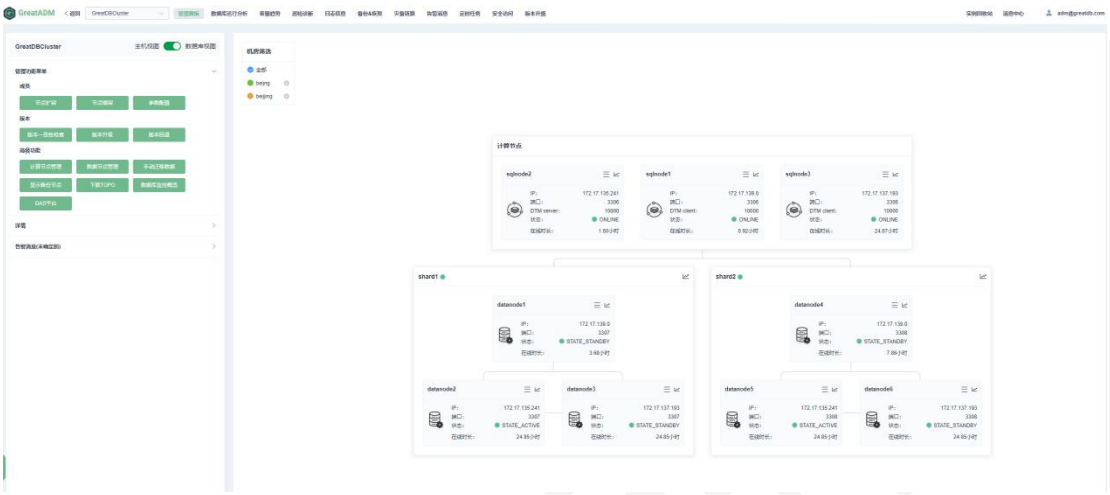
4.2.8 压缩功能

支持数据压缩。表数据压缩是通过对表及分区表进行数据压缩来减少磁盘存储空间，在进行压缩操作前，可以对压缩率进行提前估算，从而做到有针对性的有效压缩，压缩算法采用 zlib 算法，压缩比在 2~11 倍之间。

4.3 数据库管理能力

GreatDB Cluster 内置了一套存储过程和视图，用户可以通过 sql 语句的方式统一监控和管理整个分布式数据库；通过公司的另外一个产品 GreatRDS，它能提供简单、易用图形化管理工具，实现了物理机、虚拟机等资源池化功能，支持对 GreatDB

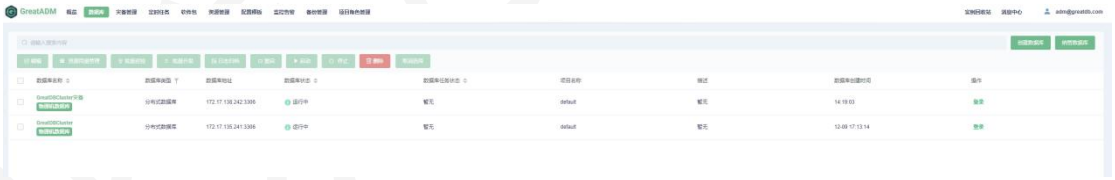
Cluster，MySQL 等多种数据库的集中管理，通过分布式部署，支持对批量数据库进行统一监控部署和管理，同时具备丰富的接口功能，方便对接企业业务与各类云平台。



具体支持的运维能力包括：

- 敏捷交付

支持多种数据库集群的自动化部署，实现跨 IaaS 的统一服务编排，支持集群部署及其生命周期管理的统一、批量和自动化管理过程，支撑业务快速上线，提供企业级、标准化的数据库服务敏捷交付能力。



