# 概述

## 简介

## 产品性能

GoldenDB通过分布式执行缓存、并发事务控制、数据复制优化等手段，对整体性能进行提升。在金融行业客户实测数据中，银行转账核心业务系统TPS能够达到4万（2017年），银行信用卡核心交易TPS10万（2020年），满足银行业务高并发低延时的特性。

# 架构

## 逻辑架构

### 数据库驱动

数据库驱动以集成的方式嵌入到具体应用中，和应用一起部署。GoldenDB支持通用数据库标准协议，并能够提供JDBC、ODBC、Java、Python、C、C++等常见开发语言使用的接口驱动。

从功能角度来看，数据库驱动实现了计算节点的透明接入、负载均衡和故障透明转移（LVS）。

分布式数据库的计算节点负责处理应用的SQL请求，计算节点集群由多个计算节点组成，用户可以规划计算节点和应用之间的对应关系（可以理解为一种负载均衡）。数据库驱动根据配置的规则将应用请求发送给合适的计算节点，并确保负载均衡地分配到这些计算节点上。当这些计算节点发生故障后，驱动层能够实施透明的故障转移，将应用的新请求发送给正常的计算节点，并在故障节点恢复后，能够将应用的请求重新路由到该节点上。

### 计算节点集群(Proxy)

计算节点集群是分布式数据库的核心层，由无状态的计算节点（Proxy）组成。计算节点从驱动层（业务操作）或者管理节点（内部管理命令）接收用户的操作（一般以结构化查询语言描述，即SQL），进行逻辑优化和物理优化，生成满足分布式事务一致性的分布式查询计划。计算节点在执行分布式查询计划时，通过持续地访问数据节点，完成用户的最终操作请求。用户可以根据应用对可靠性、可用性、性能等因素的不同要求，对计算节点进行合理的规划。

### SQL引擎

### 数据节点集群

数据节点集群是应用数据的最终存储组件。所有的数据节点组成一个或多个数据库集群，用户操作的事务尽量不要跨越多个数据库集群（目前已经支持跨集群，但是还是不够完善）。

数据库集群由一个或多个安全组（DBGroup）组成，集群中每个表中的数据按照某种策略进行横向分片后存放到对应的安全组中，分片策略包括复制策略、哈希策略、范围策略和列表策略。

数据按照上述策略（复制策略除外）分片后，每个安全组上的实际数据在理论上只有总数据量的1/N（数据分布的均匀程度依赖于切分策略和真实数据分布的匹配程度）。随着安全组数量的增加，每个安全组承载的数据量和读写负载会相应的减少，从而在数据节点集群内部具备了读写能力的水平拓展。

安全组是由一个或者多个数据节点组成的数据库节点组，组内的数据库节点拥有相同的数据。当安全组存在多个数据节点时，其中一个数据节点为主用节点其他数据节点都是备用节点，数据在主备节点之间实时复制。主用节点具备读写能力，备用节点可以提供度能力。

### 管理节点

管理节点在分布式数据库中负责集群管理流程，不涉及业务的访问流程，无负载压力，一般采用两节点主备方式部署。管理节点按照功能分工，可以有如下几个模块：

#### 统一运维管理(OMM)

OMM是GoldenDB分布式数据库的统一操作维护入口，可以进行用户和权限管理、元数据管理、计算节点管理、数据节点管理、DDL执行、节点扩容、备份恢复、统计及告警管理等。

#### 元数据管理器(MetaDtaServer)

元数据指的是数据的元信息，如库、表、视图、存储过程、触发器、自定义函数等数据模型的定义，元数据管理器存放系统的全量元数据，是整个分布式数据库集群的元数据中心。

此外，GoldenDB元数据管理器还保存了整个集群的拓扑信息，因此是更广义的元数据管理。

为了提高启动和运行的效率，除了元数据管理器存有元数据定义外，计算节点和数据节点也会存放元数据定义，但是计算节点和数据节点只存放本节点所涉及应用的元数据定义，即当计算节点中的元数据和管理节点不一致时，会同步管理节点的元数据到本地。

