腾讯云数据库 TDSQL

Tencent Distributed MySQL

[2019 年 1 月]

[版本 2.1.0]

腾讯云

【版权声明】

【商标声明】

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或模式的承诺或保证。

目录

1 发展历史

腾讯云数据库（TDSQL-Tencent Distributed MySQL）是随着腾讯业务规模不断扩大而发展起来的，其定位是基于互联网分布式架构的金融级数据库。从 2004 年开始，腾讯充值等计费类业务快速发展，传统方案已经无法支撑日益增长的需求，通过多年摸索，最终选择兼容 MySQL 协议的分布式架构。并根据业务需求推动数据库架构不断革新。截止到 2019 年，腾讯云数据库 TDSQL 经过 15 年的发展，已经覆盖公有云、专有云部署模式，为银行、保险、证券，财付通、腾讯充值，阅文集团，区块链等 5000 多个金融类业务提供底层支撑。

腾讯公司90%金融计费的业务都采用TDSQL！

2 业务挑战

2.1 数据一致性

对某些业务（如金融业务）来讲，数据的强一致（Consistency）尤为重要。如果出现数据丢失，就意味会给组织或用户带来直接的金钱方面的损失，甚至影响企业的商誉和信誉。因此，数据的一致性是数据库管理员(DBA)最需要考虑的问题之一。

然而，多数开源不适用于共享存储架构，基于主从高可用架构难以做到既满足性能又保障主库出问题时数据不丢失，无法满足业务高并发需求。

2.2 服务可用性

随着业务需求的不断提高，搭建一个数据库高可用环境已经成为很多企业迫切的需求。确保企业中计算资源的持续可用是各个数据库管理员(DBA)的主要目标。如果支持应用程序

的数据库和服务器不可用，会导致大量投诉或用户流失，甚至带来金钱方面的损失，影响信誉和商誉。高可用性和减少停机时间是数据库系统的目标，某些业务甚至需要 24*7 无障碍运行。

2.3 扩展性

业务在采购之初很难准确预测未来业务增长的速度和总量，这就导致业务不得不采购比自己实际需求更多的资源。这可能导致：**资源的浪费，**您可能利用了 10%的资源，而浪费了 90%；或者**难以扩展：**您的业务发展可能远超预期，您又不得马不停蹄地采购更高配置的资源，不断的停机迁移。当然，scale-out （横向扩展）的分布式架构可以解决了这个矛盾，但目前这一起步门槛会较高。

2.4 信息安全

在这个大数据时代，数据和数据库安全比以往任何时候都更加珍贵。一旦数据发生泄露，那么付出的代价将是非常惨痛的。由于数据泄露而导致的业务中断、客户信心丧失、法律成本、监管罚款，这些后果可能导致数百万的花费甚至是灾难性的。

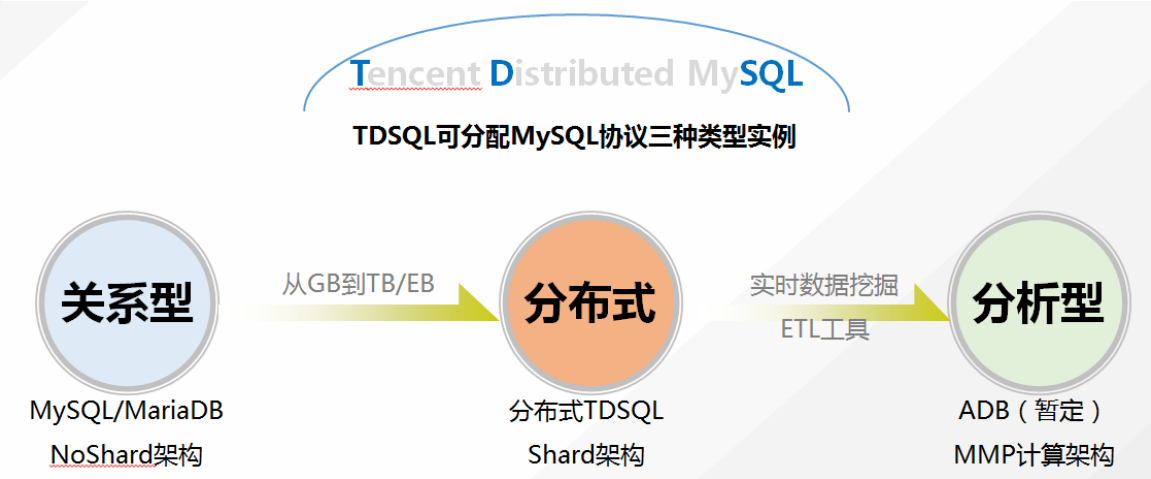
2.5 数据库优化

随着业务的发展，数据库数量越来越多，如何保障所有数据库做到性能优异，业务不出问题；这对数据库管理员（DBA）提出诸多要求，在了解数据库基础运维知识的基础上，还要求 DBA 对 SQL 优化，性能检测，甚至业务逻辑和业务编程的了解。随着业务的快速发展，这种重人工模式意味着 DBA 不可能“照顾”到所有数据库的，那是否能将机器学习、深度学习这样的技术引入到数据库领域，帮助 DBA 更好的优化数据库呢？

3 腾讯云数据库 TDSQL 解决方案

3.1 概述

TDSQL 也是腾讯云数据库团队维护的金融级分布式架构，TDSQL 可以提供公有云、专有云两种部署方案，可以提供**关系型数据库实例、分布式数据库实例、分析性数据库实例**（如下图）。同时 TDSQL 具备虚拟化多租户、强同步复制、线程池、热点更新、内核优化等能力，能够为用户提供事前、事中、事后的全维度安全方案，获得了多项国际和国家认证。

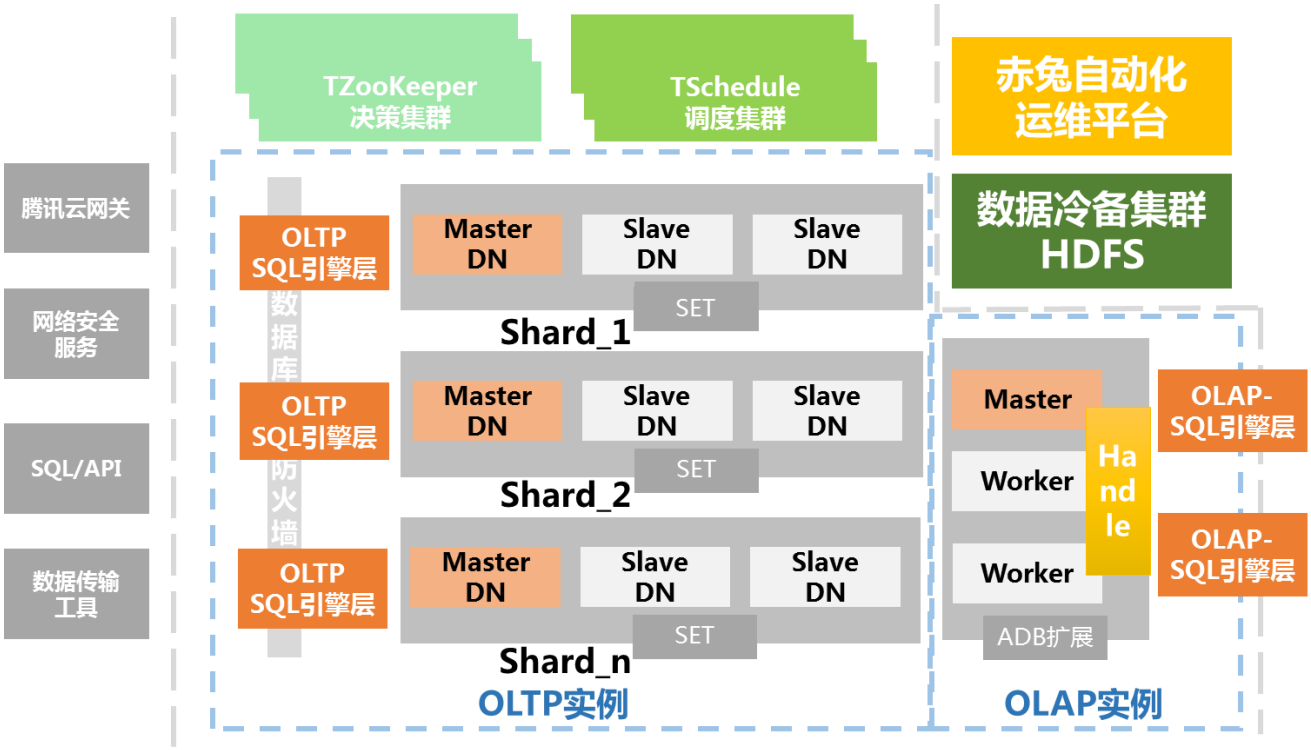


说明:因腾讯云数据库产品品牌变更等原因，您可能会发现腾讯云**公有云**数据库命名会有如下变动：

曾用名	当前名	备注
CDB for TDSQL	TencentDB for MariaDB	主从架构关系型数据库
DCDB	TencentDB for TDSQL	水平拆分的分布式数据库

3.2 产品架构

TDSQL 采用分布式集群架构（如下图），这种集群架构具有较高的灵活性，简化了各个节点之间的通信机制，也简化了对于硬件的需求。这不仅意味着 TDSQL 的关系型实例、分布式实例、分析性实例可以混合部署在同一集群中，也意味着即使是简单的 x86 服务器，也可以搭建出类似于小型机、共享存储等一样稳定可靠的数据库。



实例：从业务视角看到的一个具有完整能力的数据库；

分片(Sharding)：是由数据库节点组（SET）和 SQL Engine（SQL Engine）和支撑系统组成一主多从数据库，也是水平拆分后承载数据的基本单元；

节点组（SET）：由数据库节点（DataNode）组成的，通常包括一个主、从节点的集合。

说明：云数据库支持虚拟化多租户能力，节点即可以是物理节点（一台物理设备），也可以是逻辑节点（一台物理设备的一部分资源）。

SQL 引擎层（SQL Engine）：账号鉴权、管理连接、SQL 解析、分配路由的 SQL Engine 模块；SQL Engine 可以混合部署在数据库节点（DataNode）之上，也可以独立部署在一台物理机中。SQL Engine 也是采用分布式架构设计，提供并行负载和高可用容灾能力；

调度集群、决策集群：作为集群的管理调度中心，主要保证数据库节点组、接入 SQL Engine 集群的正常运行；

- **调度集群（Scheduler）：** 帮助 DBA 或者数据库用户自动调度和运行各种类型的作业，比如数据库备份、收集监控、生成各种报表或者执行业务流程等等，TDSQL 把 Schedule、Zookeeper、Oss（运营支撑系统）结合起来，通过时间窗口激活指定的资源计划，完成数据库在资源管理和作业调度上的各种复杂需求，Oracle 也用 DBMS_SCHEDULER 支持类似的能力。
- **决策集群（ZooKeeper）：**在 TDSQL 中，它的主要功能是配置维护、选举决策、路由同步等，ZooKeeper 支撑数据库节点组（分片）的创建、删除、替换等工作，集群部署要求大于等于 3 组且跨机房部署。

TDSpark 节点：基于 Spark 扩展的计算节点，采用只读的方式与 SET 连接，以 JDBC 的方式获取数据。

赤兔运营平台（chitu）：基于 TDSQL 定制开发的一套综合的业务运营和管理平台，将数据库的管理特点，将网络管理、系统管理、监控服务有机整合在一起。

3.3 多租户的云数据库

TDSQL 是基于“云”的数据库，这使得 TDSQL 可以多租户申请实例，实现租户共享同一堆栈的软硬件资源的能力。多租户给数据库能够提供灵活的、具高可伸缩性的基础架构以保证不同负载下的性能。为每个租户配置一个实例，利用统一资源调度与迁移，能够很好的

实现架构扩展，并且能够解决峰值平衡的问题。即可以应用在公有云环境，为不同企业、个人提供服务。

说明: 这里的租户即可以表示不同的企业、个人；可以是同一个企业内不同的业务团队。在云计算领域，云数据库虚拟化是介于“虚拟操作系统”和“共享软硬件资源”之间的一种方案，由云数据库自身的虚拟隔离技术来做管理。这与直接在虚拟主机上安装数据库镜像的方案不同，但毋庸置疑的是，云数据库给每个租户创建一个独立的数据库是隔离的，租户之间的实例不会互相影响。

4 产品功能

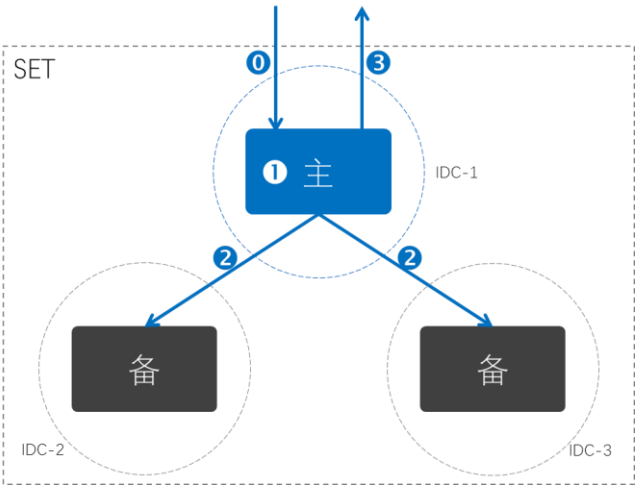
4.1 高度兼容 MySQL

关系型实例：即可以理解为 MySQL/MariaDB 主从高可用架构的数据库，因此其完全兼容 MySQL/MariaDB。

分布式实例：我们的设计理念是让用户像使用普通 MySQL 一样使用分布式数据库。因此 TDSQL 设计淡化水平拆分的概念，无需用户手动去配置分表逻辑，无需用户额外去部署管理中间件，只需要在建表是指定分表关键字即可。分布式实例也高端兼容 MySQL，您可以用连接 MySQL 的方式去连接 TDSQL 的分布式实例，可以使用熟悉的对象映射框架使用 TDSQL 的分布式实例。

