

# 游戏数据库 TcaplusDB

## 常见问题



腾讯云

## 【 版权声明 】

©2013–2024 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

## 【 商标声明 】



及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

## 【 服务声明 】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。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默示的承诺或保证。

## 【 联系我们 】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或 95716。

# 文档目录

## 常见问题

信息咨询类

数据库使用类

数据库原理类

# 常见问题

## 信息咨询类

最近更新时间：2024-03-06 10:09:31

- [TcaplusDB 支持数据淘汰吗？](#)
- [TcaplusDB 的数据结构有哪些？](#)
- [TcaplusDB SDK 的单实例内存和 CPU 消耗？](#)
- [TcaplusDB 单个表格组的表格个数是多少？](#)
- [TcaplusDB 对 key 和 value 字段的限制是什么？](#)
- [TcaplusDB 备份文件存放的时间？](#)
- [gameserver 与 tcaproxy（接入层）是全链接吗？](#)
- [TcaplusDB 的数据怎么做数据分析？](#)
- [TcaplusDB 单张表的最大大小是多少？记录条数限制是多少？](#)
- [TcaplusDB 支持多表事务操作吗、支持批量写操作吗？](#)
- [TcaplusDB API 升级是前向兼容的吗？](#)

### TcaplusDB 支持数据淘汰吗？

TcaplusDB 支持表级别数据淘汰，淘汰的依据是记录最后的写时间。

### TcaplusDB 的数据结构有哪些？

TcaplusDB 支持的数据结构有 list 数组、按照部分 key 查询（索引）、key-value、key-object（即单个 key 的 value 值可以是任意的数据结构，例如 gameserver 可以将 lua table 序列化到 value 字段里）。

### TcaplusDB API 的单实例内存和 CPU 消耗？

API 单实例内存消耗最大73MB、CPU 消耗最大30%。

### TcaplusDB 单个表格组的表格个数是多少？

TcaplusDB 单个表格组中表的个数最大是256，如果您单个表格组下的表的数量大于256，可以采用再新增新的表格组或者将表进行合并操作，如果需要技术支持，请 [提交工单](#) 选择“其他腾讯云产品”申请。

### TcaplusDB 对 key 和 value 字段的限制是什么？

generic 表 key 字段的个数是4，list 表 key 字段的个数是3，单个 key 字段的长度是1024B，generic 表的 value 字段的个数是128，list 表的 value 字段个数是127，单个 value 字段大小是256KB，整体记录最大是1MB。

### TcaplusDB 备份文件存放的时间？

TcaplusDB 备份的引擎文件保存7天，Ulog 流水保存7天，各个 TcaplusDB 环境是不一样的，您可根据具体的 TcaplusDB 环境向 [客服咨询](#)。

## gameserver 与 tcaproxy（接入层）是全链接吗？

为了节省 gameserver、tcaproxy（接入层）维护 TCP 链接的成本，gameserver 支持选择部分 tcaproxy（接入层）建立链接。

## TcaplusDB 的数据怎么做数据分析？

TcaplusDB 的数据可以导出任何格式，包括 json、pb 等方式，可以入库 tdw 等数据分析系统；TcaplusDB 支持将实时的数据导入到 MySQL 数据库等。

## TcaplusDB 单张表的最大大小是多少？记录条数限制是多少？

单个表能细分到1w个数据分片，每个数据分片是256GB，即单个表的总大小是1w \* 256GB，单个表没有记录条数的限制，单个表的记录条数和单条记录大小是相关的。

## TcaplusDB 支持多表事务操作吗、支持批量写操作吗？

TcaplusDB 目前不支持多事务操作，即同时提交多个操作，如果后面操作失败则回滚所有已做操作。对于这样的情况，通常需要从业务侧解决。可以采用“先扣再加”的策略，或记录重要操作的对账日志等，也暂时不支持批量写操作。