#### 计算节点管理(ProxyManager)

负责管理计算节点集群。管理工作一般分为两类：

一类为集群的组建管理，包括计算节点的创建、启用、禁用和删除；

另一类为集群的应用管理，包括定义计算节点和应用的对应关系、计算节点异常后的数据恢复调度。

#### 数据节点管理(ClusterManager)

数据节点管理也分为两类：

一类为集群的组建管理，包括数据节点、安全组、数据节点集群的创建、变更和删除；

另一类为集群的任务管理，包括数据节点异常、恢复后的调度管理、数据节点备份恢复的调度、数据重分布等功能的任务调度管理。

#### 全局事务管理节点(GlobalTransactionManager)

全局事务管理器在分布式数据库中维护全局事务的全生命周期，提供申请、释放、查询全局事务的能力，并采用双活方式部署。

## 物理架构

数据库驱动：可以直接使用标准数据库驱动接入GoldenDB集群，如JDBC，可以支持按照最快响应时间和均匀随机方式将SQL请求下发至多个计算节点，当前也可以使用常用的负载均衡设备LVS、F5等接入GoldenDB集群。

**计算节点Proxy：**计算节点理论上可以单机部署，但是通常要求部署2个以上，以便当某些计算节点发生故障后，驱动层能够实施透明的故障转移，将应用的新请求发送给正确的计算节点。

**数据库集群：**数据节点集群可以部署一个或多个，当业务之间的数据要求物理隔离时，可以将不同的业务数据存放在不同的数据节点集群；也可以多个业务的数据共享一个数据节点集群。每个集群内部的安全组数目可以根据业务量、访问性能要求、硬件条件确定；理论上每个安全组内部的副本数目越多，可靠性就越高，但是成本也越高。

**管理节点：**管理节点的各个模块可以单机部署也可以使用HA（现在已经使用ZK）软件进行双击冷备部署。

**全局事务控制节点：**分布式事务涉及节点间数据提交的一致性，该数据的一致性主要依赖于全局事务的状态，所以全局事务节点必须是高可靠的，全局事务控制节点为双活模式，在发生异常时可以进行服务快速接管。

计算节点集群具备横向扩展能力，数据节点集群具备横向和纵向扩展能力。在高性能要求下，分布式数据库集群会涉及大量的计算节点和数据节点。物理部署计算节点和数据节点时可以非常灵活，既可以部署在物理机器上，也可以部署在虚拟化平台上；同时每个节点既可以部署在一个设备上，也可以将多种类型的节点部署在同一设备上。不管采用哪种方式，一般都需要遵守如下原则：

1. 计算节点需要分配更高的计算资源，数据节点需要分配更高的内存和存储资源；
2. 归属于同一个应用的多个计算节点避免部署在同一台物理机器上，更进一步避免部署在同一机框或者数据中心；
3. 归属于同一个安全组的多个数据节点避免部署在同一台物理机器上，更进一步避免部署在同一机框或数据中心。

## 系统数据流向

# 原理

## 协作流程

## 数据库客户端

## Proxy

Proxy是无状态的代理节点

### 逻辑架构

### 前后端通信协议

### 元数据缓存及持久化

### SQL解析

### SQL处理及优化

### SQL执行

### SQL路由

## GTM

全局事务协调中心，用于协助Proxy进行分布式事务管理，主要包括全局事务ID的生成和活跃事务的维护以及当前活跃GTIDs的快照。

### 逻辑架构

消息分发模块：负责将收到的外部请求消息根据clusterid转发至相应的消息处理模块。

消息执行模块：负责处理收到的外部请求消息，并进行GTID的维护，同时将内存消息同步至备机模块。

数据持久化模块：后台定时将激活的GTIDs持久化，同时需要考虑其他手段降低GTIDs丢失的概率，但仍可能丢失一部分最新的GTIDs，如果GTM异常，则需将大于已持久化的GTIDs的所有分布式事务回退，这样可能会误差一些已经结束的分布式事务，但是却是保证数据强一致性的很安全的方案。

### 设计要求

#### 性能要求

为提高GTM全局争抢GTID的性能，GTM的线程结构设计为一个分发线程加若干可配置的执行线程。

1. 消息分发模块根据请求消息中携带的clusterid将其分发至固定的执行线程
2. 每个执行线程对应的clusterid维护一套独立的GTIDs
3. GTM应能根据clusterid和proxyid索引到GTIDs