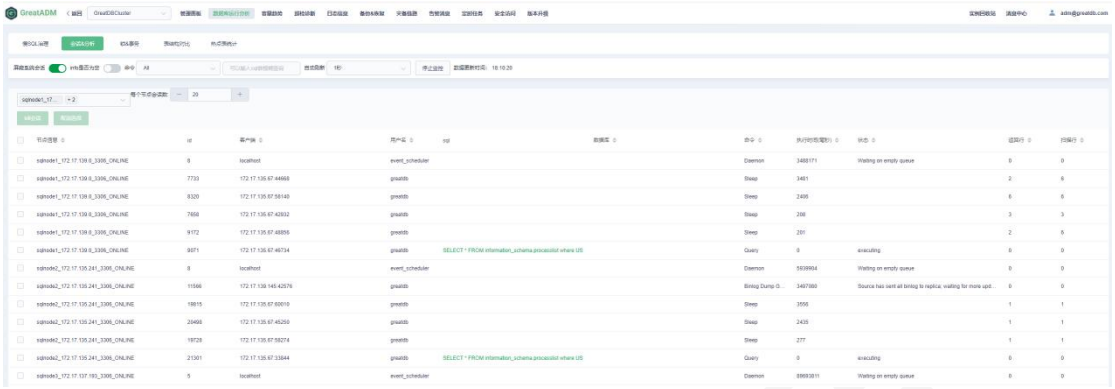
- 资源伸缩

基于云平台资源池和虚拟化管理能力，支持灵活的数据库服务伸缩调整，包括数据库集群纵向的 scale-up 扩展和横向的 scale-out 扩展，集群架构调整后，可自动实施数据重分布过程，满足业务增长对集群数据库的要求。

- 智能运维

运维管理平台对所管理的数据库提供自动化运维服务，包括数据库规范化部署与配置管理、生命周期管理、数据库备份恢复、监控告警等，定期的健康检查可协助定

位问题，给出改进建议，降低分布式集群运维复杂度分析数据库运行情况，发现异常 SQL 等能力，支持业务实现自动化、智能化运维。

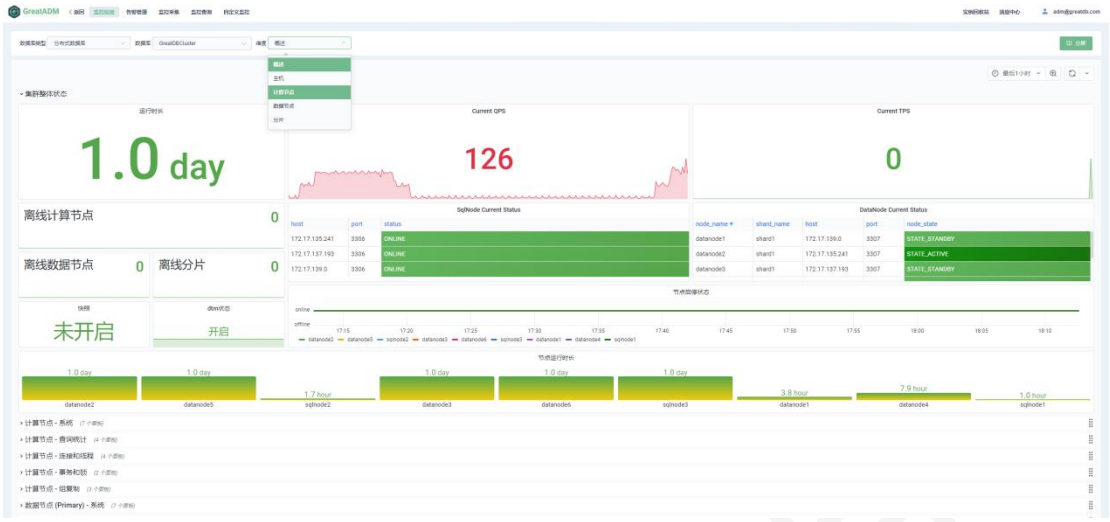


- 深度巡检

实现数据库集群深度巡检，升级全面的巡检算法和知识库，细化并支持可定制的巡检需求，提供全面的数据库健康报告，协助用户掌握数据库健康度，可视化展现数据库最近详细的运行情况。

- 监控告警

提供多层级多粒度的数据库集群监控告警功能，覆盖集群、节点、主机等层面，面向业务、运维等维度，实现可定制的监控与告警需求。



- 备份恢复

对管理的关系数据库和缓存数据库提供物理备份、逻辑备份、增量备份等多种备份方案，支持对分布式集群数据库的强一致性备份，并按照计划调度执行备份。基于全量和增量备份，可以按时间点恢复数据库。可以恢复集群到一台服务器上或虚拟化环境，用于资源受限场景的快速恢复。通过调度恢复任务，支持自动周期性的备份有效性验证。