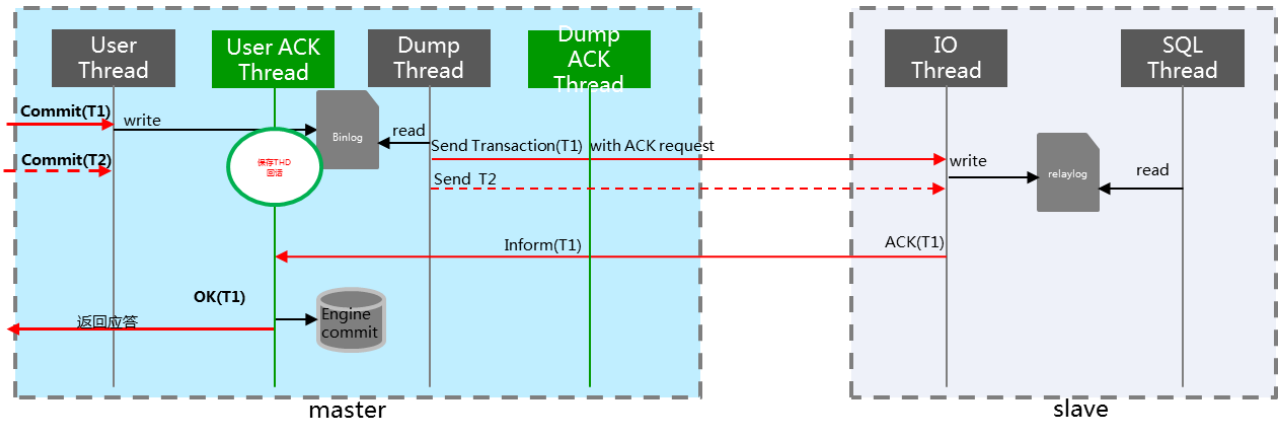
4.2 强同步复制（MAR）

由于数据库中记录了数据，想要通过高可用架构实现切换，数据必须是完全一致且同步的，所以**数据同步技术是数据库高可用方案的基础**，通常数据同步的流程如下图

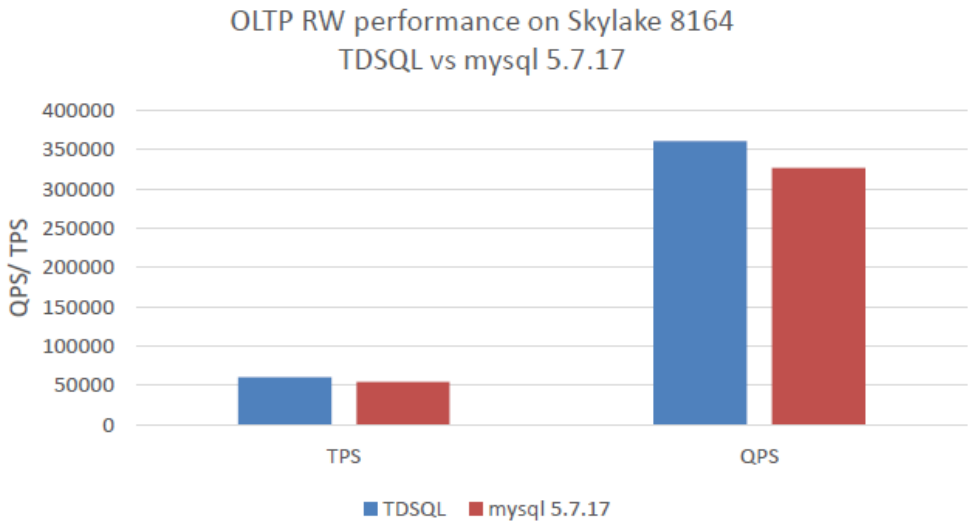


当前，开源 MySQL 数据库数据复制包括**异步复制、半同步复制**两种。这两种复制技术的主要问题是，节点故障时，有可能导致数据丢失或错乱。而且，这类复制技术以串行复制为主，性能相对比较低。而腾讯自主研发了的基于 MySQL 协议的**并行多线程强同步复制方案（Multi-thread Asynchronous Replication，MAR）**，在应用发起请求时，只有当从节点(Slave)节点返回成功信息后，主节点（Master）节点才向应用应答请求成功（如下流程图）；这样就可以确保主从节点数据完全一致。

说明：使用“强同步”复制时，如果主库与备库自建网络中断或备库出现问题，主库也会被锁住（hang），而此时如果只有一个主库或一个备库，那么是无法做高可用方案的。（因为单一服务器服务，如果故障则直接导致部分数据完全丢失，不符合金融级数据安全要求。）因此，TDSQL 在强同步技术的基础上，提供强同步可退化的方案，方案原理类似于半同步，但实现方案与 google 的半同步技术不同。



另外，TDSQL 强同步将串行同步线程并行化，引入工作线程能力，大幅度提高性能；对比在跨可用区(IDC 机房，延迟约 10~20ms)同样的测试方案下，我们发现 MAR 技术性能优于 MySQL 5.6 的半同步约 5 倍 ,优于 MariaDB Galera Cluster 性能 1.5 倍 ,在 OLTP RW(读写混合 ,主从架构) ,是 MySQL 5.7 异步的 1.2 倍(如下由**英特尔®**技术团队测试的性能图):



为进一步验证强同步数据一致性，我们在每秒插入 2 万行数据的场景下，直接杀掉主机数据库进程，并在切换备机后导出流水做对比，发现数据完全一致。

```
plancklin@mmbiztdsqlsz1[qq]:~> awk -F"|" '{if($1!=$2){print $0}}' /tmp/export_data.lst
```

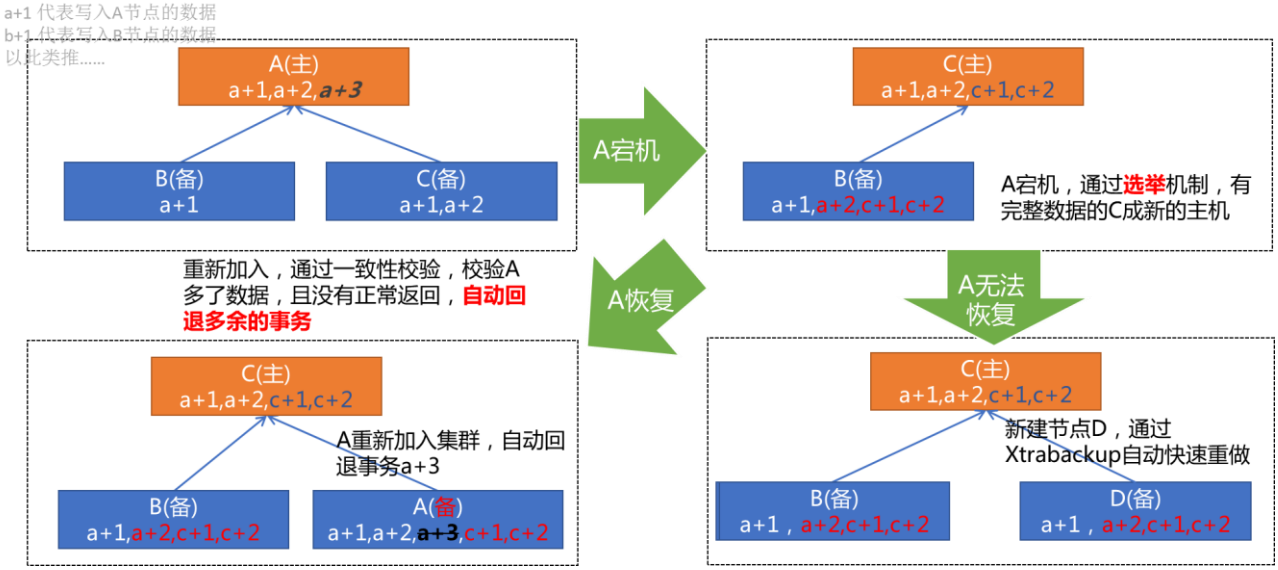
```
plancklin@mmbiztdsqlsz1[qq]:~> wc -l /tmp/export_data.lst /tmp/source_data.lst
   520000 /tmp/export_data.lst
   520000 /tmp/source_data.lst
  1040000 total
plancklin@mmbiztdsqlsz1[qq]:~>
```

4.3 自动故障转移与恢复

在生产系统中，通常都需要用高可用方案来保证系统不间断运行；数据库作为系统数据存储和服务的核心能力，其可用要求高于计算服务资源。目前，TDSQL 高可用方案通常是让多个数据库服务协同工作，当一台数据库故障，余下的立即顶替上去工作，这样就可以做到不中断服务或只中断很短时间，该方案简称主从高可用，也可以叫做主备高可用。在普通的主从高可用基础上，TDSQL 支持：

- 支持故障自动转移 ,集群自动成员控制 ,故障节点自动从集群中移除 ;如果是实例级的主从切换，换后 VIP（虚拟 IP）不变；基于强同步复制策略下，主从切换将保证主从数据完全一致，可满足金融级数据一致性要求。
- 支持故障自动恢复，承载分片的物理节点故障，调度系统自动尝试恢复节点，如果原节点无法恢复，将在 30 分钟内自动申请新资源，并通过备份重建（Rebuild）节点，并将节点自动加入集群，已确保实例长期来保持完整的高可用架构。
- 每个节点都包含完整的数据副本，可以根据 DBA 需求切换；
- 支持免切设置，即可以设置在某一特殊时期，不处理故障转移。
- 仅需 x86 设备，且无需共享存储设备即可支持；
- 支持跨可用区部署，实例的主机和从机可分处于不同机房（无论是否同城），数据之间通过专线网络进行实时的数据复制。本地为主机，远程为从机，首先访问本地的节点，若本地实例发生故障或访问不可达，则访问远程从机。若配合腾讯 VPC 网络环境下，可支持同城双活架构，即业务系统可以直接在两个中心读写数据库。跨可用区部署特性为 TDSQL 提供了多可用区容灾的能力，避免了单 IDC 部署的运营风险。

TDSQL 的每一个分片都支持基于强同步的高可用方案，主数据库故障时将自动选举出最优备机立即顶替工作，切换过程对用户透明，且不改变访问 IP。并且对数据库和底层物理设备提供 7X24 小时持续监控。发生故障时，TDSQL 将自动重启数据库及相关进程；如果节点崩溃无法恢复，将通过备份文件自动重建节点（如下图）：



4.4 弹性的扩展能力

4.4.1 概述

TDSQL 基于分布式架构和多租户方案，天生具有良好的弹性。这意味着数据库实例的并发性能、处理能力、存储容量可线性增长。

4.4.2 集群的弹性扩展能力

如 3.2 章节架构图所示，TDSQL 是由一系列数据库节点组组成。集群的整体承载规模取决于集群中所有设备的总规模。若集群性能不足以支撑，可以通过更换更高配置的硬件、或增加新的节点予以扩展。

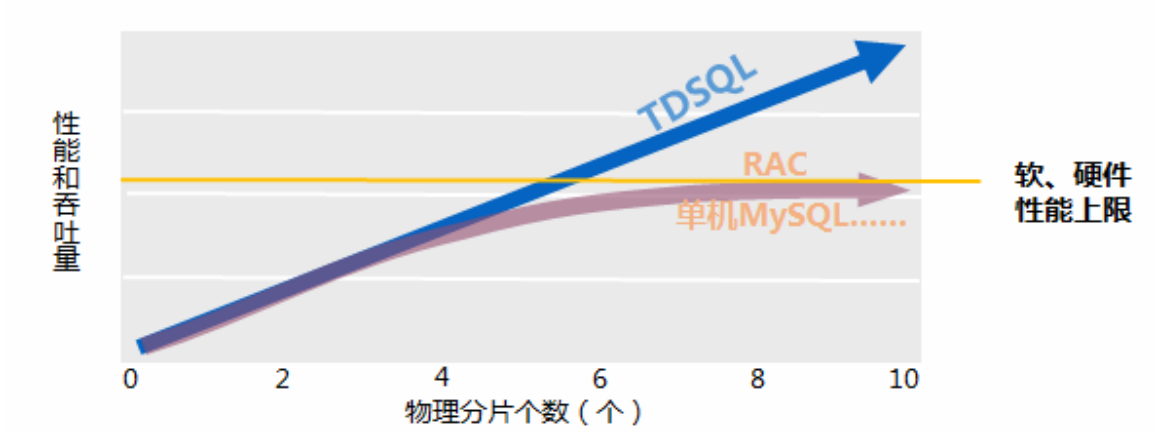
为便于 DBA 操作，TDSQL 的赤兔运营平台提供自动化的实例迁移方案，DBA 只需在设

备初始化后，在系统中点击即可完成资源上线、集群扩容、实例迁移等操作。

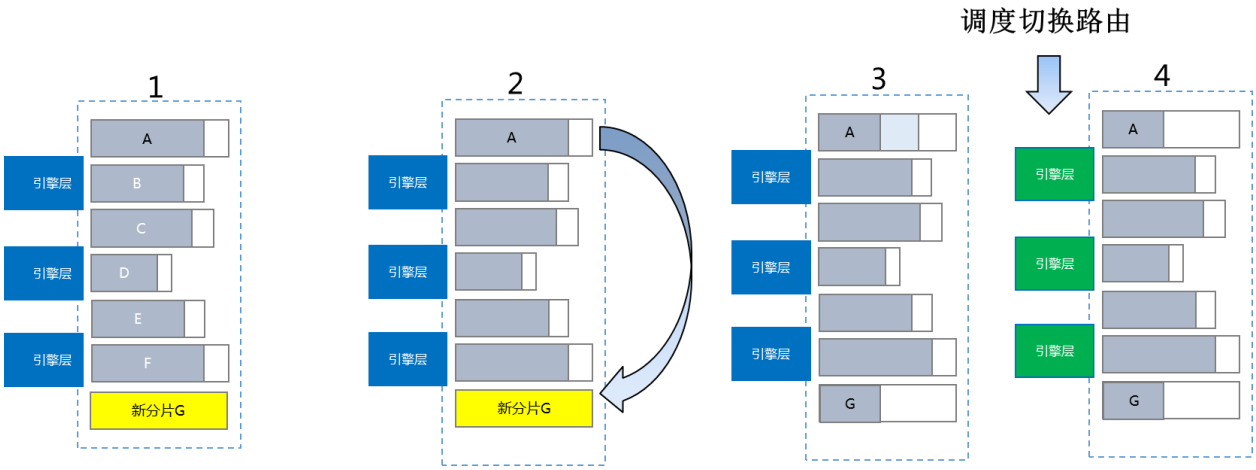
4.4.3 实例的弹性扩展能力

如果是关系型实例，除最大规格实例外均提供无缝升级功能。当遇到性能瓶颈时，只需在页面上通过鼠标点击操作，一键升级到更高性能和容量的实例规格，升级过程不影响业务正常访问和使用，并可指定在低谷期切换，以实现快速、平滑扩容，满足业务快速发展需要。

如果是分布式实例，由于其采用分布式架构，水平拆分逻辑，性能和容量均可以随分片的数量增长而线性增长（如下图）。由于分片数据可能存在不均衡情况，TDSQL 提供新增分片，或扩容单个分片等的扩展方案，相关扩展方案仍然只需要在控制台上简单操作，即实现快速、平滑扩容。



TDSQL 的分布式实例扩容，主要是采用腾讯自研的自动再均衡技术（Rebalance）保证自动化的扩容和稳定，以新增分片为例，扩容过程如下下图：



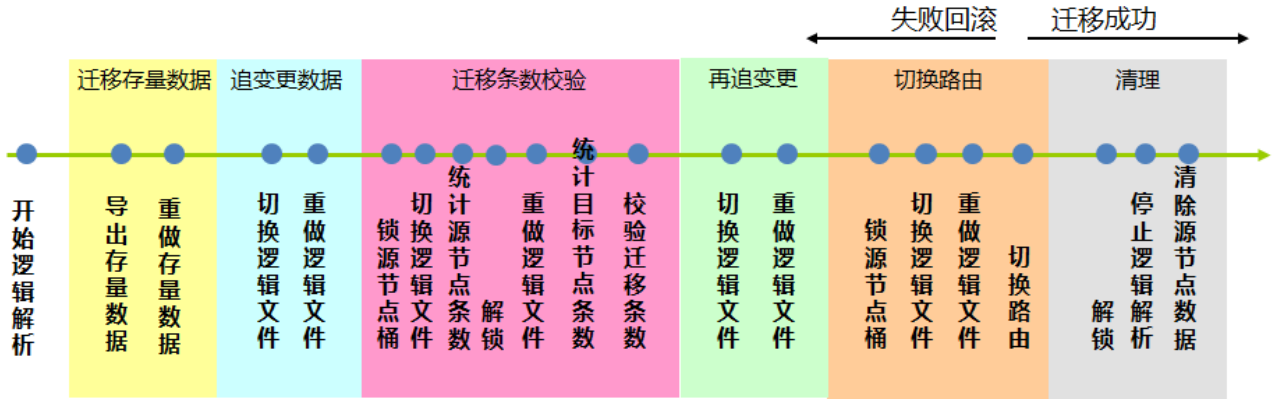
1) 控制台点击扩容 A 分片后，系统计算需要搬迁的数据，并开始配置 G 节点；