#### 可靠性设计

GTM采用双活的模式，双机之间有心跳机制，双机内存同步更新。

脑裂问题的解决：引入第三方监控，在主备机之间发生网络故障时，由第三方决定。认为不存在双机之间及双机和第三方之间均出现网络故障的情况。

## MetaDataServer

## ProxyManager

## ClusterManager

## DBAgent

部署在DB节点上，负责DB的配置更改、复制变更、状态监控、统计上报等，同时配合实施事务回退、集群管理、数据备份、数据迁移等功能，DBAgent使得这些功能落地。

## DataManager

数据控制模块，主要负责数据重分布的管理、数据备份与恢复的管理。

## ClusterReplication

集群复制模块，用于集群间数据复制，保证数据一致性。

## 模块间统一通讯

统一消息通信是ZXOS统一支撑的核心组件，完成消息线程的建立、调度、管理、回收等，负责底层的通信链路管理包括端口的侦听、建链、数据的发送和接收、多条并发链路的管理等。

## OMMServer

OMMServer通过web节点提供人机交互功能，具体包括集群规划、集群变更、元数据操作、权限管理、告警与监控、日志分析等功能。

OMMServer与MetaDataServer共用一个数据库，但实行分库管理，各自维护自己的数据；OMMServer会根据需要读取MetaDataServer的库做展示，但是不会对其进行写操作，要做好权限控制。

# 操作

# 分布式事务

分布式事务和单机事务一样，都需要保证事务的ACID四个属性。在分布式环境下，由于网络通信延迟、故障和服务器故障等因素，保证事务一致性尤其具有挑战性。目前业界分布式数据库的数据一致性解决方案为强一致性、弱一致性和最终一致性。GoldenDB实现了强一致性的解决方案，即当用户完成数据更新之后，后续所有读操作都能且只能读到更新以后的值，这是事务一致性的最高级别。

注：强一致性虽然安全性更高，但是也存在性能的问题，即在出现故障时回滚代价非常高。

## 方案

在GoldenDB分布式事务解决方案中，数据节点作为分布式事务参与者要保证对自身数据库操作的本地事务满足ACID属性，计算节点作为分布式事务协调者协调多个本地事务完成整个分布式事务控制，同时将全局事务状态实时记录在全局事务管理器中。

采用全局事务ID对分布式事务进行控制，其核心思想是全局事务控制和标签数据。前者为每一个分布式写事务分配一个全局唯一的有序事务ID（GTID），并根据事物的存活情况维护对应ID的生命周期。后者在用户表中增加对应用透明的GTID列，并在该列中维护操作本行数据的最近一次分布式事务对应的全局事务ID。

总体在流程交互上为一阶段提交+补偿事务的方式，如果事务在提交阶段有部分节点失败，本方案将采用回滚已成功提交的事务。

## 特点

GoldenDB的分布式事务实现方案最大的特点即为其在获得事务强一致性的同时还能具备较高的性能，可以与常用的两（三）阶段提交做对比：

两（三）阶段提交的核心思想是通过前期的多次准备和协调，尽可能的让最后的提交操作能够成功。而在实际场景中SQL执行成功，最后commit阶段失败的概率极低。本方案认为大部分事务可以一次提交成功，因此采用一阶段提交+补偿事务的方式，与两（三）阶段提交相比，本方案在大部分情况下减少了与数据节点的交互次数，降低了锁冲突的概率，提升了事务处理效率。

虽然如此，分布式事务相比非分布式事务还是会占用更多的资源，GoldenDB提供了灵活的分布式事务控制，应用可以根据实际情况控制是否启动分布式事务。控制参数具备多粒度能力，可以在系统级、会话级、事务级三个维度控制，同时也提供了多种读隔离级别：

UR（uncommited read）：未提交读，即计算节点不做任何分布式事务控制，业务要么允许脏读，要么不存在读的时候同时写；

CR（consistency read）：强一致性读，在高并发读写时，严格杜绝脏读的可能性。

## 待优化

# 数据分布

集群中每个表中的数据按照某种策略进行横向分片后存放到对应的安全组中，分片策略包括复制策略、哈希策略、范围策略和列表策略。

## 复制策略

复制策略（duplcate）：适用于不经常修改（因为修改需要同步多个节点，比较耗时），且频繁出现在关联或子查询中的小表。复制策略下，复制表的数据保存在每一个节点都需要这个表中数据的情况下，可以减少节点之间网络数据（RPC）的传输，提高查询性能。

## 哈希策略

哈希策略（Hash）：适用于将数据均匀的分布到预定义的各个安全组中，保证各个安全组的数据量大致一致。一般用户不需要关心分发字段的取值范围和具体含义，且对该表的SQL操作基本都是等值操作的场景。