- 全生命周期管理

作为数据库的一体化运维管理平台，支持资源及虚拟化管理、数据库集群启停、扩缩容运维管理、日常监控巡检、数据库升级、集群实例删除回收、SQL 开发等，实现分布式数据库集群全生命周期的管理。

4.4 产品性能

GreatDB Cluste 从整体架构上看，采用 Shared-Nothing 的计算存储分离分布式架构的设计，天生具有极强的可扩展性。目前在基于国际机构 TPC（Transaction Processing Performance Council，事务处理性能委员会）指定的 TPC-C 事务标准进行测试，测试参数选择 1000 仓库的情况下，6 个节点泰山服务器（64 核、512G 内存）的配置下性能 tpmC 达百万级，扩展比 0.7-1 之间。

4.4.1 并行计算能力

从具体实现细节上看，通过并发事务控制、事务锁优化、全新的数据流控算法、B+树并行扫描、计算节点条件下推、优化器采用 MPP 架构等方法提升并行计算能力，使得运行的更加平滑。

4.4.2 高并发的优化

计算节点和存储节点采用连接池，提升连接的复用，支持启用线程池技术，降低高连接时对内存资源的消耗，在高并发时数据库的性能衰竭小，保证了可支撑高并发连接访问时性能平稳，进而支持海量用户的大规模并发访问。

5. 安全特性

为了确保数据的安全性，数据库需要对数据的全生命周期进行数据安全防护。万里分布式数据库系统支持的数据安全功能，内容如下：

5.1 身份鉴别

GreatDB Cluster 中支持基于密码复杂度增强的身份鉴别机制，提供多种密码复杂度规则（如最小长度、字符类型等）来防止使用弱密码，从而降低账户被暴力破解的风险，提升数据库系统的整体安全性。

5.2 数据安全传输

传统 HTTP 不具备安全传输机制，它采用明文的形式传输数据，HTTP 本身没有提供数据完整性验证的功能，数据在传输过程中可能会被篡改，而接收方很难发现。安全性较弱，攻击者可以在客户端和服务器之间进行中间人攻击。GreatDB Cluster 支持通过 SSL 协议实现客户端和服务器之间的安全数据传输，使得数据在传输过程中难以被窃听、篡改、重放和伪造。

5.3 三权分立

GreatDB Cluster 根据《GB/T 20273-2006 信息安全技术 数据库管理系统安全技术》中关于“数据库管理系统安全技术分等级要求”支持三权分立特性，利用该特性可以在数据库安全性中通常指将数据库管理的职责分为三个不同的角色，以确保数据库的安全性、完整性和可用性。通过三权分立，可以建立一个多层次的安全防护体系，有效防止内外部威胁，确保数据库系统的安全性、完整性和合规性。

5.4 安全审计

任何系统的安全保护措施都不是完美无缺的，蓄意盗窃、破坏数据的人总是想方设法打破控制，审计功能将用户对数据库的所有操作和用户最后一次登录的信息自动

记录下来放入审计日志中，审计员可以通过对审计日志的分析，对潜在的威胁提前采取有效措施加以防范。

GreatDB Cluster 具备完整的审计机制，实现对数据库的全面监控并记录数据库操作，提升了数据库系统的安全性、合规性和管理效率，对数据安全和管理具有重要意义。

5.5 数据保护

GreatDB Cluster 数据库中的数据脱敏（data masking）特性通过对敏感数据进行掩码处理，防止未经授权的用户访问真实数据，从而保障数据隐私和安全。这在开发、测试和数据分析环境中尤为重要，有助于满足隐私法规要求并减少数据泄漏风险。

5.6 支持国密

GreatDB Cluster 支持在表空间加密以及链接传输中采用国密加密，确保在用户敏感数据存储和数据传输过程中免受未经授权的访问和篡改。这些加密标准符合国家网络安全法规，有助于保障国家信息安全和企业数据隐私。GreatDB Cluster 支持的国密算法有 SM2 非对称加密、SM3 信息摘要、SM4 对称加密。