2) G 节点配置完成后，将 A 节点需要迁移的数据（通过从机）同步到 G 节点；

3) 数据完全同步后，AG 开始互相校验数据（存在 1~几十秒的只读），但整个实例不会停止运行；

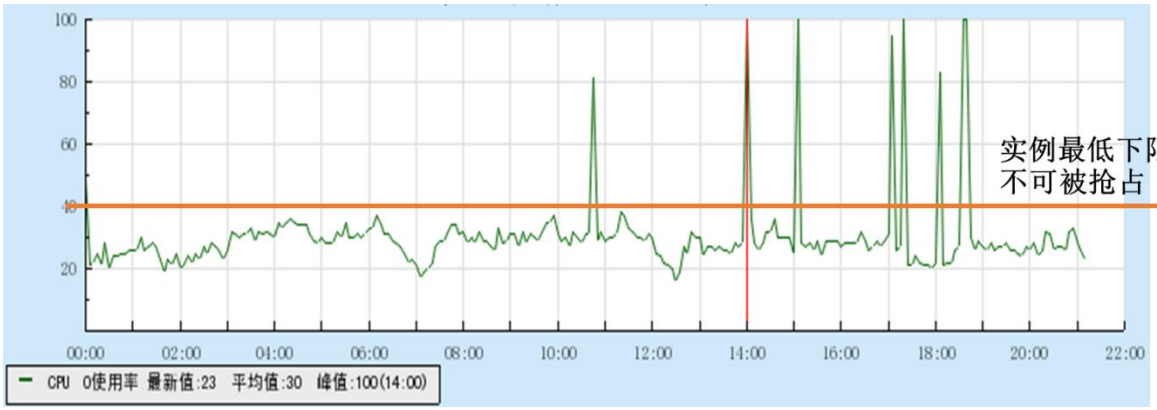
4) 调度通知 SQL Engine 切换路由，完成后将 A/G 节点置位正常状态，A 进入慢速删除状态。

为确保业务不停以及数据一致性，TDSQL 的整个迁移过程采用迁移存量数据、迁移增量数据、数据检验、再追增量、切换路由、清理六个步骤循环迭代进行。该能力经过腾讯内外海量业务迁移的检验，至今未发生过一次数据异常错误或全集群停机。



4.4.4 闲时超用技术

虚拟化让多个租户的业务共享物理设备性能，而传统隔离方案严格限制了每个租户实例的性能大小。这种限制方案很公平，但没有考虑到业务特点：大多数业务仅在一天（一月）的少数时刻有较大的业务压力（如下图）： 该业务日 CPU 平均使用率仅 30%，而一天中仅存在 7 次业务压力较大，CPU 使用率在 80%~100%。虽然云能够基于弹性扩容，然而普通的弹性方案在这种突发性的压力面前，仍然无能为力——可能当您反应过来，您的业务峰值已过；最终，您还得基于业务峰值配置实例，浪费实例性能。



闲时超用技术，即在**绝对保证每个实例预分配性能下限的基础上**，允许实例使用超过预分配的性能。举个例子：假定 A 实例承载新闻业务，B 实例是承载游戏业务，A、B 实例被分配到一台物理设备中，A 可以在 B 的空闲时间，抢占（有限的，并发全部）一部分空闲性能。当然，A、B 同时面对峰值时，系统会确保 A、B 两实例底线的性能需求。

相对于传统的方案，闲时超用是一种更加灵活的性能隔离方案，让您的业务在面对偶然性峰值时也能游刃有余。也经常用于实例之间性能的削峰填谷，节省成本。当然，在集群中实例相对较多分配较均衡的情况下，或已经预知实例之间可削峰填谷的情况下，闲时超用有较大意义。

但若您的多个实例峰值点接近，开启闲时超用就不合适了。此时可以在赤兔运营平台中，

关闭该功能。

4.4.5 四种读写分离方案

TDSQL 默认支持读写分离能力，架构中的每个从机都能支持只读能力，如果配置有多个从机，将由 SQL Engine 集群（SQL Engine）自动分配到低负载从机上，以支撑大型应用程序的读取流量。我们提供多种读写分离方案供您选择 ,且您无需关注若干从机是否完全存活，因为系统将根据策略自动调度。

- **只读帐号（推荐方案）:** 您仅需要在创建帐号时，标记为只读帐号，系统将根据只读策略向将读请求发往从机；只读策略可以根据主从延迟等维度进行灵活配置。
- **/*slave*/注释（推荐方案）:** 您可以在编程过程中，通过注释/*slave*/，系统将把该条语句发往从机，常用于编程阶段将特殊的读逻辑嵌入代码。
- **全局自动读写分离 :**您可以开启全局自动读写分离 ,该配置会自动将 SQL 中的读请求发向从机，且能识别事务、存储过程中的读语法并灵活处理。当然如果从机延迟较大，全局自动读写分离并不具备策略。
- **只读实例 :** 您也可以自建或申请只读实例，只读实例是专用于读请求的一种实例，不参与高可用切换。

读写分离由此为您的应用提高总的读取吞吐量。通过多种只读方案的组合，您可以配置出复杂的只读方案，以满足您各种业务需求和开发的灵活性。

4.4.6 热点更新

在“秒杀”和“限时抢购”等这样的场景下，大量的用户在极短的时间内请求大量商品。而体现在 MySQL 数据库中，同一商品在数据库里肯定是一行存储，所以会有大量的线程来

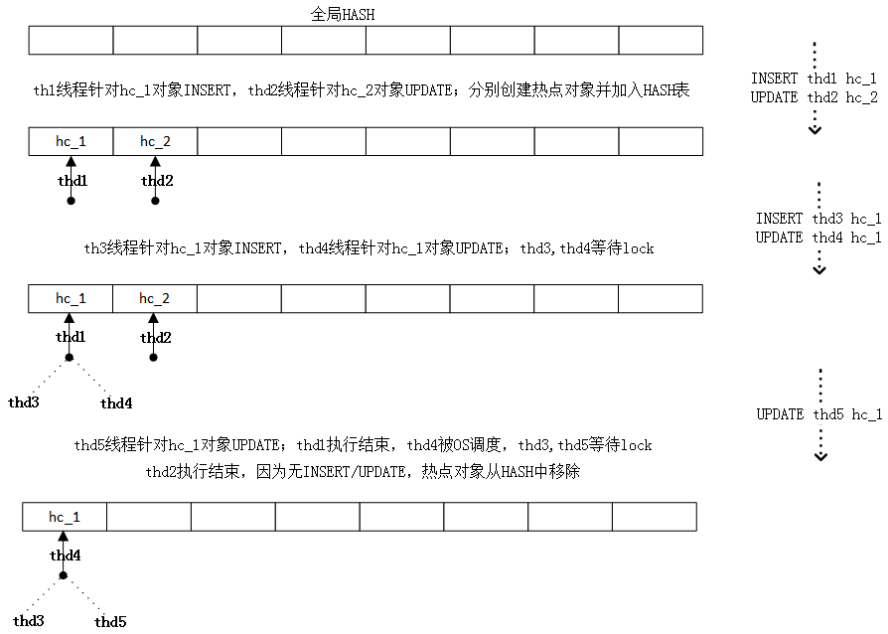
竞争 InnoDB 行锁，当并发度越高时等待的线程也会越多，TPS 会下降 RT 会上升，数据库的吞吐量会严重受到影响。这会导致什么问题呢？

- 单个热点商品会影响整个数据库的性能，即 1 个商品做秒杀，影响整个平台性能和稳定性。
- 数据不一致，如 100 个库存结果卖出去 101 个。

业内的通常采用引入多层架构，如热点缓存 cache，热点库等方案。当然，这一类方案维护成本略高，且如果 cache 被击穿，则容易带来雪崩问题。

而 TDSQL 的目标是让业务用尽量少的修改量（增加几个关键字的使用），便可以快速支撑热点更新功能，以为类似于秒杀，限时抢购等业务场景服务。同时对于已用缓存、热点库的场景下，在为业务进一步提高性能，减少故障发生概率，减少 cache 击穿带来的雪崩风险，提高平台整体稳定性。我们的解决方法如下：

在 SQL 层，通过一个全局 Hash 表存储有 INSERT/UPDATE 请求的热点对象，其大小与热点对象上限相等。INSERT/UPDATE 请求到达时，先查找 Hash 表中有无对应的热点对象，有就获取 lock，会被阻塞；没有该热点对象，那么创建该热点对象，如果热点对象达到上限，那么返回错误，如果创建成功，那么持有 lock 并添加到 Hash 表。成功获取到热点对象 lock，可以继续执行 INSERT/UPDATE，否则等待 lock 被释放和系统调度到该线程。INSERT/UPDATE 返回后，释放热点对象 lock，使得后续线程可以执行，如果后续线程不再 INSERT/UPDATE 改热点对象，将其从 Hash 表移除（如下图）。



4.5 信息安全保障

4.5.1 多项国家或国际认证

TDSQL 现已代表腾讯云云数据库通过多项国家或国际认证，包括但不限于：ISO22301 认证、ISO27001 认证、ISO20000 认证、ISO9001 认证、可信云服务认证、信息安全等级保护（三级或以上）、CSA STAR 认证、PCI DSS 1 级服务提供商、SOC 审计、ITSS 云服务增强级认证等。

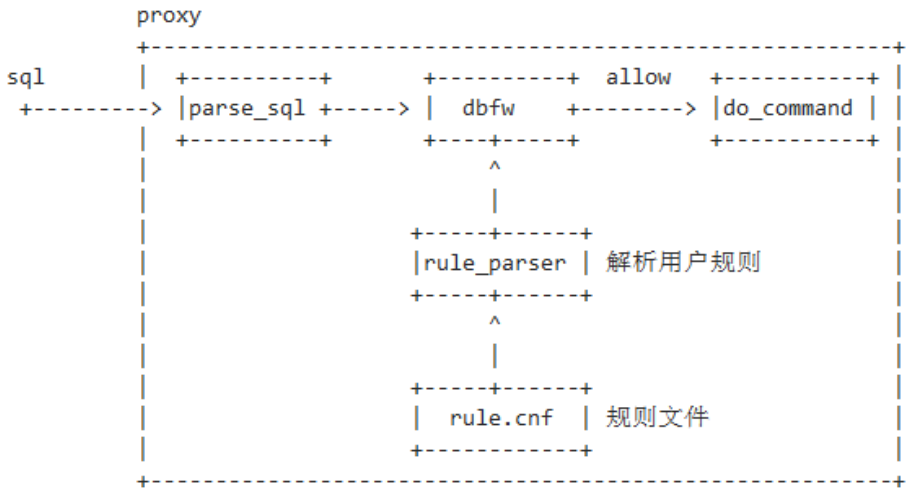
4.5.2 数据安全加密

TDSQL 支持表空间加密（透明加密）和连接加密（SSL 连接加密）。对于没有 KMS（腾讯密钥管理服务）的场景，TDSQL 支持密钥环服务，使内部服务器组件和插件能够安全地存储敏感信息以备以后检索，该服务包含了一系列 API 供加密功能调用密钥服务。

说明: TDSQL 仅在专有云及 MySQL5.7 或更高版本的内核支持加密能力。

4.5.3 SQL 防火墙

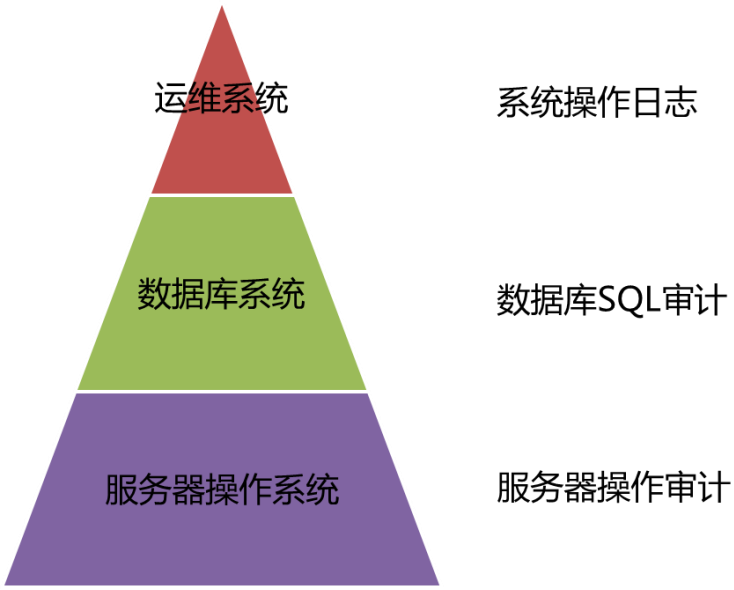
SQL 防火墙是对用户发送的 SQL 进行语法解析，过滤非法的 SQL 的一种安全能力。其与 SQL Engine 配合（如下架构图），可以对用户预先定义的一些非法 SQL 进行判断，从而对非法 SQL 进行过滤、阻断，有效的预防 SQL 注入或一些恶意非法攻击。



说明：SQL 防火墙可以配合 WAF 等一起使用。考虑到业务情况和 SQL 复杂性，目前 TDSQL 的 SQL 防火墙暂未预存规则。

4.5.4 全维度的安全审计

安全审计是最重要的一种事后追溯手段，例如国家等级保护（三级）明确要求（7.1.3.3）明确要求信息系统支持审计能力。而全维度安全保障的云数据库系统来说，TDSQL 包括三个个层面的审计能力（如下图），为用户提供完善的安全保护。



其中，系统操作日志是赤兔运营平台自带的安全能力。数据库 SQL 审计是由腾讯云自研的数据库审计系统完成，公有云默认配置，专有云为选配。服务器操作审计是腾讯云自研的铁将军系统提供，公有云默认配置，专有云为选配。

说明：在专有云中 SQL 审计、服务器审计是选件选配。

4.5.5 内核级安全策略

TDSQL 也在数据库内核层面提供了多种安全方案并开源，部分功能也已获得社区认可，在新版本中使用腾讯云提供的方案。此处以列举的方式，列举 TDSQL 的一些内核安全手段，例如：

- **慢速删除：**当用户执行 drop table 或者 alter table ... drop partition 时，数据库不是立刻删除表空间文件，而是将这些文件重命名并且在后台逐步缩小这些文件并最终删除。慢速删除可避免一次性删除巨大的表空间文件给服务器的文件系统带来突增的 IO 负载，导致系统出现波动。
- **防止误删元数据：**只允许通过规定登录方案的授权用户删除存储元数据的库表，以便防止用户用户误操作导致业务不可用。

- **禁止非授权用户安装插件：**虽然数据库提供了标准的接口允许用户实现自定义的功能，但黑客经常利用这个漏洞以实现共计。因此，只允许规定的管理员用户挂载插件。
- **禁止非授权用户访问物理服务器文件系统：**在脆弱性报告中，黑客经常通过 select into out file 、注入文件、路径探测等的方式绕开安全系统，因此我们禁止非授权用户访问物理服务器的目录结构和文件系统……

说明：由于本白皮书篇幅有限，内核安全策略仅列举部分，更多资料 TDSQL 会逐步在腾讯云官网公开。相关内核级更改均已提交社区开源。

4.6 完善的运维能力

4.6.1 赤兔运营平台&云数据库管理系统

当前，云的运维的呈现两极化趋势，一方面是高星级业务的精细化运维，一方面是大量的普通数据库运维需求。基于云数据库运维层面差异化、数量多、变化快等特点，TDSQL 提供了两套平台：赤兔运营平台&云数据库管理系统；以应对不同场景不同客户的需求。

赤兔运营平台：从管理员视角，提供 TDSQL 的全部运维功能，可管理 TDSQL 集群的物理资源、调度决策系统、备份与恢复系统、可用区管理、实例管理、智能性能分析与监控告警、主要工具等，主要用于腾讯内部或企业 IT 运维团队使用。

云数据库管理系统：从租户视角，提供 TDSQL 实例的运维功能，可管理 TDSQL 数据库实例，包括实例申请与退还、系统监控与告警、备份与恢复、性能优化等，主要用于云的客户或业务团队使用。