## TcaplusDB API 升级是前向兼容的吗？

TcaplusDB API 升级是前向兼容的，已有的接口、命令字、功能不会修改。

## 什么是 TcaplusDB？

游戏数据库（Tcaplus Database，TcaplusDB）是腾讯专为游戏设计的分布式 NoSQL 数据库，针对游戏业务爆发性的增长和海量游戏数据管理提供高性能、业务无感知弹性伸缩、毫秒级读写延时与安全的数据存储服务，广泛应用于腾讯数百款游戏中，包括《和平精英》、《王者荣耀》、《穿越火线》等数百款精品游戏中，每天支持超过4000万用户同时在线的数据请求。

## TcaplusDB 适合哪些场景？

手游、端游、页游，支持分区分服与全区全服部署方式，同时也支持非游戏领域中高并发、低延时访问要求的使用场景。

## 相比 Redis+MySQL 作为游戏数据库，TcaplusDB 有哪些优势？

TcaplusDB 采用内存加高效 SSD 云硬盘混合部署的方式，能同时满足数据快速读写与持久化存储的需求；对于关系型数据库用户，TcaplusDB 提供类 SQL 的命令行工具 Tcaplus Client，用户可无缝切换执行 CRUD 操作；

用户不需要担心表文件过大导致查询加载缓慢的问题，TcaplusDB 支持单表大小高达2.56PB，且不会降低查询效率；

Redis+MySQL 扩容需要手工干预且无法满足业务突增场景且使用成本高昂，TcaplusDB 用户只需要关注业务开发，数据库的扩容和性能优化全部后台由专业的 DBA 完成。

## TcaplusDB 如何保证数据安全？

**数据容灾：**集群主从节点主要采用同城跨 IDC 部署(或跨机架、跨楼层方式，主从节点同步延时最多10ms)，每15分钟 binlog 备份，每天全量冷备；整个集群各个模块都是无损升级与扩缩容，用户业务无感知；

**数据一致性：**主从数据实时同步并采用 req/ack 机制确保数据完全一致；主从切换时会确认主从数据一致后再进行切换；故障发生时基于冷备+binlog流水重建方式进行数据恢复，保障数据完整性；主从数据会进行周期性一致性全量核对，并有异常修复机制；用户请求层面，根据实时反馈的响应码判断数据操作是否正常；

**数据内容安全：**数据落地 CRC 校验防篡改；数据文件存储采用谷歌 Snappy 压缩，且数据读写压缩或解压缩后会采用序列化和反序列化操作，即使数据文件被劫持也无法解析数据内容；

**访问安全：**集成腾讯云用户权限管理体系 CAM；TcaplusDB 后台支持基于 IP 白名单访问，即指定 IP 客户端读写 TcaplusDB 数据；表格组内的数据交互均需密码验证，所有用户操作均有审计日志；

**外部合规：**数据文件会采用 aes-128-cbc 加密，满足欧盟 GDPR 标准。

## TcaplusDB API 各版本字段大小限制

小于 Tcaplus 3.32.0 API，单个 value 字段128KB，单条记录256KB。

大于等于 Tcaplus 3.32.0 API 但是小于 Tcaplus 3.40.0 API，单个 value 字段256KB，单条记录1MB。

大于等于 Tcaplus 3.40.0 API，单个 value 字段，单条记录10MB。

## 目前 TcaplusDB 的发送缓冲区和接收缓冲区大小分别是多大？

目前 gameserver 与单个 tcaproxy 的发送和接收缓冲区是10MB。

## 批量接口的错误码怎么使用？

批量接口的错误码分为两种：总错误码和子操作错误码。当至少一个子操作成功时，总错误码为成功；当所有子操作都失败时，总错误码为失败。总错误码可以通过 response 的返回值获取，子操作错误码通过获取单个 record 的返回值得到。