5.7 备份恢复

GreatDB Cluster 拥有强大的备份和恢复功能，其备份类型分为两种：

- 全量备份：通过添加 backup_node 角色节点，在 SQLNODE 计算层直接执行 greatdb_start_full_backup 命令，启动全局一致性备份，备份集默认存在专有 backup_node 中。
- 增量/差异备份：在 SQLNODE 执行 greatdb_start_inc_backup 命令，启动增量或者差异备份，增量备份支持基于全量备份或基于历史增量上执行多次增量备份。

GreatDB Cluster 采用计算存储分离的架构，SQLNode 存储集群元数据，DataNode 存储用户数据。备份的时候，需要针对 SQLNode 和 DataNode 分别进行备份。备份和恢复具备以下特点：

- 恢复任意时刻（PITR）

GreatDB Cluster 采用计算存储分离的架构，SQLNode 存储集群元数据，DataNode 存储用户数据。备份时自动对 SQLNode 和 DataNode 分别进行备份。DTM server 记录着集群的全局一致信息 snapshot 启动实时备份该信息，用于保证任意时刻全局节点恢复数据一致性。恢复时可基于备份集+DTS server 全局一致性点位恢复出新的数据库集群后，提取误删和需要恢复的数据重新写回原集群。

- 任务管理可视化

GreatDB Cluster 提供备份过程的任务视图 greatdb_backup_tasks，支持实时查看备份过程 greatdb_backup_task_routine 状态和历史备份集信息等。同时 GreatADM 图形化运维管理平台为 DBA 提供便捷、可跟踪、可溯源的图形化界面，进一步提升 GreatDB Cluster 备份恢复的高效管理能力。

6. 部署环境和生态适配

6.1 部署环境

- 最低部署环境配置要求

服务器参数	参数大小
CPU	2 Cores
内存	4GB
磁盘空间	20GB
网络带宽	100Mbps/s
操作系统	Linux 系统（内核>=3.10.0，glibc>=2.17）

- 推荐分布式集群单节点部署环境配置模板一（以约承载 2000 TPS，10000 QPS，数据库容量 500GB 规模的 HTAP 业务场景为例）

服务器参数	参数大小
CPU	>= 16Cores
内存	>= 96GB
磁盘空间	>= 1TB
网络带宽	>= 1000Mbps/s
操作系统	Linux 系统（内核>=3.10.0，glibc>=2.17）

- 推荐分布式集群单节点部署环境配置模板二（以约承载 5000 +TPS，20000 QPS，数据库容量 1TB 规模的 HTAP 业务场景为例）

服务器参数	参数大小
CPU	>= 96Cores

内存	>= 256GB
磁盘空间	>= 4TB
网络带宽	>= 10000Mbps/s
操作系统	Linux 系统（内核>=3.10.0，glibc>=2.17）

6.2 生态适配

GreatDB Cluster 可以和产业链上下游产品无缝衔接，比如 ogg、informatica、kettle。所有的第三方工具、编程语言、客户程序可以像使用单机数据库一样与 GreatDB Cluster 集群进行完美的交互。

6.2.1 可适配的 CPU 架构列表

GreatDB Cluster 在硬件芯片和操作系统兼容性方面，不仅全面支持英特尔 x86_64 架构处理器，也对主流国产 CPU 平台实现了广泛适配，目前已兼容的国产 CPU 型号包括但不限：鲲鹏、飞腾、海光、兆芯、龙芯、申威等国产芯片。操作系统支持包括但不限于欧拉、银河麒麟、麒麟信安、凝思、统信、BCLinux、新支点、龙蜥等国产操作系统。

6.2.2 支持原生 MySQL 的访问接口

GreatDB Cluster 100%支持原生 MySQL 的访问接口。使用 MySQL 的连接器和 API 都使您能够从其他语言或环境（包括 ODBC、JDBC)、C++、Python、Node.js、PHP、Perl、Ruby 和 C）连接和执行 GreatDB Cluster 语句，同时支持的 MySQL API 和 Interfaces，除上述的生态兼容，GreatDB 也提供自有的配套驱动。

语言环境	API	类型	参考链接
Ada	GNU Ada MySQL Bindings	libmysqlclient	See MySQL Bindings for GNU Ada
C	C API	libmysqlclient	See MySQL 8.0 C API Developer Guide.
C++	Connector/C++	libmysqlclient	See MySQL Connector/C++ 8.0 Developer Guide.
	MySQL++	libmysqlclient	See MySQL++ website.
	MySQL wrapped	libmysqlclient	See MySQL wrapped.
Cocoa	MySQL-Cocoa	libmysqlclient	Compatible with the Objective-C Cocoa environment. See http://mysql-cocoa.sourceforge.net/
D	MySQL for D	libmysqlclient	See MySQL for D.