赤兔运营平台&云数据库管理系统 两个运维平台配合使用，可有效覆盖绝大部分运维需求，极大的释放 DBA 的常规工作，让 DBA 有精力投入到优化业务等工作中。由于使用对象

和设计差异的差异，在不同场景下，管理平台选件会有所不同，如下表格：

	赤兔运营系统	云数据库管理系统
腾讯公有云平台	√ （仅限腾讯工作人员使用）	√
腾讯专有云平台（TCE）	√	√
腾讯企业云平台（TStack）	√	√
标准化单品部署(仅 TDSQL)	√	不提供

4.6.2 典型运维能力盘点

实例管理：用户可以一键申请 TDSQL 的关系型实例、分布式实例、分析型实例；申请后系统将自动创建实例；创建实例后，用户可以在实例列表页面查看、变更配置、隔离并销毁实例等操作。

说明：由于版本规划原因，您暂时无法在腾讯公有云中申请到TDSQL的分析型实例。

系统监控与告警：系统提供多种方法对数据库实例、决策调度系统、备份系统、管理系统自身的性能和运行状况进行跟踪。可配置告警策略，对异常进行告警，当超过阀值时提供电话、短信或邮件告警。

参数管理：用户可以利用管理控制台的参数设置管理数据库引擎配置，一组数据库参数包括一系列自定义的引擎配置值，这些配置应用于实例中的数据库。如果用户在创建数据库实例时，不修改参数设置，TDSQL 会采用系统分配的默认参数。

备份与恢复：TDSQL 提供将数据库定时备份到指定存储位置的能力，备份方案支持物理备份、逻辑备份、增量备份等多种方案。备份系统可支持 HDFS，NAS，COS（腾讯云对象

存储）等方案。如果出现不慎将数据删除、写乱等情况，系统还提供一键恢复到指定时间的“回档”功能。

在线修改表结构：TDSQL在赤兔运营平台中支持在线修改表结构的方案，由于大表修改的过程会发生锁表，造成当前操作的表无法写入数据，影响用户使用。在线修改表结构方案采用了类似online-schema-change的方案,实现了在线更改表结构的同时，避免了锁表。

 说明:因篇幅限制，本文不在一一列举。

4.7 智能性能分析

智能性能分析（扁鹊系统）是 TDSQL 提供包括数据采集，自动处理，性能检测、SQL 性能检测、业务诊断等多种智能工具的集合，并根据分析结果提供智能优化建议。例如：

- **性能检测与健康评估：**数据库定期采集关键的性能指标并上报，当数据库出现波动时往往伴随着一些监控指标的变动，虽然可以通过监控曲线得出这一结论，但这种排查比较“机械”，当 DBA 管理数据库实例较多时，很难全部照顾到。扁鹊系统将腾讯多年的运营经验，诊断过程通过规则配置的方式沉淀在系统中，并利用深度学习组件不断强化对规则修订。扁鹊系统应用这些规则，结合监控数据，将数据库一段时间内的状态信息进行分析，形成检测报告告知用户。并对其中问题可能的原因和解决方案反馈出来。通过检测报告中的问题，根据业内通用的打分方式对数据库进行打分，方便您快速的解一段时间内 DB 的健康情况。
- **实时检测：**业务常常会遇到 SQL 执行被卡住导致业务频繁超时的问题，通常的原因是慢查询较多 ,或长时间有锁等。扁鹊系统提供了实时检测工具 ,帮助用户快速定位当前的数据库性能问题。
- **慢查询分析：**慢查询常常会引起 DB 响应慢，CPU 消耗过高等多种问题，而引起慢查询的原因常常是因为索引缺失或索引设计不合理导致的，这也常常需要 DBA 手动分析 SQL，结合 SQL 涉及的表结构找出缺失的索引反馈给用户。扁鹊系统可自动分析并合并同类慢查询，并通过可视化方

式展示出来，配合 SQL 优化功能，可有效协助 DBA 优化业务。

- **SQL 优化：**同一条 SQL 语句，不同的写法，是否有索引等，性能可能不一样；扁鹊系统中提供的智能 SQL 优化工具，可以帮助您写出更好的 SQL 或针对性地优化。

说明**:** 因篇幅限制，本文不在一一列举。我们将扁鹊系统与赤兔运营平台进行深度集成，因此您在使用赤兔的过程中，可以无缝使用扁鹊系统。公有云云数据库管理系统正在逐步集成中。

4.8 高度兼容 MySQL 语法

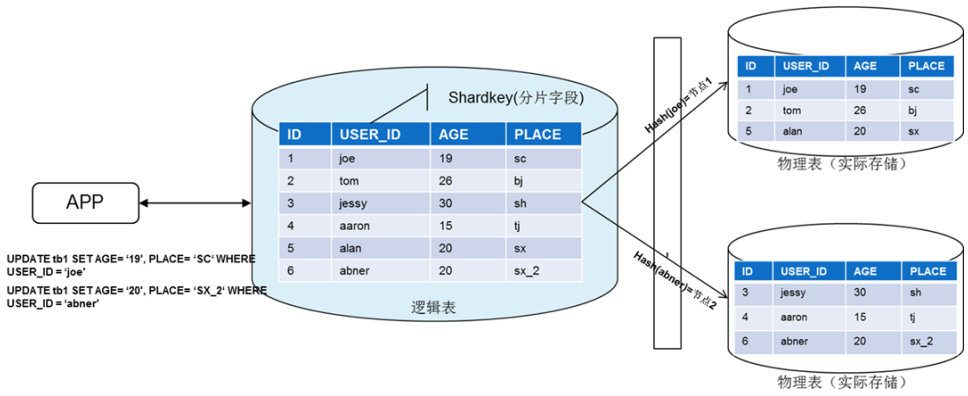
4.8.1 自动拆分原理简介

关系型数据库是一个二维模型，数据的切分通常就需要找到一个分表字段（shardkey）以确定拆分维度，再通过定义规则来实现数据库的拆分。业内的几种常见的分表规则如下：

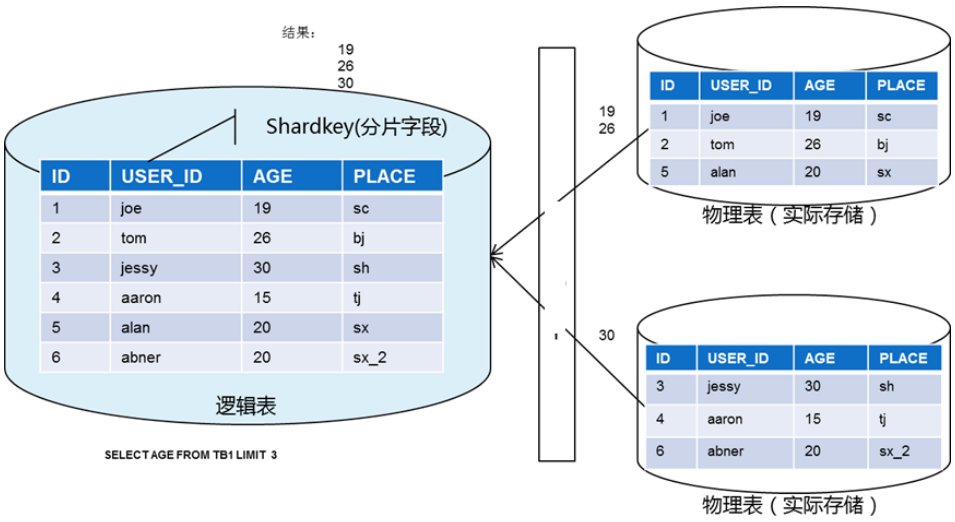
- 基于日期顺序（Time），如按年拆分，2015 年一个分表，2016 年一个分表。
- 基于某字段划分范围（Range），如按用户 ID 划分，0~1000 一个分表，1001~2000 一个分表。
- 基于某字段求模（Hash），将求模后字段的特定范围分散到不同库中。

无论是 Time、Range 都容易导致严重的数据倾斜，即分片之间负载和数据容量严重不均衡。例如，在大部分数据库系统中，数据有明显的冷热特征——显然当前的订单被访问的概率比半年前的订单要高的多——而采用 Time 分表或 range 分表，就意味着很容易出现大部分热数据将会被路由在少数几分片中，而剩下的分片设备性能却被白白浪费掉了。因此，TDSQL 通常采用某个字段求模（Hash）的方案进行分表。因为 Hash 算法的原理能够基本保证数据相对均匀的分散在不同的物理设备中（某些情况下除外，我们将在后续章节进行介绍）。

基于上述原理，TDSQL 分布式实例在创建表的时候，要求 SQL 中显示指定拆分建 shardkey，例如：create table tb1 (user_id int not null,age int not null, place char(20) not null,primary key(user_id, age) ,unique key(user_id, place)) shardkey= user_id;此时，Hash 的过程大致就是，当某条记录(如下图)请求时被发起时，TDSQL 会理解 SQL 语句的含义，然后将拆分键 shardkey (此处为 user_id)的值进行 Hash，根据 Hash 后的值和 SQL Engine 中预设的路由表进行匹配，然后将 SQL 路由到对应分片（或指定节点）执行。



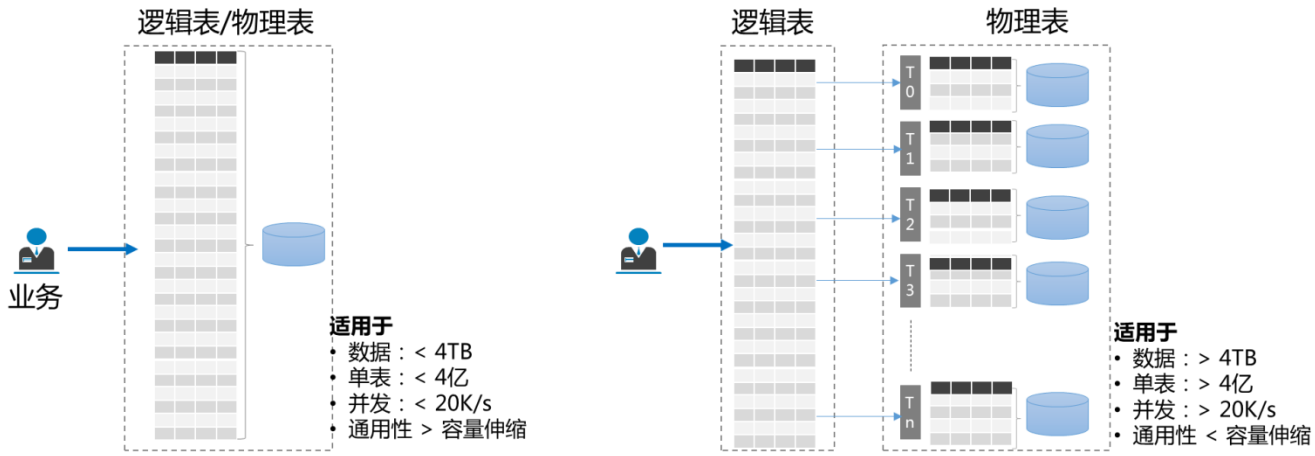
如果一个查询 SQL 语句的数据涉及到多个分表，此时 SQL 会被路由到多个分表执行，TDSQL 会将各个分表返回的数据按照原始 SQL 语义进行合并，并将最终结果返回给用户。



查询 SQL 在处理逻辑上，分为两类情况：如果 SQL 有明确 shardkey 值，数据将直接从对应的分片取出，此时效率最高；如果 SQL 没有 shardkey，SQL 请求将发往所有分片，并在 SQL Engine 中聚合在反馈给业务，此时效率会略差。从上述原理来看，查询 SQL 中含有 shardkey 值比不含 shardkey 值效率将会更高。

4.8.2 逻辑表

TDSQL 对应用来说，读写数据完全透明，对业务呈现的表实际上是逻辑表。逻辑表屏蔽了物理层实际存储规则，业务无需关心数据层如何存储，只需要关注基于业务表应该如何设计。



TDSQL 为用户提供了三种类似的表：分表，广播表及单表：

- **分表：**是指那些原有的很大数据的表，需要切分到多个数据库的表，这样每个分片都有一部分数据，所有分片构成了完整的数据。（仅分布式架构实例可使用）
- **广播表：**名小表广播功能，设置为广播表后，该表的所有操作都将广播到所有物理分片（set）中，每个分片都有改表的全量数据。（仅分布式架构实例可使用）
- **单表：**即无需拆分的表，又叫做普通表，目前单表都放在第一个物理分片（set）中。

4.8.3 如何选择拆分键

拆分键是在水平拆分过程中用于生成拆分规则的数据表字段。TDSQL 建议拆分键要尽可能找到数据表中的数据在业务逻辑上的主体，并确定大部分（或核心的）数据库操作都是围绕这个主体的数据进行，然后可使用该主体对应的字段作为拆分键，进行分表（该分表方案叫做 groupshard），如下图：

Groupshard 的分表方案，可以确保某些复杂的业务逻辑运算，聚合到一个物理分片内。例如，某电商平台订单表和用户表都是基于用户维度（UserID）拆分，平台就可以很容易的通过联合查询（不会存在跨节点 JOIN，或分布式事务）快速计算某个用户近期产生了多少订单。

下面的一些典型应用场景都有明确的业务逻辑主体，可用于拆分键：

- 面向用户的互联网应用，都是围绕用户维度来做各种操作，那么业务逻辑主体就是用户，可使用用户对应的字段作为拆分键；
- 电商应用或 O2O 应用，都是围绕卖家/买家维度来进行各种操作，那么业务逻辑主体就是卖家/买家，可使用卖家/买家对应的字段作为拆分键；但请注意，某些情况下几个超大卖家占到绝大多数交易额，这种情况会导致某几个分片的负载和压力明显高于其他分片，我们会在后面章节予以说明。

- 游戏类的应用，是围绕玩家维度来做各种操作，那么业务逻辑主体就是玩家，可使用玩家对应的字段作为拆分键；
- 物联网方面的应用，则是基于物联信息进行操作，那么业务逻辑主体就是传感器/SIM 卡，可使用传感器、独立设备、SIM 卡的 IMEI 作为对应的字段作为拆分键；
- 税务/工商类的应用，主要是基于纳税人/法人的信息来开展前台业务，那么业务逻辑主体就是纳税人/法人，可使用纳税人/法人对应的字段作为拆分键；