## tcaplus 表的一级字段数量上限是256？

目前一个表最多支持256个一级字段

## 请问对于一个游戏来说，tcaplus 最多支持多少连接数

默认一个 proxy 最多支持3000个连接，但是可以通过修改配置调大 proxy 的连接数，同时，一个 api 实例目前最多连接200个 proxy；

## Tcaplus 访问数据库是并行的吗？

支持并行，API 端每个连接（会话）的请求是串行的，不同连接是并行的。在 API 端，连接不是线程安全的，不用多线程使用同一个连接。

## 一个 shard 可以存储多大数据量？

单个 shard 最大可以存256GB的数据，一般推荐数据量维持在10GB以内。注：shard 是有初始空间的，但不是固定的，和具体设置的表引擎参数有关的，例如 hash 桶设置数量等。一个表最多支持10000个 shard，每个 shard 256G，Tcaplus DB 是分布式数据库，没有行数限制。

## TcaplusDB 的 pb 版本对于不同版本的兼容性如何？

pb 是前后兼容的

## 请问跟 int 相比，用 string 做 tcaplus 的 primary key 或者 sharding key，有什么需要注意的吗？

跟int相比的性能差距不大，总体原则: key 不要太长。

## 一张表的数据量很大，例如10亿条记录，导致一个物理机器的放不下，那么 tcaplus 这边会把这些数据分散到多个物理机器上吗？

在运营过程中可以加机器，把数据分散到新存储上，每组存储都是主从模式。

## 主从数据实时同步的时候，若主机器因为物理原因宕机了，理论上可能会有短时间的数据丢失吗？

理论上是存在短时间数据丢失的，是最终一致性的同步。因为是先写内存，之后系统刷盘。如果主机挂了，马上会主从切换。

## 如果 Tcaplus 远端的 meta 版本更高，而本地 svr 的 tcaplus meta 版本更低，中间有差异的这部分数据会存入 tcaplus\_db 吗（例如新增了一个字段）？还是会被直接舍弃掉？

以默认值写入

## 关于表定义中的分表因子，未定义时有默认分表因子吗？已有的表 splittablekey 可以再新增字段吗？

TDR 表：没有定义分表因子字段，系统默认将主键字段作为分表因子字段。

PB 表：没有定义分表因子字段，多个本地索引中的交集字段为分表因子字段，若没有定义本地索引交集字段则还是默认主键是分表因子。

表 splittablekey 一旦定义不能修改，不能增加和删除。

## 关于 list 表变更时，首次新增 customattr2中定义 listnum 的注意事项：

list 表的元素个数可以改大，但是首次新增 customattr2做表变更时先不能改大 customattr2中的 listnum，变更成功后再修改 customattr2中的listnum，否则会变更失败。

## **Tcaplus 是主备模式吗？写请求的时候，主机写成功就给业务回包，还是要等备机同步后才回包？**

是的，一主一从主备模式，主写成功回包。

# 数据库使用类

最近更新时间：2024-03-06 10:23:01

- [如何获取某个响应包错误码的含义？](#)
- [TcaplusDB 的乐观锁原理和使用方法？](#)
- [list 表的使用场景及注意事项？](#)
- [insert、update、replace 有什么区别？](#)
- [如何获取某个表的记录数目？](#)
- [TcaplusDB 支持遍历操作吗](#)
- [TcaplusDB 支持部分字段更新吗？能只是获取部分字段吗？](#)
- [TcaplusDB 对单个主键连续操作是保序的吗？](#)
- [TcaplusDB 支持表定义变更吗？](#)
- [怎么判断响应包分包是否已经结束？](#)
- [GetRecordCount 与 GetRecordMatchCount 的区别？](#)
- [TcaplusDB 有透传的字段吗？](#)
- [SetResultFlag 的作用？](#)
- [increase 操作可以一次对多个 普通字段进行 increase 操作吗？如果 主键字段不存在的场景呢？](#)
- [TcaplusDB 在读取记录时，有哪些省流量的方法？](#)
- [TcaplusDB 支持回档吗？支持的回档粒度如何？](#)
- [TcaplusDB 按照部分 key 查询（索引）的效率怎么样？](#)
- [TcaplusDB API 的超时机制是什么？it is timeout 错误什么意思？](#)
- [TcaplusDB API 的 SendRequest、OnUpdate、ReceiveResponse 函数具体的作用是什么？](#)
- [TcaplusDB 如何实现字段全局自增的功能？](#)
- [关于部分 key 查询（索引）定义有哪些规则？](#)
- [单个表 id 和 name 互查怎么实现？](#)
- [tcaplus\\_client 支持二级字段及以上字段显示吗？](#)
- [TcaplusDB 表变更时，需要注意什么？](#)
- [TcaplusDB 表定义时需要注意什么？](#)