Eiffel	Eiffel MySQL	libmysqlclient	See Section 29.13, “MySQL Eiffel Wrapper”.
Erlang	erlang-mysql-driver	libmysqlclient	See erlang-mysql-driver.
Haskell	Haskell MySQL Bindings	Native Driver	See Brian O'Sullivan's pure Haskell MySQL bindings.
	hsql-mysql	libmysqlclient	See MySQL driver for Haskell.
Java/JDBC	Connector/J	Native Driver	See MySQL Connector/J 5.1 Developer Guide.
Kaya	MyDB	libmysqlclient	See MyDB.
Lua	LuaSQL	libmysqlclient	See LuaSQL.
.NET/Mono	Connector/NET	Native Driver	See MySQL Connector/NET Developer Guide.
Objective Caml	OBjective Caml MySQL Bindings	libmysqlclient	See MySQL Bindings for Objective Caml.
Octave	Database bindings for GNU Octave	libmysqlclient	See Database bindings for GNU Octave.
ODBC	Connector/ODBC	libmysqlclient	See MySQL Connector/ODBC Developer Guide.
Perl	DBI/DBD::mysql	libmysqlclient	See Section 29.9, “MySQL Perl API”.
	Net::MySQL	Native Driver	See Net::MySQL at CPAN
PHP	mysql, ext/mysql interface (deprecated)	libmysqlclient	See Original MySQL API.
	mysqli, ext/mysqli interface	libmysqlclient	See MySQL Improved Extension.
	PDO_MYSQL	libmysqlclient	See MySQL Functions (PDO_MYSQL).
	PDO mysqlnd	Native Driver	
Python	Connector/Python	Native Driver	See MySQL Connector/Python Developer Guide.
	Connector/Python C Extension	libmysqlclient	See MySQL Connector/Python Developer Guide.
	MySQLdb	libmysqlclient	See Section 29.10, “MySQL Python API”.
Ruby	mysql2	libmysqlclient	Uses libmysqlclient. See Section 29.11, “MySQL Ruby APIs”.
Scheme	Myscsh	libmysqlclient	See Myscsh.
SPL	sql_mysql	libmysqlclient	See sql_mysql for SPL.
Tcl	MySQLtcl	libmysqlclient	See Section 29.12, “MySQL Tcl API”.

6.3 良好的生态合作伙伴

GreatDB Cluster 具备完备的上下游生态适配体系

7. 典型案例

7.1 某银行缴费平台系统

- 业务概述

某银行缴费系统是国内一个便民缴费平台。便民缴费服务业务除传统的水、电、燃气、宽带通信、各类生活充值外，还有新兴的物业费、学费/住宿费、ETC 以及加油卡等，覆盖百姓生活方方面面。