## 范围策略

范围策略（range）：适用于指定一个给定的列值或列值集合应该保存在哪个安全组上，常用于时间、日期、数值等类型的字段上，如数据按照自然月或者自然天分布存储。

## 列表策略

列表策略（list）：适用于含有一定列限定性或枚举性的字段上，如数据按照机构代码、国家代码、地区代码分布存储。

数据按照上述策略（复制策略除外）分片后，每个安全组上的实际数据在理论上只有总数据量的1/N（数据分布的均匀程度依赖于切分策略和真实数据分布的匹配程度）。随着安全组数量的增加，每个安全组承载的数据量和读写负载会相应的减少，从而在数据节点集群内部具备了读写能力的水平拓展。

安全组是由一个或者多个数据节点组成的数据库节点组，组内的数据库节点拥有相同的数据。当安全组存在多个数据节点时，其中一个数据节点为主用节点其他数据节点都是备用节点，数据在主备节点之间实时复制。主用节点具备读写能力，备用节点可以提供度能力。

# 主从复制/一致性

## 主从复制

## 一致性检查

# 备份恢复

## 备份

GoldenDB的备份属于物理备份，备份的数据包括每个分片的数据文件、每个分片的binlog日志及每个时刻的全局事务列表快照，后两种数据在进行集群的一致性数据恢复以及数据恢复到任意时刻时使用。

对GoldenDB的数据备份功能概括如下：

1. 备份任务管理。

GoldenDB支持在OMM上进行备份任务的管理，包括备份任务的发起、备份历史操作的查询等。

备份任务的发起一般有两种模式：

一种为实时备份，即用户配置好后立即会发起一次备份任务；另一种为定时备份，用户可以设置定时备份策略，典型的策略为每周备份全量数据，其他时间分别在周日全量备份的基础上做一次增量备份，如此，可以通过全量数据和其他任一增量备份数据，快速恢复出想要的那天的数据。

1. 备份文件存储。

GoldenDB的备份文件可以在分片节点本地挂载NFS共享目录，将备份结果文件统一存放；同时GoldenDB也支持与IBM TSM系统对接，可以通过TSM直接将备份结果灌入到磁带库。

## 恢复

GoldenDB的数据恢复功能概括如下：

1. 可恢复到任意时刻。

由于GoldenDB有数据节点的全量及增量备份文件，同时有其运行过程中的binlog日志，借助这些数据可以将系统恢复到任意需要的时刻，但需要注意全量及增量备份文件恢复的速度要大大快于binlog日志的回放速度，因此无特殊要求，建议选择恢复到某次备份的结束时刻，以便更快的完成数据恢复。

1. 一致性的数据恢复。

由于GOldenDB备份了运行过程中的binlog日志及每个时刻的全局事务列表快照，因此可以根据这些信息恢复到一个全局一致性的数据副本。

1. 恢复任务管理。

GoldenDB支持在OMM运维平台上进行节点的一键恢复操作，用户直接在OMM上选定要恢复的节点以及要恢复到的时间点，即可以进行自动化的数据恢复。

# 兼容性

# 拓展性

## 多级扩展

GoldenDB的架构设计使得其具有良好的扩展性，可以从如下几个角度分析：

1. Proxy计算节点是无状态的，可以动态任意扩展；
2. 数据库集群可以根据需要进行如下维度的扩展：
3. 在安全组内部添加数据节点增加数据副本数。一方面可以获得更高的可靠性，同时也可以通过读写分离功能达到读能力的线性扩展；
4. 增加安全组数目。获得数据节点存储和计算能力的线性扩展，在增加数据的分片时需要考虑如何将数据重新迁移到新分片上，即表数据的重分布；
5. 增加数据节点集群数目，扩展组建新的集群。该方法可以在得到处理能力扩展的同时做到不同业务间数据的隔离性，用户可以根据要求选择。

GOldenDB计算节点和数据分片的数目配比推荐为1:1，性能可以随计算节点和分片数目的增加线性扩展，损耗控制在10%内。若业务为CPU密集型，则计算节点书目要适当增加，若为IO密集型，计算节点数目可以适当减少。

## 数据重分布

业务的增长不可避免的需要对资源进行扩容，由于使用了分片技术，数据被切分橙细小的分片分布在数据节点集群中。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）。