以此类推，其它类型的应用场景，大多也能找到合适的业务逻辑主体作为拆分键的选择。

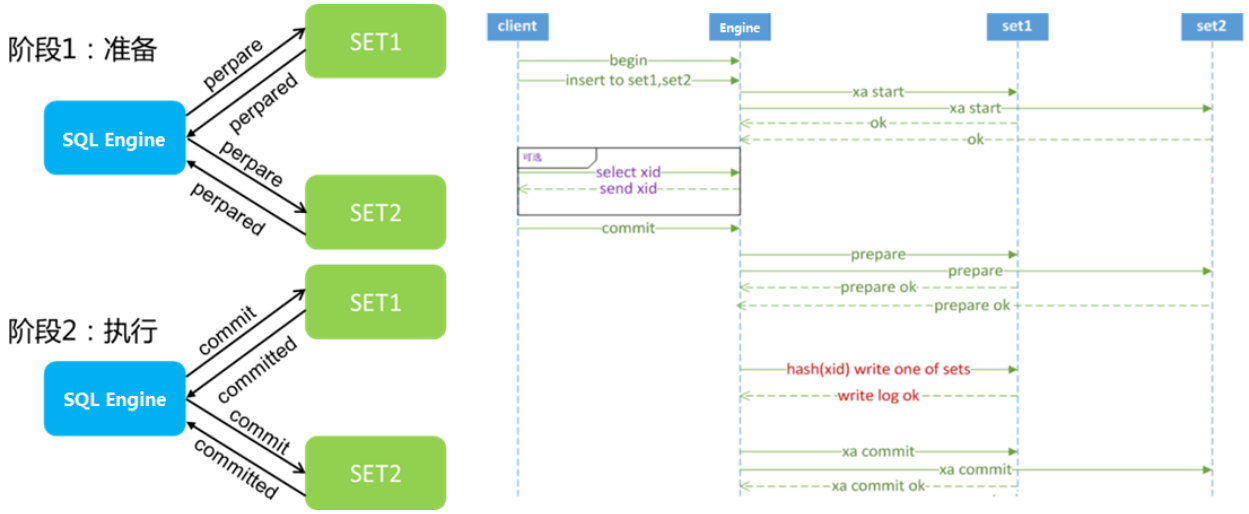
4.8.4 拆分键的限制

为了提高语法解析效率，避免因为 shardkey 设置导致路由错误，TDSQL 规定了拆分键设定的技术限制（更多详情，请参考腾讯云官方文档）：

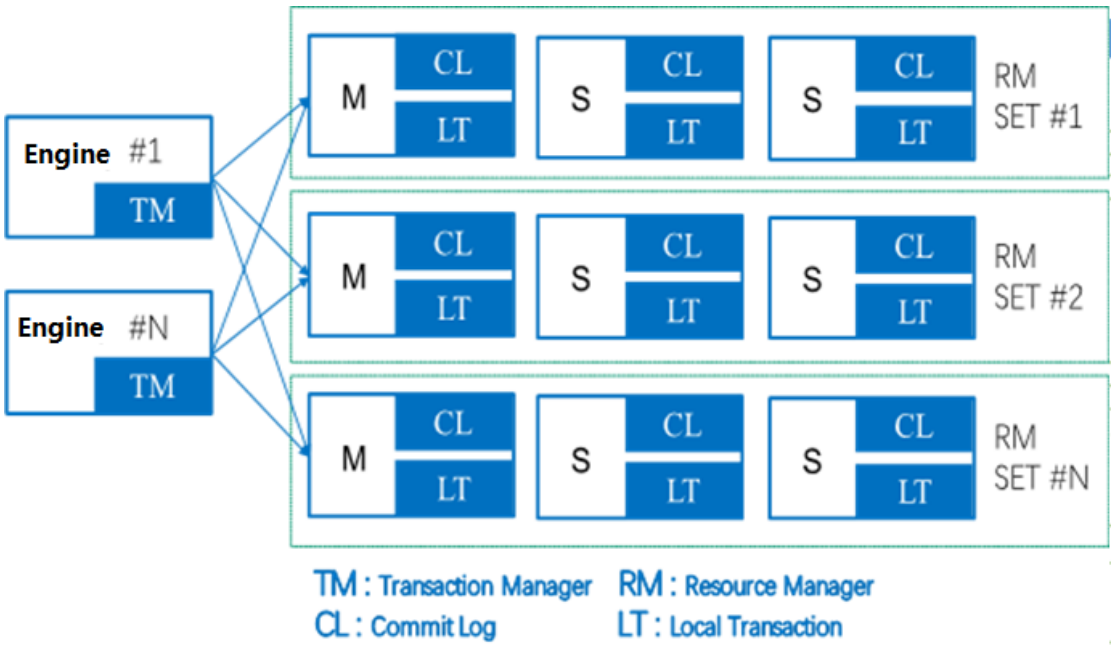
- Shardkey 需要是主键以及所有唯一索引的一部分；
- shardkey 字段的类型必须是 int,bigint,smallint/char/varchar
- shardkey 字段的值不应该有中文，SQL Engine 不会转换字符集，所以不同字符集可能会路由到不同的分区
- 不要 update shardkey 字段的值
- shardkey=a 放在 sql 的最后面
- 访问数据尽量都能带上 shardkey 字段，这个不是强制要求，但是不带 shardkey 的 sql 会路由到所有节点，消耗较多资源

4.8.5 分布式事务

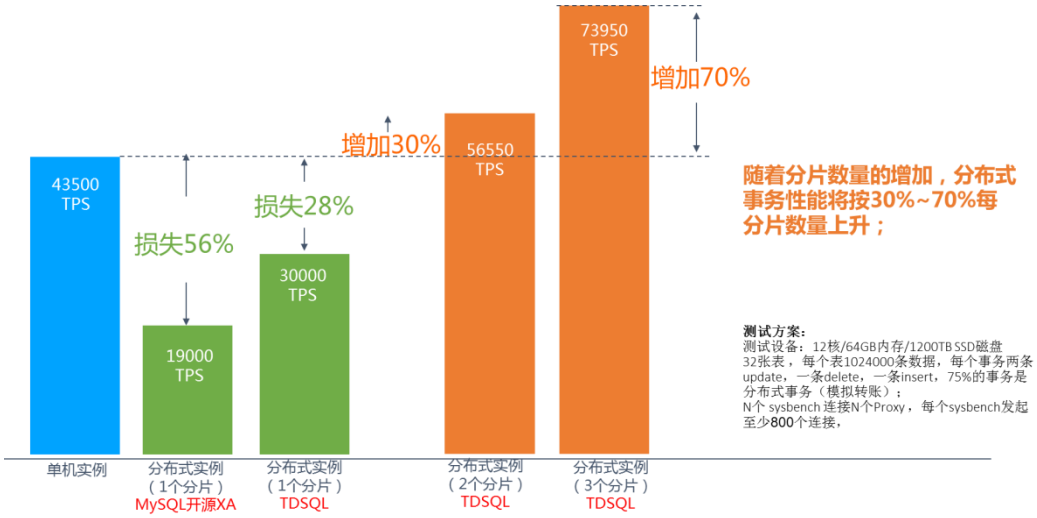
分布式事务，就是一个数据库事务在多个数据库实例上面执行，并且多个实例上面都执行了写入（insert/update/delete） 操作。实现分布式事务处理的最大难点，就是在多个数据库实例上面实现统一的数据库事务的 **ACID 保障**，而这里面最重要的算法就是**两阶段提交**算法。分布式事务能力理论虽然很早就被提出，而业内实际工程化实现和大规模业务验证的产品还较少。



TDSQL 支持分布事务，可以为银行转账、电商交易等业务提供有效支持。当然，分布式事务处理的开销会比单机架构事务处理开销要大一些，使用分布式事务会导致系统 TPS 降低，事务提交延时增大。而腾讯 TDSQL 通过多种优化，提供了高于开源 XA（分布式事务简称）的性能。



由于理论上，一个事务不会操作全部分片，仅操作 1~2 个分片（如转账业务），再加上 TDSQL 的 MPP 架构的原因，因此一个分布式实例多个分片的分布式事务性能可以叠加。



所以是否使用分布式事务要根据实际应用需求来定。数据量非常大或者数据访问负载非常高时，分布式事务会大大降低应用开发难度，TDSQL 每个事务的查询语句的写法与使用单机架构实例完全相同，且获得事务的 ACID 保障。对于涉及跨分片的分布式事务，我们建议业务开发时，平衡性能和开发难度的关系，或将事务拆解，巧妙设计或引入一些等待机制，以优化用户体验。

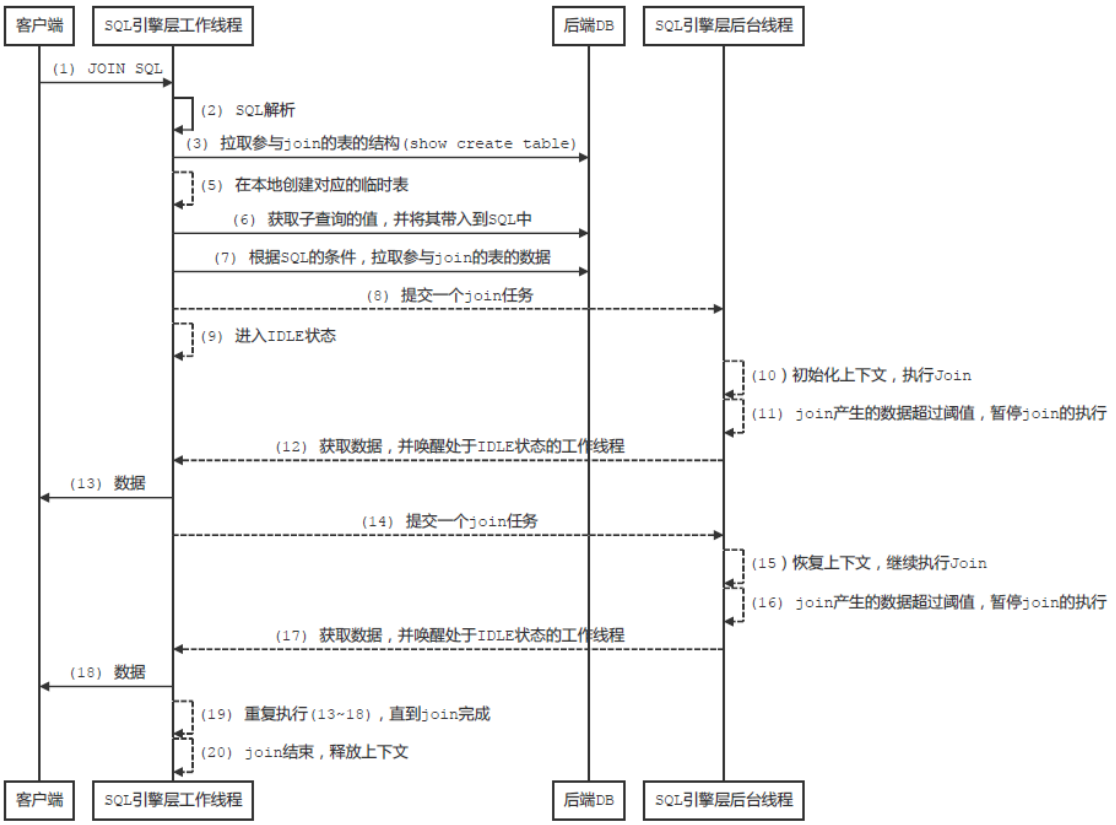
4.8.6 分布式 JOIN

业务逻辑中，经常需要查询两个或多个表中的列之间的关系（JOIN），这在关系型实例（单机架构）上的简单操作，在分布式实例中就比较复杂。由于需要 JOIN 的数据可能分别存储在多个物理节点中，导致 JOIN 过程需要大量网络交互，这导致某些分布式数据库处理 JOIN 请求时，无法提供数据一致性和性能兼得的方案。TDSQL 业内少数几个支持分布式 JOIN，且在大规模业务中验证过的产品。一般来说，分布式 JOIN 分为可下推和不可下推：

可下推 JOIN，是指可在存储层直接 JOIN 的情况，通常包括：

- 同纬度（拆分建）的 JOIN：两张表采用相同的拆分键，例如：SELECT * FROM user JOIN user_order ON user.user_id=user_order.user_id;由于 user 与 user_order 均已 user_id 为拆分键，因此同一用户（user_id）的记录位于同一分片上，JOIN 直接由底层出错了完成。此时性能最高。
- 分表与广播表的 JOIN：由于所有分片中都存在一个完整的广播表副本，因此分表与广播表的 JOIN 也可下推到存储层执行。

不可下推的 JOIN，是指需要由存储层和 SQL Engine 共同完成的 JOIN，通常包括：单表与分表的 JOIN ,分表与分表且不同字段的 JOIN 等。腾讯云优化不可下推的分布式 JOIN，并采用如下的过程执行（如下图）。



另外，分布式实例也支持子查询、函数等复杂语句。

4.8.7 其他特性

数据切分后，原有的关系数据库中的主键约束在分布式条件下将无法使用，因此需要引入外部机制保证数据唯一性标识，这就是全局唯一数字序列（sequence）。

TDSQL 全局唯一数字序列（以下简称 sequence，使用的是 unsigned long 类型，8 个字节长），使用方法与 MySQL 的 AUTO_INCREMENT 类似。目前 TDSQL 可以保证该字段全局唯一和有序递增，但不保证连续性。

4.9 兼容 JSON

TDSQL 的分布式实例、关系型实例已在（MySQL 5.7 内核）已支持 JSON 功能，对比于 MongoDB 目前的三大核心功能：JSON 的灵活性，复制集群保证高可用，sharding 保

证可扩展，TDSQL 均可以支持，基于腾讯云金融级特性，无论是其自身数据强一致、高可用和可扩展也有着完善的解决方案，且能够支持关系型数据库的事务，JOIN 等功能。如果您既希望使用 JSON 类型，又对数据一致性，事务，JOIN 等传统数据库具备的能力也有一定要求的话，TDSQL 将是一个很好的选择。

而通过对比 TDSQL 与 MongoDB 在 JSON 方面支持的区别，我们注意到：

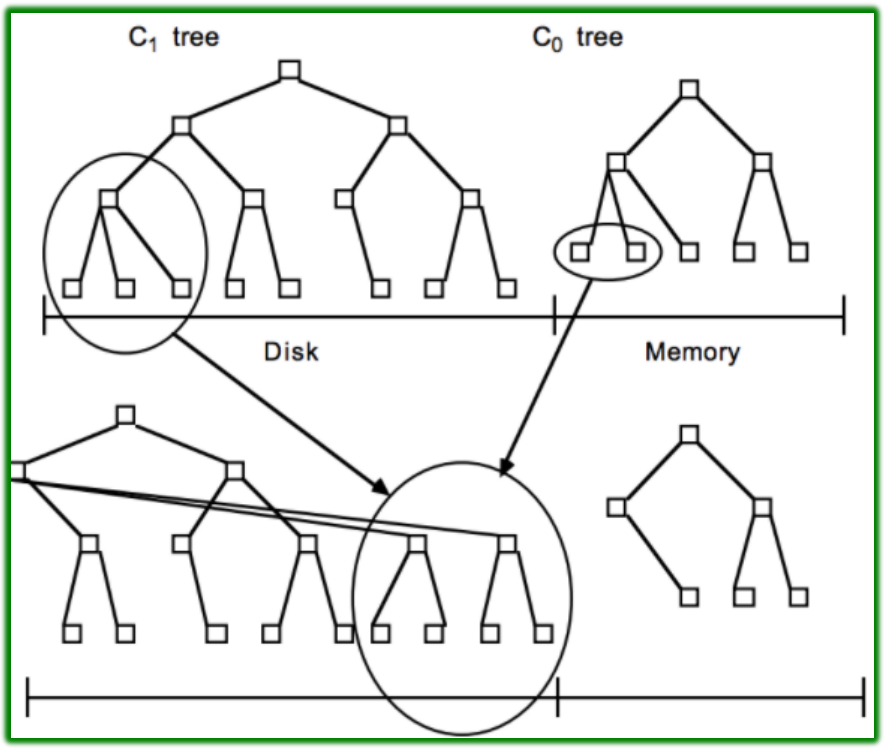
- **JOIN：** TDSQL 支持多表根据 JSON 字段进行 JOIN 操作，MongoDB 只支持多个 unsharded 表 LEFT JOIN。
- **Index：**两者都支持根据 JSON 的某些（int,string）字段建立索引，MongoDB 还额外支持 multikey index 等索引。
- **访问 JSON 内部元素：**两者都有各自完善的语法可以访问到 JSON 内部的各个字段，无需应用层进行 JSON 解析。
- **搜索条件：**MongoDB 提供的搜索和匹配方面的功能更完善，相比之下，TDSQL 需要时刻注意对选择条件进行类型转换后再进行判断，对开发人员来说不是很友好，并且筛选的功能方面也较 MongoDB 稍弱，适用于对 JSON 操作相对简单的应用。
- **写入数据：**两者都可以以方便的写入 JSON 串和更新 JSON 内部的某些字段，但 MongoDB 不支持事务，只有单行操作可保证原子性，多行操作如果需要原子性需要应用层实现两阶段提交。而 TDSQL 的 JSON 操作可以完整的支持事务特性，也支持分布式事务。