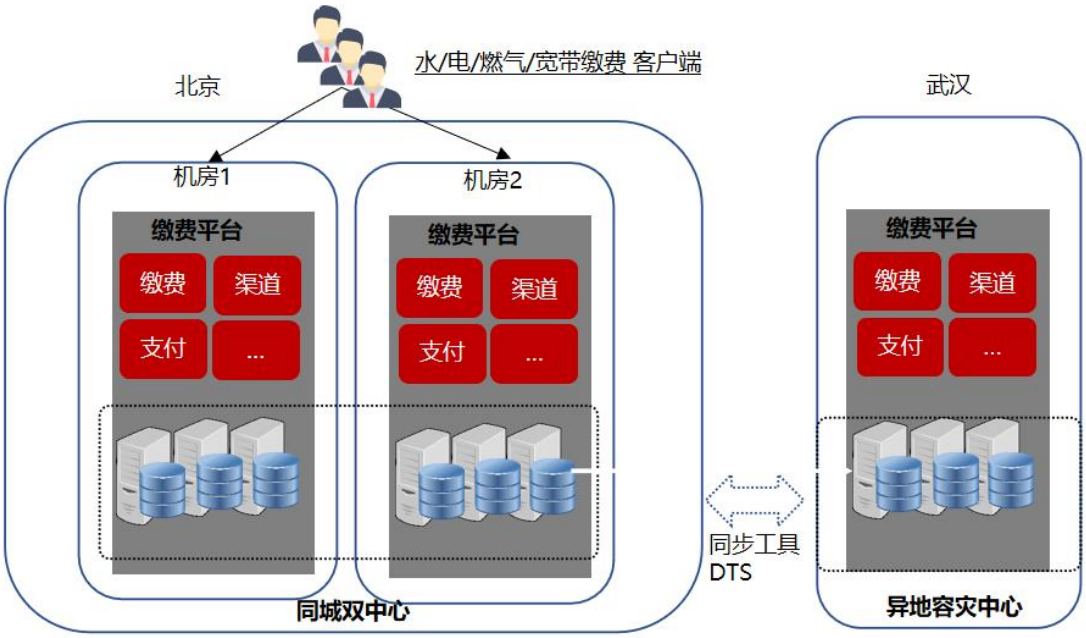
在基础公共便民缴费层面，该项目在电、水、燃气、有线电视、通讯、供暖费 6 大基础缴费服务方面，面向全国省、市、县 3 级区域进行覆盖，电力缴费服务已覆盖全国，水费地级市覆盖率达 72%，燃气费地级市覆盖率 57%，有线电视费覆盖 27 个省，供暖费实现北方供暖区全覆盖。截止 2020 年，该云缴费服务项目总数已突破 8400 项，覆盖全国 300+城市，累计用户共 5.49 亿户。

- 建设方案

集群采用两地三中心部署方式，北京同城双机房部署单套双活，同时从灾备角度部署武汉异地灾备集群，通过集群数据同步工具进行数据同步。数据库运维监控管理平台接管集群全组件的全生命周期运维管理。

该集群采用多副本全冗余设计与增强一致性协议，保证集群高可靠及数据强一致。同城双活部署，提供机房级高可靠，保证任意故障场景下 RPO 为零。集群故障自

动切换，实现秒级故障切换，切换对业务影响可控。



- 核心亮点

1. 支撑缴费系统峰值 5000TPS 交易量，QPS 超 70000，业务响应延迟<60ms，达到客户预期业务性能需求。
2. 打破核心业务 Oracle 长期依赖和高额扩展成本，实现业务增长的在线实时扩容，降低数据库扩展所需的维护窗口时间。
3. 通过配套同步工具实现异地同步，实现两地三中心部署，既满足客户业务容灾需求，又增强了年度容灾演练的可靠性。支撑银行缴费业务，实现事务强一致，确保 RPO 为 0。
4. 同时构建企业集中式数据库、分布式数据库统一运维管理的平台，实现多套数据库一套平台集中监控、巡检和运行维护。

7.2 某运营商经营分析系统

- 项目背景

经营分析系统为运营商的经营提供数据化运营服务，它以数据仓库作为支撑，将业务系统分散的数据整合集中起来，进行企业级数据建模、进行元数据管理、进行数据质量控制，提供“统一口径”的数据信息，同时向其他业务系统提供分析的支撑服务。