## 如何获取某个响应包错误码的含义？

推荐您在代码里采用在 gameserver 里调用函数 `TcapusErrCode::TcapusErrCodeInit` 和 `TcapusErrCode::GetErrStr` 函数来获取错误码，也可以本地搜索 TcaplusDB API 的头文件。

## TcaplusDB 的乐观锁原理和使用方法？

举个抢火车票例子：

1. 100个人去抢同一张火车票，此时火车票的记录版本号是10，100个人 get 时，都是记录版本号为10。然后这100个人拿着记录版本号为10来抢火车票。
2. 100个人来对这张火车票进行写操作，因为写操作后记录版本号++，对于单个 key 的操作，tcapsvr 工作线程是排队的，第一个人抢到了火车票，此时火车票的记录版本号为11了，而剩余的99个人来抢时，拿着的记录版本号还是10。
3. tcapsvr 在处理这99个人的写请求时，全部报错了，因为 tcapsvr 端的记录的版本号和请求里的版本号不一致导致的，并发时的原则：
  - N个请求，只是需要第一个请求成功，剩余的N - 1个请求失败，则采用版本保护规则。
  - N个请求，需要N个请求全部执行完毕，则不需要版本保护规则，TcaplusDB 这边对同一个 key 的操作。

排队执行调用函数 `SetCheckDataVersionPolicy`，其值包括：

- `CHECKDATAVERSION_AUTOINCREASE`：检测记录版本号，只有当该版本号与服务器端的版本号相同时，该版本号才会自增。
- `NOCHECKDATAVERSION_OVERWRITE`：不检测记录版本号，强制把客户端的记录版本号写入到服务器中。
- `NOCHECKDATAVERSION_AUTOINCREASE`：不检测记录版本号，将服务器端的版本号自增。
- 默认类型为 `CHECKDATAVERSION_AUTOINCREASE`，推荐您使用。

## list 表的使用场景及注意事项？

凡是存在1: N 的使用场景， $N < 1024$ 时，优先考虑 list 表，例如存储玩家最近的100封邮件、最近的100场战斗记录等，list 表支持按照队头插入、队尾淘汰，队尾插入、队头淘汰，支持按照插入时间排序的 Top N 操作，单个 key 下的单元个数可以通过表变更修改大，因为需要兼容旧数据不能修改小。采用 `listgetall` 可以获得单个 key 下总的记录数目，推荐按照 `offset` 自己维护，`limit` 等于一定阈值的方式获取数据，`listreplace`、`listdelete`、`listdeletebatch` 时需要指定正确的 `index`。`listaddafter` 时需要指定单个 key 下的元素单元个数满时淘汰的规则，调用 `SetListShiftFlag` 函数设置所以 list 表有2个增加方向，2个获取方向，就是4种可能：

1. A、B、C、D、E，按照 `offset = 正数`，`limit = 2` 拉取就是A、B，C、D，E。
2. A、B、C、D、E，按照 `offset = 负数`，`limit = 2` 拉取就是D、E，B、C，A。
3. E、D、C、B、A，按照 `offset = 正数`，`limit = 2` 拉取就是E、D，C、B，A。
4. E、D、C、B、A，按照 `offset = 负数`，`limit = 2` 拉取就是B、A，D、C，E。