GoldenDB数据重分布功能的基本原理是对重分布表，建一个和原表字段结构相同但是分发策略为目标分发策略的新表，对于需要重分布的数据分片，将其上的数据导出，然后将该部分数据导入到新表，导入过程会使用新表的分发策略，最后将新表改名为原表。其中数据的导出导入过程又是个多次反复渐进的过程，每次导出的数据均是在前一次导出导入过程中产生的增量数据，直至最后仅剩少量差距时对应用进行短暂禁写，最终完成切换。

GoldenDB的数据重分布对在线业务影响小、且课操作性强，具备以下特点：

1. 执行时间可控制，对业务影响秒级。重分布过程中数据的迁出迁入是循序渐进的过程，仅在原分片和目标分片数据存在少量差距时才会禁写，因此对在线业务影响秒级，并不影响读服务；
2. 支持多表并发重分布。在分布式数据库的数据模型中，会将有关联关系的表采用相同的分片策略。在数据重分布过程中，这些相关表可以选择同步进行处理，以避免在数据重分布过程中由于相关表的分片策略临时不同而导致该SQL语句无法直接下压，影响SQL执行性能；
3. 重分布数据智能识别。数据重分布过程中会涉及到数据从源分片至目的分片的移动，开销很可观，GoldenDB支持对表的源分片信息和目标分片信息进行详细的分析比对，最大程度的减少数据在分片间的移动，提升重分布效率（采用更加细粒度的bucketid，作为隐藏列存入表中）；
4. 可操作性好。GoldenDB在统一运维界面OMM上提供重分布操作界面，可全程可视化操作，包括执行、暂停、继续、取消、异常情况下的重试等；
5. 产品通用性好。对比需要预先做好分片规则的预sharding方案，GOldenDB的重分布方案对设计人员的要求相对较低，GoldenDB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。

## 异构数据库

采用slot、o2g（oracle to goledendb）、g2o（goldendb to oracle）实现异构数据库之间的在线业务和离线业务对接。

# 高并发

## LVS

## Proxy

### 池化技术

#### 线程池

#### 连接池

### 缓存

GoldenDB通过构建执行计划缓存、SQL缓存，提高SQL解析效率、提高数据读取效率，在高频度读写系统中可以减少磁盘I/O负担，提升整体系统效率。

#### 元数据缓存

#### 执行计划缓存

#### 结果集缓存

### 并发

GoldenDB在事务处理上通过采用成熟的主流技术来实现高效的事务管理，这些技术包括：以行级为主表级锁为辅的锁技术、多版本并发控制技术、全乐观锁+自动补偿机制。这些技术在保证事务ACID特征的前提下大大提高了事务的并发处理能力。

### 限流

根据消息积压数进行计算，设置流量限定阈值。

### 分包

对于大结果集进行分包处理（结果集在内存中缓存），防止一次性处理卡顿。

分布式批处理。提供分布式架构下批处理功能，满足金融、政企、运营商等行业日终大数据批处理的要求，通过分布式FetchSize和存储过程功能对数据进行批处理，减少客户端与DB的交互次数，批量返回数据集并进行批量处理。

### 读写分离

读写分离是指利用数据节点集群安全组多副本，将部分读请求发往备节点，提升系统的读能力。

在启动读写分离时，GoldenDB的计算节点在受到应用SQL请求时，根据当前的语句类型和负载策略选择SQL下发的数据节点，将写操作发往主节点，将读操作发往备节点。

注意：GoldenDB的读写分离是事务间的读写分离，如果一个事务内同时存在读写请求，该事务的所有SQL都会发往主节点。

在多个应用接入一个数据节点集群时，为了满足不同应用的需求，GoldenDB支持对同一集群不同的连接实例，设置不同的读写分离模式，包括以下三种：

1. 读主节点。读操作默认发往主节点，当应用强制指定将读请求发往备节点时，读请求即在备机间做负载均衡；
2. 读备节点。读操作仅在备机间根据配置的权重做读负载均衡。

除了上述连接实例级别的读写分离模式设置，GoldenDB还支持SQL级别的读写分离模式指定，应用可以在SQL语句后面添加hint信息强制发往主节点或备节点，SQL级别的优先级高于连接实例级别。常见的使用场景如下：