经营分析系统的应用场景主要包括统计报表、即时查询、OLAP 分析、分析应用专题和 BI 分析结果和应用固化到业务流程等。

- 需求分析

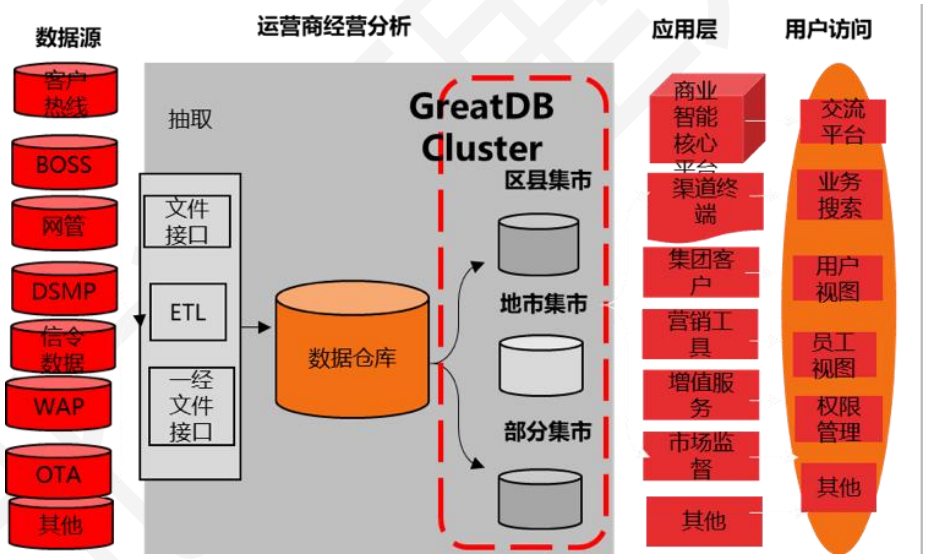
目前，经营分析系统的建设虽然产生了一定的效果，但从实际运行情况看，仍存在数据分散、重点不突出等问题。

统计报表作为目前经营分析系统的主要功能，包含了大量的经营数据，但各类经营数据存放在不同的后端数据库中，不同的数据库之间没有经过合理的组织，当开发者需要查询某个数据时，需要分别查找，数据分散，使用较为麻烦，最终反映到管理者层面，报表展现的效率低下，很难一目了然地看到公司运营中的关键问题，影响了经分系统的使用效果。

- 解决方案

万里开源软件有限公司针对经营分析系统目前存在的问题，提出一套完善的数据集市方案，包括业务库集群和门户配置库集群。业务库集群存储经分系统的维度数据和报表数据，为应用层提供指标计算结果和汇总信息；针对不同的业务部门和业务系统，搭建门户配置库，为用户访问统计报表和配置数据提供数据入口。

通过搭建业务库集群和配置库集群，实现了不同业务访问不同数据的目标，也借助并行计算和 HTAP 整体解决方案提升了报表查询性能，解决数据查找困难、报表展现效率低下等问题。



- 价值体现

存储海量结构化数据存储，其中北京移动数据量达到 130TB，河南移动数据量有 40TB，满足用户对历史数据的存储需求。

1. 数据库集群完成了一体化集中管控和运维，统一合并迁移超过 20 个独立特性的业务系统，完成统一标准化、规范化的集中管理。

2. 性能表现优于原始数据库，提升报表查询效率 40%，最快做到秒级汇总呈现，解决了原报表 SQL 聚合耗时较长的问题。

3. 解决了业务长期依赖国外数据库产品，实现数据库自主可控的改造目标。

8. 版权声明

8.1 法律声明

若接收北京万里开源软件有限公司（以下称为“万里数据库”）的此份文档，即表示您已同意以下条款。若不同意以下条款，请停止使用本文档。

本文档所载内容受著作权法的保护，著作权为北京万里开源软件有限公司所有，但注明引用其他方的内容除外。北京万里开源软件有限公司保留任何未在本文档中明示授予的权利。文档中涉及万里数据库的专有信息。未经万里数据库事先书面许可，任何单位和个人不得复制、传递、分发、使用和泄漏该文档以及该文档包含的任何图片、表格、数据及其他信息或者其他任何商业目的的使用。

8.2 商标声明

GreatDB 和 GreatDB Cluster 是万里数据库的注册商标。万里数据库产品的名称和标志是万里数据库的商标或注册商标。在本文档中提及的其他产品或公司名称可能是其各自所有者的商标或注册商标。在未经万里数据库或第三方权利人事先书面同意的情况下，阅读本文档并不表示以默示、不可反言或其他方式授予阅读者任何使用本文档中出现的任何标记的权利。

8.3 服务声明

本产品符合有关环境保护和人身安全方面的设计要求，产品的存放、使用和弃置应遵照产品手册、相关合同或相关国家法律法规的要求进行。

本文档按“现状”和“仅此状态”提供，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。本文档中的信息随着万里数据库产品和技术的进步将不断更新，万里数据库不再通知此类信息的更新。

GreatDB
万里数据库

联系我们 Contact Us

地址：北京市朝阳区CBD国际大厦7层701B
电话：400-032-7868
邮箱：sales@greatdb.com
网站：https://www.greatdb.com

北京万里开源软件有限公司

稳定 · 性能 · 易用