4.10 RocksDB 引擎

RocksDB 是由 Facebook 开源的一套基于 LSM 的 KV 存储系统，通常适用于写量超大业务场景，例如物联网，电商订单等场景。目前 RocksDB 已在 TDSQL 的分布式实例和关系

型实例中支持，您可以在创建实例是选择 RocksDB。使用 RocksDB 后，您也可以在建表时显示指定 InnoDB 或 XtraDB 引擎。当然，腾讯 RocksDB 支持 MySQL 协议的语法，支持数据库多版本并非控制(mvcc)，（分布式）事务(2PC)，（分布式）JOIN，主从数据复制，备份恢复等关系型数据库特性。而 RocksDB 强大的写性能优异性主要来自于 LSM Tree 算法。

LSM Tree 是一种将随机写合并为顺序写的算法（如下图），数据以 KV 值有序存储，随机写入数据库时，先以树状结构（tree）组织更新在内存中；由于每一个数据的 KV 值是全局有序的，随着写入的增加，内存的 tree 不断变大（丰富其树状枝丫），当一定程度后，相同层级的 tree 值发出与磁盘的 tree 合并操作。一次性批量写入已经组织好的有序数据，即将随机写入，通过 LSM 组织为顺序写入。这样就大大的提高了写入效率和并发。



4.11 分析型实例 TDSpark

4.11.1 概述

TDSpark 是 TDSQL 推出的为了解决用户复杂 OLAP 需求的解决方案。借助 Spark 平台本身的优势，同时融合 TDSQL 分布式集群的优势，为用户一站式解决 HTAP（Hybrid Transactional/Analytical Processing）需求。

众所周知，MySQL 无法有效应对高计算强度的 OLAP 业务需求，通常需要借助 ETL 工具将数据同步到 OLAP 类数据库中；但该方案将极大的增加系统架构的复杂性。而 TDSpark 深度整合了 Spark Catalyst 引擎，并集成腾讯自研的且 MySQL 协议的 SQL Engine，利用 TDSQL 关系型实例或分布式实例现有数据（通过从机）直接拉取到 Spark 引擎中进行计算。从数据集群的角度看，TDSpark 可以让您直接在同一个平台进行事务和分析两种工作，简化了系统架构和运维。

4.11.2 系统架构

如 3.2 章节架构图，TDSpark 包括如下核心模块：

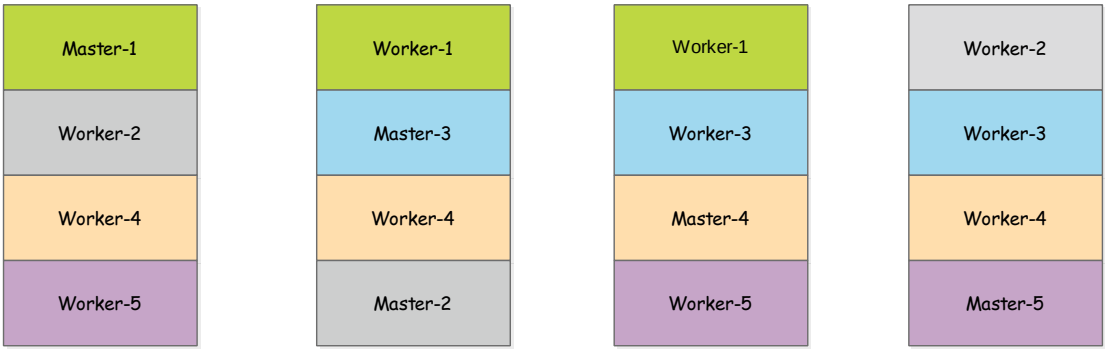
OLAP SQL Engine：专门用于接收并处理客户端发来的 OLAP 类型的 SQL。SQL Engine 不与后端数据库相连，而是与运行在 Spark 集群集群上的 Handle-sever 交互。在获取到客户端发来的请求后，将 SQL 语句以 JSON 的形式发送至 Handle-sever 进行处理，并等待 Handle-sever 的响应。

Handle-server：负责接收 OLAP SQL Engine 发来的 JSON 协议的请求，并给出响应。在接收到 OLAP SQL Engine 的请求后，对 JSON 串进行解析，得到 SQL 语句，同时对 SQL 进行语法解析得到库表名，并封装成独立的 Spark job 提交到 Spark 集群进行计算，待 Spark

得到结果后，将结果集转换成 mysql 协议并返回给 SQL Engine。Handle-server 部署在 Spark 集群的每一台机器上， SQL Engine 可以访问 Spark 集群机器中的任意一台，并且该 server 可以随着 Spark 集群的扩容，动态的增加服务节点。

Spark 集群在执行 job 的过程中，需要连接后端 OLTP 类实例（关系型或分布式实例）的 SQL Engine，以 JDBC 的方式获取数据，在连接后端 OLTP 类实例的 SQL Engine 时，采用只读账户，从从机获取数据，从而降低对主机性能的影响。

而 Spark 集群的资源管理由独立的 Spark-manager 进程完成。Spark-manager 负责 Spark 集群工作节点的购买，资源回收，集群扩容等工作。多租户下，同一组机器资源 Spark 的部署方式如下图所示，每个模块之间资源隔离。



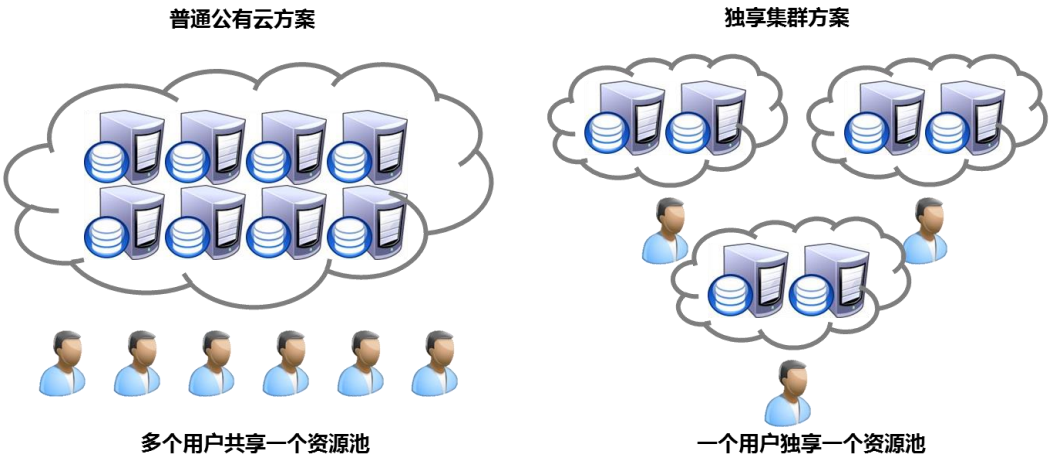
Spark 集群采用 standalone 方式部署，便于集群的扩容。当需要对集群扩容时，只需要增加一个 worker 节点并指向 master，同时将 Handle-server 以 client 方式提交至集群即可。当集群需要缩容时，直接停止需要裁撤掉的 worker 即可。

4.12 物理独享解决方案

有时业务希望在云数据库中，申请一组独立且不与其他业务共用硬件资源池，以实现物理隔离/性能强隔离等目标。TDSQL 支持物理独享解决方案，即可以让您在大的公有云资源池中，申请一个小的物理集群；该方案除资源独立以外，其他云数据库应有的功能完全具备。

让您的业务即能享受公有云的弹性，又具有类似于专有云的独立性。

独享集群与普通公有云的方案可以用下图表示：



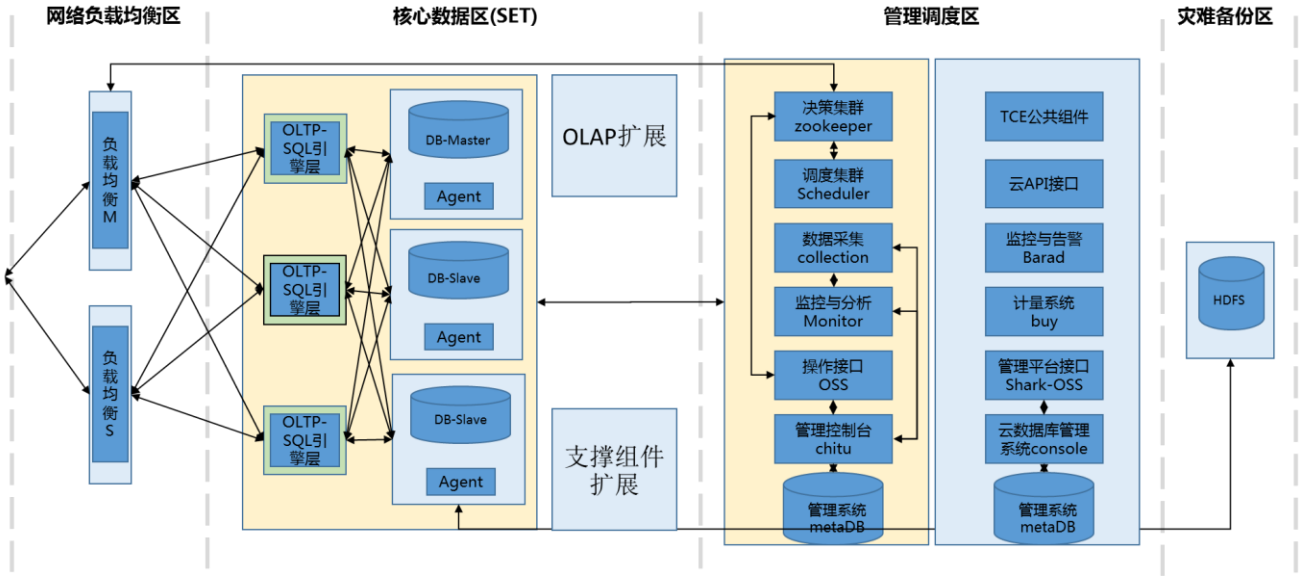
4.13 数据传输工具（选件）

TDSQL 支持选配数据传输工具（DTS），支持 Oracle，Kafka 等与 TDSQL 的互相同步；支持 TDSQL 实例之间的互相同步。同步拓扑支持一对一，一对多，多对一等方式，且支持冥等策略，以确保数据同步一致性。

5 专有云方案简介

5.1 部署架构与软件模块

TDSQL 是分布式架构的数据库，其部署架构和软件模块如下简图，其中浅黄底必须安装，浅蓝色为为可选配模块。



5.1.1 核心模块

SQL Engine/DB：可以混合部署在同一物理机中，也可分离部署；此模块对 CPU 和磁盘性能要求较高，建议采用较高配置的 CPU 和 SSD 存储，并考虑高可用 2 台起部署。

赤兔运营系统：可部署在虚拟机或物理机中，建议为 1/3/5 等基数台部署。

5.1.2 选配模块

负载均衡模块：提供虚拟 IP，数据库负载；目前支持 LVS、腾讯云网关 TGW、腾讯私有网络 VPC 等开源或商用负载；

说明：如果不安装负载均衡模块，业务可以通过访问 SQL Engine 的 IP 和端口访问数据库，并通过连接池管理与多个 SQL Engine 的连接。

云数据库管理系统：云数据库管理系统需与腾讯专有云 TCE 合并安装。用于提供类似于集团云、行业云、政务云等场景下的租户端使用。

说明： 云数据库管理系统的部署依赖于赤兔运营系统。

数据备份系统：目前可以支持分布式文件存储系统 HDFS，腾讯云对象存储 COS，网络

存储 NAS 等。

说明：如果不安装数据备份模块，将影响数据库备份、恢复与回档、备份与日志下载等功能。

OLAP 扩展：指分析型数据库扩展，通常需要 3 台物理机起。

说明：OLAP 的部署依赖于 SQL Engine/DB 模块的部署。

支撑组件：指数据库审计、数据同步等其他功能，若不安装不影响数据库核心功能。具体需求请咨询腾讯相关工作人员。

5.2 建议设备选型

专有云资源池在设计时也应考虑到业务连续性目标和业务发展规划，根据业务性能需求、监管要求、业务性质和服务范围、数据集中程度、业务时间的敏感性、功能的关联性等要素进行业务需求分析，在此基础上评估业务中断可能造成的影响，确定灾难恢复需求，再通过需求决定资源池建设计划。一般来说，我们提供如下参考：

	OLTP型	HTAP	OLAP型	冷数据存储型
	旨在使事务应用程序写入或读取数据； 特点是：逻辑简单/并发高/处理快；	OLAP&OLTP混合的业务，但相对于纯OLAP类业务，逻辑复杂度和数据量较低。 特点是：并非高/处理快/逻辑相对复杂	支持复杂的分析操作，侧重对决策人员和高层管理人员的决策支持； 特点是：逻辑复杂/数据量大/计算量多/过程数据多/计算耗时多	存储不会经常使用的数据，但这些数据需要长期存储； 特点是：数据写入大于读取/长期存储/且需要压缩/单位存储密度的性价比高
常规配置	✓ CPU/内存比可小于1/10 ✓ 内存需留25%冗余 ✓ 磁盘需选择SSD ✓ 万兆网卡	✓ 建议选择多路CPU ✓ CPU/内存比可小于1/5 ✓ 内存需留25%冗余 ✓ 磁盘选择SSD ✓ 磁盘需留25%冗余（可能产生较多临时文件和日志） ✓ 万兆或20GB网卡	✓ CPU配置多路（64core或以上） ✓ 内存建议512GB以上 ✓ CPU/内存比可小于1/4 ✓ 内存需留30%冗余 ✓ 磁盘选择SSD ✓ 磁盘需留30%冗余 ✓ 20GB或以上网卡	✓ 性价比高 ✓ 磁盘存储容量大并稳定

测试环境配置：

组件名称（英文）	组件名称（中文）	功能介绍	数量（台）	建议配置（CPU/内存/磁盘）
LVS	虚拟负载网络	LVS（DR 模式），提供负载访问能力，为每一个数据库实例提供唯一虚拟 IP，可替换为 TGW	选配 1 台	4-8-200（虚拟机）
chitu	WBE 化运维平台	赤兔管理平台，提供 WEB 化管理整个集群的能力	无需	4-8-200
zookeeper/monitor	决策集群	提供高可用和一致性能力	1	4-8-200（虚拟机）
scheduler/manager	调度集群	提供主备切换/扩缩容/资源管理等能力	无需	4-8-200
DB	数据库模块	包括 Proxy，agent，MySQL，存储和计算核心节点	3*N	16-64-1000（虚拟机）
HDFS	冷备数据存储模块	长期存储实例备份文件，binlog，为故障恢复提供数据服务	选配 1 台	4-16-6000（虚拟机）

正式环境配置（中等档次）：

组件名称（英文）	组件名称（中文）	功能介绍	数量（台）	建议配置（CPU/内存/磁盘）
LVS	虚拟负载网络	LVS（DR 模式），提供负载访问能力，为每一个数据库实例提供唯一虚拟 IP，可替换为 TGW	选配 2 台	8-16-500
chitu	WBE 化运维平台	赤兔管理平台，提供 WEB 化管理整个集群的能力	无需	4-8-200
zookeeper/monitor	决策集群	提供高可用和一致性能力	3	8-16-500（虚拟机）
scheduler/manager	调度集群	提供主备切换/扩缩容/资源管理等能力	无需	8-16-500
DB	数据库模块	包括 Proxy，agent，MySQL，存储和计算核心节点	3*N	32-128-2000
HDFS	冷备数据存储模块	长期存储实例备份文件，binlog，为故障恢复提供数据服务	选配 3 台	8-64-6000