1. 由于数据在安全组内部的节点间同步存在时延，因此对实时性要求比较高的SQL请求，应用希望将其发往主节点；
2. 对于一些SQL如分析聚合类SQL，应用希望将其发往备节点，减少对主节点的影响。

结合上述连接实例和SQL级别的读写分离模式设置，应用可以根据自己的希望设计合理的读写分离策略。

### 负载均衡

可以设置不同group的优先级，同城机房高于异地灾备机房。

## SQL引擎

### 全局(唯一)索引

### MPP

采用presto大数据组件，对于复杂的SQL进行计算。

### force index

为了避免update、delete中where条件索引失效，造成全表锁（对于悲观锁，proxy会先下发select for update where锁住对应数据，然后执行更新），采用在where条件中增加force\_index。

### 指定group

不需要计算分片，直接到对应的group。

### samedb

如果可以确定某一个操作全部是对一个group的操作，前面第一个计算分片信息后，后面的全部添加samedb的hint信息，这样就可以直接用前面缓存的group信息了。

## DB

### 快同步

分布式数据库通过增加副本数来提高系统的可用性，为了数据的安全可靠，数据必须在满足拥有一定数量的副本之后，才返回处理结果给客户端。GoldenDB采用自研的gSync复制技术，在配置的副本同步策略满足后，主库返回操作结果给客户端；通过线程池、非阻塞式同步、并行复制等关键技术可以实现在确保数据RPO为0的同时保证系统的吞吐量。

### 大容量数据处理

GoldenDB具备完备的海量数据管理功能，提供PB级的数据容量支持，具有高效索引和查询优化技术，具备海量数据处理能力并支持大数据量用户的并发访问。

## GTM

### 多线程

Proxy分别采用三个线程线程分别执行申请、提交、回滚GTID的操作。

GTM线程结构设计为一个分发线程加若干可配置的执行线程。

## 待优化

# 高可用

## 方案

## 组件高可用

## 部署

### 机房内高可用

即组件高可用

### 同城双活

### 两地三中心

# 数据压缩

GoldenDB支持表数据压缩和备份压缩。表数据压缩时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之间。

备份压缩是指备份出来的数据会进行要说来减少空间占用，GoldenDB采用quicklz算法对备份文件进行压缩，压缩比在2~10之间。

GoldenDB同时支持对表的数据空间进行归并整理，可以实现对通过数据压缩和归并整理优化出来的表空间进行释放到表空间，从而实现最大限度的存储优化能力。

# 数据安全

## 备份恢复

## 加密审计

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保证。分布式数据库系统支持数据安全功能包括：

1. 访问鉴权

分布式数据库需要对来访用户进行鉴权，通过IP白名单的方式过滤连接请求，支持通过国密算法对用户名密码进行解密校验。

1. 通道加密

分布式数据库对客户端和数据库的连接进行SSL通道加密，防止通过抓包等手段查看业务数据。

1. 数据加密

分布式数据库支持表级别和列级别的加密。通过列级别的加密，表中仅存储加密后的字符串，在不知道加密串的情况下，即使查询出来也会是乱吗；表级别的加密可以加密整张表，在没有秘钥文件的情况下，即使拷贝走表数据文件，也无法破解表中数据。

1. SQL日志审计

分布式数据库支持所有的业务SQL写入审计日志文件，以便对SQL语句进行统计和审计，找出潜在的威胁。

# 数据迁移

## 重分布

对于少量数据的迁移可以采用重分布。

## 导入导出

数据导入导出一般用于系统间的数据迁移，小到导入导出一部分数据，大到数据割接、数据库迁移式升级、数据的分库、海量数据的迁移等，使用场景十分丰富。

GoldenDB支持将集群中符合查询条件的记录导出到指定的数据文件中，也支持将外部数据文件导入到集群合适的数据分片中。

导入导出功能：

1. 外部接口。GoldenDB导入导出的外部接口为文件，待导入的格式要符合要求，一般为字段间采用分隔符隔离的文本文件，导出结果也同样为文本文件。
2. 工具插件。导入导出可以认为是系统的一个功能插件，部署灵活，与在线交易流分离，可以降低对联机交易的影响。
3. 原始文件自动导入数据库集群。GoldenDB的导入导出工具，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，将数据下发至对应数据节点，最终完成导入。这其中的各环节均支持并发处理，提升整体导入效率。工具也支持带条件导入，即仅将满足条件的记录导入至系统中。
4. 多分片数据统一出口导出。GoldenDB的导入导出工具，能够自动控制各个分片并发进行数据导出，并可以选择将导出文件上传至同一个服务器，进一步的，也可以将多个导出文件汇总成单个大文件，用户可以灵活选择控制导出过程。
5. 容错处理。导入操作耗时长，对系统资源占用多，且通常是将外部系统的数据文件导入到分布式系统中，而外部的文件往往会有异常数据。因此，导入操作经常会遇到错误，GoldenDB充分考虑导入异常的错误处理，举例如下：
6. 导入过程中遇到异常数据时能够将异常数据分拣出来，待后续处理；
7. 流程处理的各个阶段都有重试机制，最大限度保证导入成功；
8. 系统异常时能够从异常位置继续执行导入流程；
9. 支持对原始文件记录末尾的空值进行处理。

**说明：**

使用loadserver或datatransfer进行数据导入导出，走的是后端链路（DB），导入导出成功后，需要在前端连接proxy设置一下sequence自增列的初始值（因为后端链路导入后sequence的值proxy并不知道）。

使用goldendumper走proxy执行导入导出，这个性能可能会消耗比较大（因为是采用insert+delete的方案），可以采用一个专门数转的proxy做这个操作，配置也做相应的优化。

## 日切卸数

在金融领域，针对应用使用传统数据库存在批前卸数丢失跨天事务，分析报告不准确的问题，需要事后进行手工修复数据。GoldenDB提供日切、卸数插件，可以获得准备的当日业务数据，更快得到准确的分析报告，无需在事后手动修复数据。

以导出日切时刻的数据快照为例，使用GoldenDB日切卸数功能导出数据无需经过ODS处理，既能满足事务全局一致性，又能满足具有业务含义的数据一致性（在日切前发生的事务都保留在快照数据中，日切后发生的事务操作都不体现在快照数据上）。

# 分布式查询优化器

分布式查询优化器是GoldenDB计算节点的核心功能，计算节点在接收到SQL语句后，首先由解析模块转成等价的可识别的语法树，然后由优化模块转成等价的执行计划树，最后由执行模块按照步骤执行该计划树，从而完成整个SQL语句的执行。

优化器的优化工作主要体现在计划树的生成上，GoldenDB的查询优化器设计实现主要考虑以下两个方面：

1. 代价模型的选择。

GoldenDB采用分布式系统代价估算模型，考虑节点间传输数据的代价，以减少数据传输的次数和数据量作为查询优化的目标，提高数据节点之间计算的并行度、减少计算节点的计算量。这主要考虑在分布式数据库系统环境中，表结构被水平或垂直拆分到多个数据节点，因此需要考虑语句如何分拆、分片之间数据如何移动、结果如何计算与合并的问题，网络通信开销不可忽视。

1. 考虑数据一致性开销。

在分布式数据库系统中，数据全局一致性机制相较于单机数据库需要更为复杂的控制。因此，如何降低数据全局一致性保证的开销，也是GoldenDB查询优化器的设计要求。

总体来讲，GOldenDB的分布式查询优化器遵循了上述的设计原则，以基于规则的优化为主，基于成本的优化为辅，在提升系统的灵活性的同时控制系统实现的复杂性。优化器内部内置大量的优化规则，通过查询重写的方式进行经验性优化。在优化规则的选择上，重点分析分片剪枝、并行执行、合并下压、条件下推、条件繁殖、排序消除、去重消除、排序下推等。

得益于完善的优化器设计，使得GoldenDB对单节点、跨节点的复杂SQL的兼容支持程度很高，包括跨节点SUM、COUNT、AVG等汇聚类操作，跨节点WHERE、FROM等子查询，跨节点JOIN，跨节点GROUP BY、ORDER BY、LIMIT等。这是GoldenDB将Proxy命名为计算节点而非中间件的原因之一，也是其和很多分布式数据库产品中间件的主要区别。

# 运维

## OMM

GoldenDB支持的运维能力包括：

1. 一键式安装、升级。
2. 配置可视化。
3. 统计监控。
4. 故障自动检测和告警。
5. 日志。
6. Explain查看执行计划。
7. Dbtoll管理工具。
8. OMM统一运维平台。提供网络拓扑、集群管理、元数据管理等功能。

## Insight

OMM的升级版本，提供了更加详细的监控信息。

## 巡检脚本