正式环境配置（中高档次）：

组件名称（英文）	组件名称（中文）	功能介绍	数量（台）	建议配置（CPU/内存/磁盘）
LVS	虚拟负载网络	LVS（DR 模式），提供负载访问能力，为每一个数据库实例提供唯一虚拟 IP，可替换为 TGW	选配 3 台	16-32-500
chitu	WBE 化运维平台	赤兔管理平台，提供 WEB 化管理整个集群的能力	无需	16-32-500
zookeeper/monitor	决策集群	提供高可用和一致性能力	5	16-32-500（虚拟机）
scheduler/manager	调度集群	提供主备切换/扩缩容/资源管理等能力	无需	16-32-500

DB	数据库模块	包括 Proxy，agent，MySQL，存储和计算核心节点	3*N	64-256-6000
HDFS	冷备数据存储模块	长期存储实例备份文件 binlog，为故障恢复提供数据服务	选配 3 台	16-32-50000

5.3 单中心容灾部署建议

单中心容灾时，数据库集群主要需要预防如下故障：

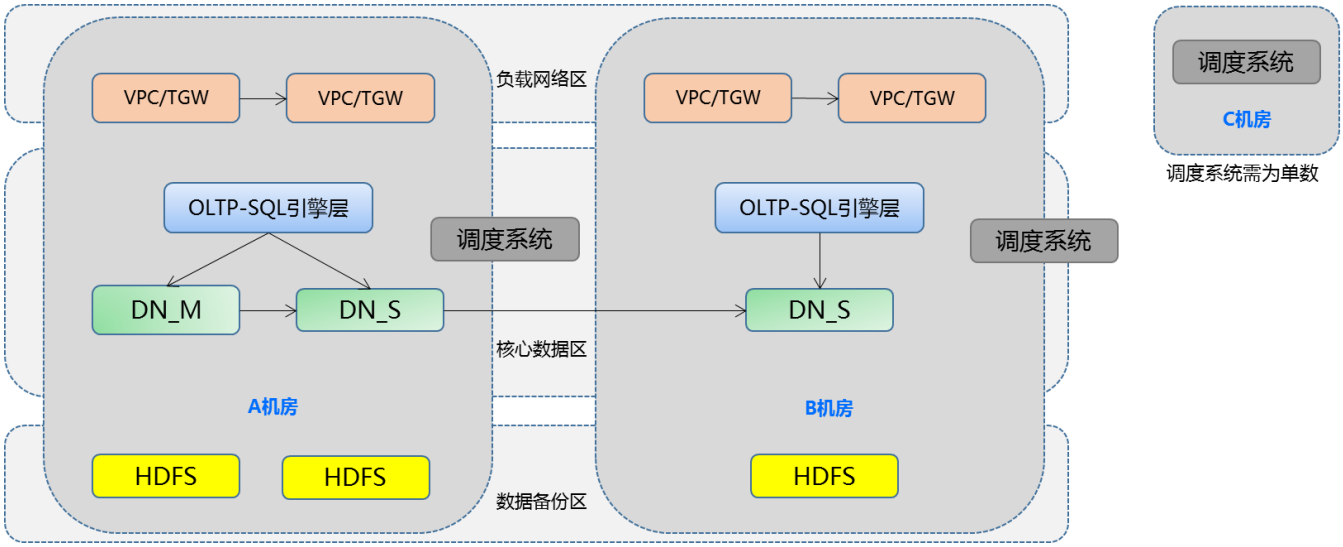
- 机房内交换机、负载转发或网卡等单点故障
- 机架电源、风扇、冷却等相关的单点故障
- 数据库服务器硬件的单点故障

因此单中心容灾部署建议至少采用以下要求：

- 交换机、负载转发等网络设备至少是双活容灾
- 数据库服务器、管理调度建议采用一主二从模式部署
- 同一模块不同的设备需跨机架部署
- 需部署数据备份模块

5.4 多中心容灾部署方案

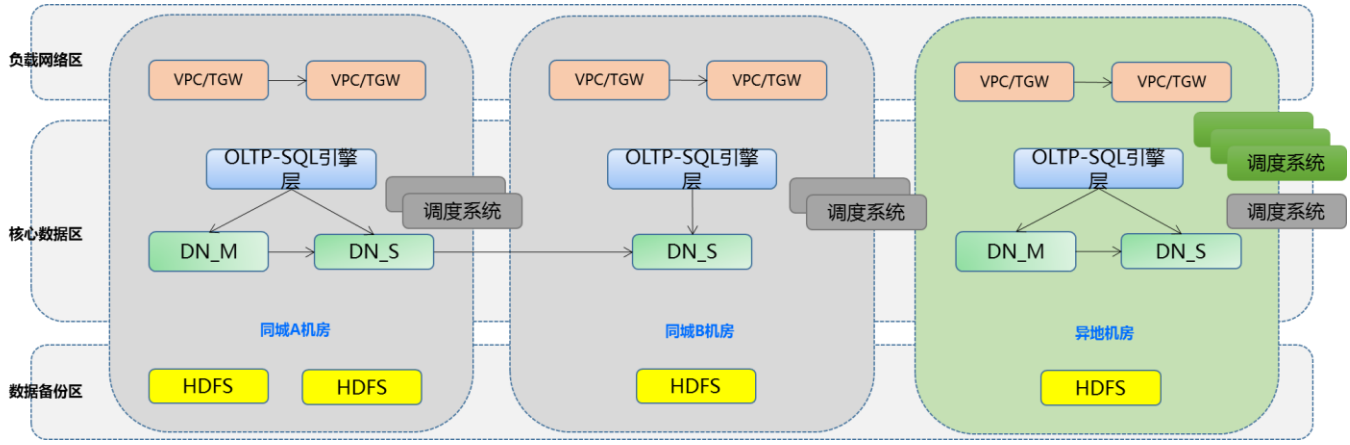
5.4.1 同城双中心部署建议



同城两中心时，建议至少采用三个节点的模式（也可以扩展为每个中心两节点（2+2 模式）），分布在两个机房，机房之间采用专线互通，每组设备主机房部署 2 台，从机房部署 1 台。负载网络采用类似于 LVS 的(DR 模式)软负载均衡方案，建议能支持机房级故障网络也可以切换的。由于 SQL Engine 会自动分配请求，因此业务系统无论从哪个机房的访问，都可以访问到正常的 DB。任何一个数据库节点或者机房当掉，TDSQL 一般在 40 秒（含 30 秒检测时间）左右完成自动地切换，事务做到 0 丢失，业务确保重连机制即可。

说明：腾讯云金融云（公有云金融专区），默认支持通常双中心架构。

5.4.2 两地三中心部署建议



两地三中心即在同城双中心的基础上 ,增加一个灾备中心。两个灾备实例之间 ,通过 DCN 方式进行同步，确保数据一致。

5.5 腾讯专有云平台（TCE）

腾讯云专有云(Tencent Cloud Enterprise, TCE)是基于腾讯公有云成熟产品体系推出的企业级云平台，支持私有化输出，具有高可用、统一管控、行业合规的特点，提供数字化转型所需的云化技术、大数据、AI 等全面能力，为行业领导者、大型集团企业提供专业一体化的私有云解决方案。

腾讯专有云平台集成公有云众多热点产品作为自身的组件，给客户提供插拔式的按需定制的产品体验，整个专有云提供包括服务器、网络、存储等 IaaS 基础组件，同时提供包括云数据库、大数据处理，容器、微服务 PaaS 相关的组件，这些都可以根据客户的需要进行增减。腾讯专有云将提供给客户极致的互联网式的应用体验，帮助客户提供应用代码以下的所有互联网公司使用过的技术服务。

目前 TDSQL 已和 TCE 进行深度集成，您可以在选购 TCE 的同时，选择 TDSQL 数据库选件。

5.6 腾讯企业云平台（TStack）

腾讯云 TStack（腾讯企业云）诞生于腾讯内部私有云使用场景，是总结公司自身多年实际运营运维经验，结合技术能力，集 IaaS、PaaS 和 SaaS 为一体的综合云服务解决方案。基于开源 OpenStack 进行二次开发，在开源平台上进行大量优化和自主创新，例如对 OpenStack 单 Region 规模的调优，对多平台兼容处理和内网级的混合云管理等。同时腾讯云 TStack 具有大量私有化部署经验，具备稳定性、统一管理、可视化运营等特点，是可进行对外输出 IT\信息化能力的一个重要途径，助力政府、企业以及各类对私有环境有要求的机构组织，构建稳定安全的云环境和健康的云生态。

目前 TDSQL 已和 TStack 进行深度集成，您可以在选购 TStack 的同时，选择 TDSQL 数据库选件。

6 产品优势

6.1 数据不丢不错乱

业界领先的强同步复制能力（Multi-thread Asynchronous Replication，MAR），在主从架构下，确保主从节点数据完全一致，即使在节点故障切换、故障恢复时，确保主从节点数据完全一致，不丢不错乱。

6.2 更可靠的数据库

完善的故障故障转移与恢复功能，灵活的运营，完善的监控，确保数据库资源的持续可用。 同城多中心部署场景下，即使是数据中心级故障，也能有效保障业务连续性。

6.3 基于云的数据库

基于分布式架构设计，无论是数据库集群，或是实例，都具有良好的扩展能力，可以根据实

际需求采购数据库资源。闲时超用策略，即使是偶然性业务波动，也能轻松应对。多种读写分离方案，RocksDB，热点更新等能力，能让数据库实例具有更强的适应性。

6.4 更安全的数据库

产品已代表腾讯云云数据库通过多项国家或国际认证，符合信息安全等级保护三级（部分四级）安全标准，目前稳定支撑金融行业多个核心业务系统。

6.5 更好用的数据库

产品即开即用，提供包括自动备份、系统监控、性能分析等多种完备能力。也是云上首批提供（扁鹊智能分析系统）能力的数据库，为运维人员更好优化系统性能提供有力支持。

7 产品资质

TDSQL 现已代表腾讯云数据库活动通过多项国家或国际认证，包括但不限于：

- MariaDB 白金会员
- ACMUG 和中国开源数据库专业委员会的主席团成员
- ISO27001
- ISO27001：2013
- ISO20000
- ISO20000-1：2011
- ISO22301
- ISO9001
- ISO27018

- PCI DSS 1 级服务提供商
- SOC 审计
- ITSS 云服务增强级认证
- 公有云三级备案和测评
- 金融云四级备案和测评
- 可信云云数据库服务认证
- 可信云云用户数据安全保护能力测评
- 可信云金牌运维专项评估
- ITSS 认证
- 金牌等级通过 CSA STAR 认证，同时获得 CNAS 和 UKAS 国内外双认可信息安全管理体系认证

8 常见应用场景

8.1 成为去 O 的中坚力量

企业的核心业务系统一般都是 OLTP 为主的应用场景，在这个领域，Oracle 一直是市场的领导者，而开源数据库 MySQL、MariaDB、PostgreSQL 等似乎仅能提供给中小企业或个人站长使用。在互联网领域，以 TDSQL 为代表的分布式数据库应用非常广泛，用普通 x86 服务器，便可轻松支撑起上亿的用户访问。经过验证的分布式数据库在性能和稳定性上甚至高于用高端设备搭建的 Oracle RAC。 当然，对于企业而言，由于 Oracle 数据库和上层应用绑定比较紧密，通常会使用到 Oracle 的存储过程、自定义函数、触发器，这就涉及到应用迁移。这个工作的工作量和时间周期通常较大，综合计算下来，即使加上软件改造成本，

采用 TDSQL 的 TCO 仍然低于使用商业数据库。当前，不管是互联网和传统行业，去 O 的成功案例比比皆是。

8.2 分支业务聚合到总部(全国/全球覆盖)

政务、银行、大型国企的组织架构通常采用总部-分部-分支的架构。因为各种原因，其某些核心 IT 系统建设也采用总部-分部-分支模式。随着业务互通，人员互通，信息互通等需求越来越强烈，业务逐渐向总部聚合。 而业务聚合这个重要问题是数据库数据同步，以及性能问题。 而产品在数据库同步上的独特优势，可以提供一对多、多对一的同步方案，同步可以基于正则表达式筛选精确到库表。云上承载的各类保险、银行客户用其自身业务稳定运行，证明了利用产品可以搭建一个承载全国业务的大型系统的可行性。

目前基于产品的米大师，阅文集团，富途等业务，已经提供了覆盖全球的业务。

8.3 混合云业务

产品可支持专有云部署方案，支持在云数据库上挂载自建只读实例，支持第三方同步工具或数据订阅解决方案，可以让用户轻松搭建一套跨云的混合云架构。业务系统和数据通过专线（或 VPN）进行安全同步，构建易扩展的混合云架构。

8.4 实时高并发交易场景

互联网金融、电商、O2O、零售等行业，普遍存在用户基数大、核心交易系统数据库单表上亿行且访问日益变慢等问题，制约业务发展。产品提供弹性扩展能力，能够极大提升数据库处理能力，可以面对类似全网推广、限时秒杀等营销活动；配合热点更新，强同步复制能力，即使是敏感交易类业务也可以完全用产品承载。

8.5 海量数据存储访问场景

面向物联网，交易订单等业务，业务数据增长迅猛，会产生超过单机数据库存储能力极限的数据，数据库实例超过 TB 级别且持续快速增长，造成数据库容量瓶颈，限制业务发展。产品 可以弹性扩展存储空间，提供 RocksDB 等高压缩比的存储能力，且可以高效利用每一个物理节点的存储量能力，避免浪费。可以广泛应用于 IOT 场景下的车联网、工业远程监控、智能家居、智能汽车、充电桩加油站等超大规模传感器数据上报存储访问场景。

8.6 游戏应用场景

游戏等需要弹性扩容和快速回档的业务。产品对计算资源的弹性伸缩能力，赋予您更高的生产力，分钟级部署游戏分区数据库。借助任意时间点回档到临时实例功能及支持批量操作的特性，您可以随时随地在不影响现网运营情况下，恢复到任意时间点，为游戏回档提供有效支持。利用分布式实例，可以轻松搭建起可弹性扩展的”全区全服”内业务。

9 案例简介

9.1 米大师

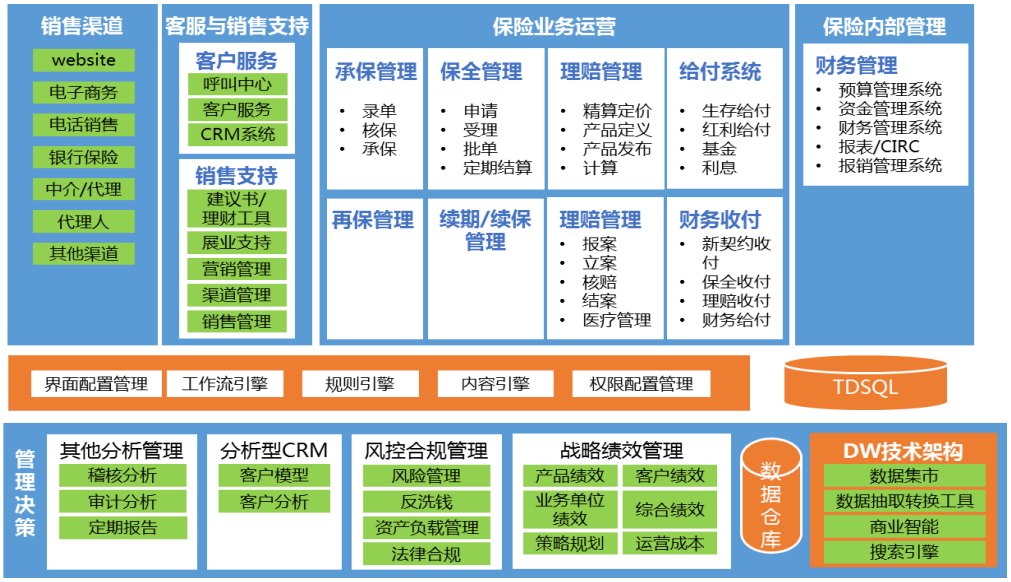
腾讯推出的移动支付组件米大师，专注移动支付解决方案，实现移动终端的更大营收。目前已全面支持微信、手机 QQ、手机 Qzone 等平台手游。当前米大师正在为多个腾讯游戏、电商支付等平台提供服务。米大师分布式物理节点数超过 1000 个，账户总量超过 100 亿，每日请求超 10 亿，平均 99.95%的请求在 5ms 以内响应，连续三年运营零中断、数据零误差。

9.2 微众银行

微众银行作为国内首批互联网银行，也是第一个核心系统完全采用 TDSQL(非 Oracle) 的公司。2016 年下半年，微众银行已经有超过 500 台 x86 服务器运行着 TDSQL 数据库服服务器，稳定支撑日交易峰值千万+次的业务。

9.3 全品类保险业务

腾讯云数据库 TDSQL 承载了全品类的保险业务，如财产险（安心财险）、人寿险（和泰寿险）、信用保证保险（阳光渝融）、互助保险（众惠相互）等；不仅协助其快速开展业务，适应当前互联网保险业务快速变化需求，并通过腾讯金融云、独享集群（金融围拢）和专有云等方案，搭建其符合银保监会信息安全要求的业务系统。



9.4 三一重工（树根互联工业物联网）

树根互联技术有限公司由三一物联网及工业大数据技术及运营核心团队创建，是独立开放的互联网高科技企业，致力于打造最具客户价值的工业物联网平台。托深厚的工业积淀，汇聚了大批工业大数据科学家团队，打造了开放的工业物联网生态系统。

工业领域采集的数据量则是消费级的成百上千倍，以三一重工的挖掘机为例，需要采集位置、油温、油位、压力、温度、工作时长等超过 5000 多个参数。如此大量数据，传统数据库根本无法承载，因此三一重工选择了 TDSQL 产品。

9.5 威富通（微信支付渠道商）

威富通作为微信支付主要渠道商之一，曾采用 Oralce 作为其订单系统，但互联网动辄千

万级用户量是小型机也无法承载的，最后通过评估决定将 Oracle 切换为开源数据库，并最终选择腾讯 TDSQL 方案，因为其威富通的金融级特性必须要确保超高性能和强大的数据一致性。

9.6 黑桃（约战）

《约战:精灵再临》是“黑桃互动”研发的，富含 Galgame 的文字恋爱冒险元素以及各类模拟养成和社交休闲玩法。其游戏采用全新 ACT+GAL 动作玩法，用户数据存储采用 TDSQL（分布式数据库实例），以提供强大的扩展性。

10 附录

10.1 通用约定格式

表 1: 格式约定

格式	说明	样例
	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 危险：重置操作将丢失用户配置数据。
	该类警示信息可能导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告：重启操作将导致业务中断，恢复业务所需时间约10分钟。
	用于警示信息、补充说明等，是用户必须了解的内容。	 注意：导出的数据中包含敏感信息，请妥善保存。
	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明：您也可以通过按**Ctrl** + **A**选中全部文件。
>	多级菜单递进。	设置 > 网络 > 设置网络类型
粗体	表示按键、菜单、页面名称等UI元素。	单击 **确定**。

courier字体	命令。	执行 `cd /d C:/windows` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[]或者[a|b]	表示可选项，至多选择一个。	`ipconfig` [-all|-t]
{}或者{a|b}	表示必选项，至多选择一个。	`swich` {stand | slave}

10.2 功能术语表

- **赤兔运营平台：**又名运维平台，运营平台，运维端等，是指 TDSQL 面向运维角色或小型化部署的 WEB 化管理系统，其提供全面的功能。
- **云数据库管理平台：**又名租户平台，租户端，是指 TDSQL 面向租户（用户）部署的 WEB 化管理平台，通常是提供在类似于公有云业务形态时使用，其功能较全。
- **实例：** 用户实际使用的一个最小单位的数据库服务集合；一般来说，用户使用数据库可独立的享有数据库实例 IP、端口和账号；实例之间资源（权限、性能、数据空间）完全隔离；实例可能仅存于 1 台服务器上，也可能在不同的物理服务器上。一个数据库实例可以包含一个或多个用户创建的数据库等。
  - **实例 ID：**又名 SET_ID，是赤兔运营平台唯一标示实例的信息。
  - **UUID：**在云数据库管理平台又名实例 ID，是云数据库管理平台唯一标示实例的信息。需注意的是，一个唯一的云数据库管理平台的 UUID 对应一个唯一的赤兔运营平台的 UUID。

- **物理节点组（SET）：**搭建数据集群的若干物理节点，通常这些节点基于数据库主从协议联结成若干组，这些物理节点组的统称叫做 SET。需要注意时，如果没有采用多租户（虚拟化技术），通常一个物理节点组就等于一个 SET，如果采用多租户（虚拟化）技术，物理节点组（SET）中可能有多个虚拟节点组，从而可能是对应多个 SET。
  - 节点（DataNode）：一般代表一台物理服务器，在多租户虚拟化技术下，一个节点通常代表一个的 mysqld 进程。
- **数据库引擎、版本：** 每个数据库实例运行一个数据库引擎，数据库引擎是用于存储、处理和保护数据的核心服务，通常我们说的 MySQL、SQLServer、Oracle 就是引擎的叫法；每个数据库引擎又包括不同的软件版本，不同的数据库引擎版本都有自己支持的功能和特性。但请注意，innodb 等在本文中被叫做存储引擎，与数据库引擎并非相同概念。

  说明：TDSQL 当前支持的 MySQL 协议默认选择 MySQL 或 MariaDB 分支。
- **集群：**一个独立网络区域的 TDSQL 服务器集合集群；一般来说，先有集群才能在集群上分配实例。而集群中的设备互为冗余，集群通常包括主数据库服务器、多个备份数据库服务器、网络设备、数据备份集群等。通常多个集群可以共享一套管理系统。
- **OSS：**此处代表赤兔运营平台的核心管理模块。
- **实例规格：**又名，实例配置是定义数据库的使用大小、性能的一种综合指标。规格使用计算能力、内存、存储容量等多种指标进行定义。由于云计算的弹性能力，实例规格实际上是可以按需伸缩的，用户可以选择在合适时机选择对规格进行扩容或缩小，以保障数据库引擎有足够的空间来写入内容和日志。
- **DDL：**数据库模式定义语言（Data Definition Language），主要的命令有 CREATE、ALTER、

DROP 等。

- **DML**：数据库操纵语言(Data Manipulation Language)，命令是 SELECT、UPDATE、INSERT、DELETE。
- **OLTP**：联机事务处理(On-Line Transaction Processing)，是传统的关系型数据库的主要应用，主要是基本的、日常的事务处理，例如银行交易。
- **OLAP**：联机分析处理（On-Line Analytical Processing），是数据仓库系统的主要应用，支持复杂的分析操作，侧重决策支持，并且提供直观易懂的查询结果。
- **主机**：MySQL 数据节点中，直接承担读写的节点，简称主机（Master）；
- **从机**：以高可用高可靠为目的，让多个 MySQL 数据节点协同工作，并通过主从复制协议（REPLICATION）将主机数据复制一个协同节点中，简称从机（Slave）；从机通常只可读取数据，不可写入数据。
- 说明：部分文献中也称为“主从、从机”为“主备、备机”；一般此处 “备”一词与备份、冷备的“备”并非相同方案，容易混淆。
- **数据备份**：以容灾的基础，为防止灾难或故障导致数据丢失，而将全部或部分数据集合从应用主机的硬盘或阵列复制到其它的存储介质的过程；通常也叫做备份服务、数据冷备。
- **HDFS**：Hadoop 分布式文件系统，是一种被设计成适合运行在通用硬件(commodity hardware)上的分布式文件系统。
- **LVS**：一种常用的开源虚拟网络服务技术，在此处的作用类似于商用负载均衡服务 F5。
- **调度集群 (Scheduler)**：帮助 DBA 或者数据库用户自动调度和运行各种类型的作业，比如数据库备份、收集监控、生成各种报表或者执行业务流程等等，TDSQL 把 Schedule、zookeeper、

OSS（运营支撑系统）结合起来通过时间窗口激活指定的资源计划，完成数据库在资源管理和作业调度上的各种复杂需求。

- **决策集群（ZooKeeper）：**它是 TDSQL 提供配置维护、选举决策、路由同步等，并能支撑数据库节点组（分片）的创建、删除、替换等工作，并统一下发和调度所有 DDL（数据库模式定义语言）操作，通常决策集群需要采用奇数台，实际部署的时候应大于等于 3 组并跨机房部署。
- **赤兔自动化运维平台（CHITU）：**基于 TDSQL 定制开发的一套综合的业务运营和管理平台，同时也是真正融合了数据库管理特点，将网络管理、系统管理、监控服务有机整合在一起。
- **接入 SQL Engine 集群（OLTP-SQL Engine）：**在网络层连接管理 SQL 解析、分配路由。(请注意，OLTP-SQL Engine 并非腾讯云网关 TGW 集群)。

  1. OLTP-SQL Engine 通常与 MySQL 混合部署，也可以部署在不同物理设备中；
  2. 从配置集群（ZooKeeper）拉取数据库节点（分片）状态，提供分片路由，实现透明读写；
  3. 记录并监控 SQL 执行信息，分析 SQL 执行效率，记录并监控用户接入信息，进行安全性鉴权，阻断风险操作；
  4. OLTP-SQL Engine 通常可以直接访问，但仍然建议前端部署需部署可提供负载均衡能力网关，并由负载网关对用户提供唯一虚拟 IP 服务。

- **非分布式实例：**又名关系型实例、CDB、RDS、noshard 实例、mysql 实例，是指完全兼容开源（如 MySQL）语法的数据库实例。在 TDSQL 中，仅有 1 个 SET 组成实例就是非分布式实例，其实例 ID 通常以 set 开头。
- **分布式实例：**又名分布式关系型实例、DCDB、DRDS、TDSQL、gs 实例、groupshard 实例、shard 实例等，是指基于分布式架构组成的实例，通常 90%兼容 MySQL。在 TDSQL 中，有多

个 SET 组成的实例就是非分布式实例，其实例 ID 通常以 Group 开头。

- **异步复制：**应用发起更新（含增加、删除、修改操作）请求，Master 完成相应操作后立即响应应用，Master 向 Slave 异步复制数据。因此异步复制方式下， Slave 不可用不影响主库上的操作，而 Master 不可用有概率会引起数据不一致。

- **强同步复制：**应用发起更新请求，Master 完成操作后向 Slave 复制数据，Slave 接收到数据后向 Master 返回成功信息，Master 接到 Slave 的反馈后再应答给应用。Master 向 Slave 复制数据是同步进行的，因此 Slave 不可用会影响 Master 上的操作，而 Master 不可用不会引起数据不一致。

- 注意:使用“强同步（不可蜕化）”复制时，如果主库与备库自建网络中断或备库出现问题，主库也会被锁住（hang），而此时如果只有一个主库或一个备库，那么是无法做高可用方案的。—— 因为单一服务器服务，如果故障则直接导致部分数据完全丢失，不符合金融级数据安全要求。

- **半同步复制（或强同步可蜕化）:** 半同步复制是 google 提出的一种同步方案，他的原理是正常情况下数据复制方式采用强同步复制方式，当 Master 向 Slave 复制数据出现异常的时候（Slave 不可用或者双节点间的网络异常）退化成异步复制。当异常恢复后，异步复制会恢复成强同步复制。半同步复制意味着 Master 不可用有概率会较小概率引起数据不一致。

- **垂直切分(通常也叫做“分库”)**也就是按功能切分数据库，这种切分方法跟业务紧密相关，实施思路也比较直接，比如“京东 JD”等电商平台，一个原有一个数据库实例，按功能切分为会员数据库、商品数据库、交易数据库、物流数据库等多个数据库实例，共同承担业务压力。

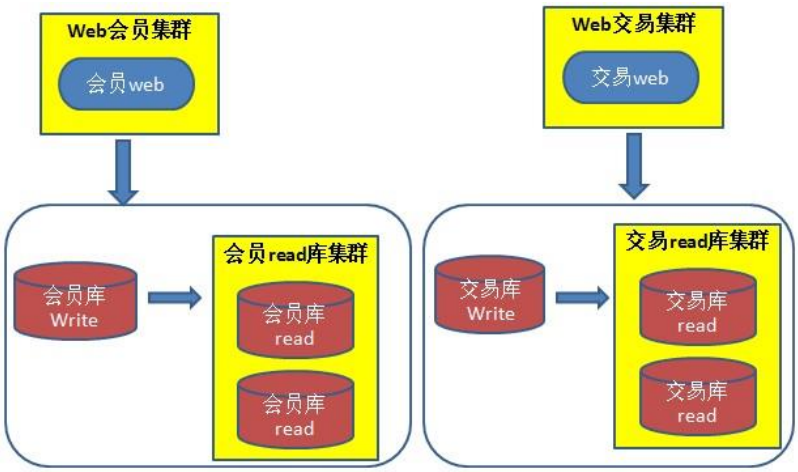
- **水平切分（又叫做“分表”、横向拆分、水平拆分等）:** 垂直拆分并不能彻底解决压力问题，因为单台数据库服务器的负载和容量也是有限的，随着业务发展势必也会成为瓶颈，解决这些问题的

常见方案就是水平切分了。水平切分是按照某种规则，将一个表的数据分散到多个物理独立的数据库服务器中，这些“独立”的数据库“分片”；多个分片组成一个逻辑完整的数据库实例。一般来说，分表的前提是分库。

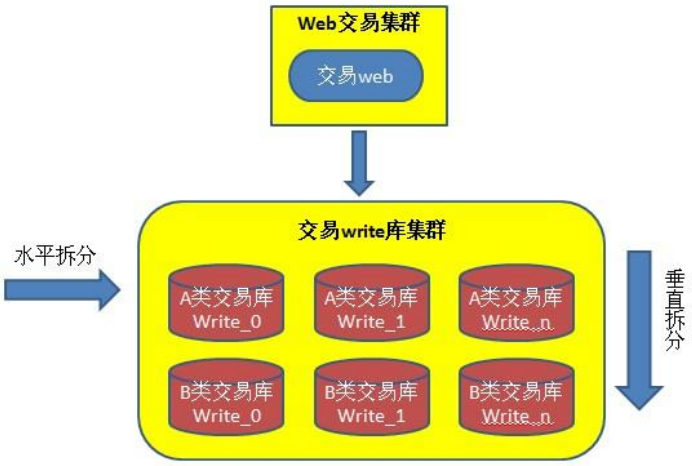
- **拆分建：**关系型数据库是一个二维模型，数据的切分通常就需要找到数据库表中某一字段作为拆分键（shardkey）以确定拆分维度，
- **资源独享：**又叫独享集群，是在一个大集群中专门分配几台物理设备给某一客户独占使用的方案。

10.3 分布式数据库的分库与分表

分库通常也叫做“垂直切分”，即按功能切分数据库。这种切分方法跟业务紧密相关，实施思路也比较直接，比如“京东 JD”等电商平台，在业务中期架构中，将一个原有数据库实例，按功能切分为会员数据库、商品数据库、交易数据库、物流数据库等多个数据库实例，共同承担业务压力（如下图）。有时候，垂直拆分并不能彻底解决压力问题，因为单台数据库服务器的负载和容量也是有限的，随着业务发展势必也会成为瓶颈，所以，分表成为解决这些问题的常见方案。



分表又叫做“水平切分”，是按照某种规则，将一个表的数据分散到多个物理独立的数据库服务器中，这些“独立”的数据库“分片”，多个分片组成一个逻辑完整的数据库实例。一般来说，分表的前提是分库。



水平拆分的方案，实际上是分布式架构最直接体现，它与 RAC 等方案的最大不同点在于每个计算节点都参与计算和数据存储，每个计算节点都仅存储一部分数据。因此，分布式数据库从架构上来讲，性能是可以线性增长的。