另外，调用 `GetRecordMatchCount` 可以获得记录的总数。

## insert、update、replace 有什么区别？

`insert` 操作，当 key 不存在时会进行插入，当 key 存在时，会返回错误码：

`TcapErrCode::SVR_ERR_FAIL_RECORD_EXIST`。

`replace` 操作，当 key 不存在会进行插入，当 key 存在时，如果采用了乐观锁，则根据乐观锁的结果执行不同的操作，成功则进行替换，失败则返回错误码：`TcapErrCode::SVR_ERR_FAIL_INVALID_VERSION`；如果没有采用乐观锁，直接替换。

update 操作，当 key 存在，如果采用了乐观锁，则根据乐观锁的结果执行不同的操作，成功则进行更新，失败则返回错误码：`TcapErrCode::SVR_ERR_FAIL_INVALID_VERSION`，如果没有采用乐观锁，直接更新；如果 key 不存在时，则返回错误码：`TcapErrCode::TXHDB_ERR_RECORD_NOT_EXIST`。

## 如何获取某个表的记录数目？

TcaplusDB API 接口里有 count 命令字，如果采用 tcaplus\_client 可以采用 count 表名命令获取表的记录数目。

## TcaplusDB 支持遍历操作吗？

TcaplusDB 支持遍历操作，包括 generic 表、list 表的遍历操作，遍历时注意设置从 tcapsvr slave 上遍历数据（从 tcapsvr slave 上遍历数据，不会影响 tcapsvr master 对外提供服务），即接口：

```
SetOnlyReadFromSlave(bool flag)。
```

## TcaplusDB 支持部分字段更新吗？能只是获取部分字段吗？

TcaplusDB 支持部分字段更新，推荐更新记录、获取记录时，显式的调用接口

```
SetFieldNames(IN const char* field_name[], IN const unsigned field_count)
```

 确定本次读写操作的字段，减少无效字段的带来的网络流量开销。

## TcaplusDB 对单个主键连续操作是保序的吗？

对于同一个 gameserver，同一个主键的操作保序，不同主键的操作不保序。对于不同的 gameserver，都不保序。

## TcaplusDB 支持表定义变更吗？

TcaplusDB 支持表定义变更，如果简单的增加普通字段、修改宏请采用表变更操作；其余的场景，需要动态修改表结构，即采用数据迁移 + 日志流水方式实现表定义变更，请 [提交工单](#) 选择“其他腾讯云产品”申请。

## 怎么判断响应包分包是否已经结束？

遍历，请按照 state 判断遍历是否结束，即接口 GetState，其余分包场景，请根据函数 `HaveMoreResPkgs` 判断分包是否结束。

## GetRecordCount 与 GetRecordMatchCount 的区别？

一次请求可能有N次响应回包，如果有多次回包，则 GetRecordCount 是指本次响应包里的记录条数，GetRecordMatchCount 指存在 tcapsvr（存储层）端的数据记录（单个 key 总的记录条数）。

## TcaplusDB 有透传的字段吗？

TcaplusDB 的 CS 协议分为 Head、Body 两部分，Head 里 UserBuff（最大1KB）、AsyncID、Sequence 都是透传的字段，请您根据实际需要使用。

## SetResultFlag 的作用？

写操作时，响应包里支持返回记录，读操作调用该函数无效，具体的 result\_flag 值含义如下：

0表示：只需返回操作执行成功与否，不需要返回 value 字段。

1表示：返回与请求字段一致的数据。

2表示：须返回变更记录的所有字段最新数据。

3表示：须返回变更记录的所有字段旧数据。

SetResultFlagForSuccess 接口可以设置成功场景下，返回的数据情况；SetResultFlagForFail 接口可以设置在失败场景下，返回的数据情况。

## increase 操作可以一次对多个普通字段进行 increase 操作吗？如果主键字段不存在的场景呢？

increase 操作可以一次对多个普通字段进行 increase 操作，需要 gameserver 传递的请求里对多个字段都进行了赋值，如果某个 key 不存在时，再进行 increase 操作，此时可以通过函数 SetAddableIncreaseFlag 设置；如果 key 不存在时，插入该 key 并进行 increase 操作，该 key 的非 increase 字段不会实际存储，在读取该条记录时非 increase 字段会采用默认值；如果 key 存在，则直接会执行 increase 操作。

## TcaplusDB 在读取记录时，有哪些省流量的方法？

TcaplusDB 读取记录时，支持在固定时间内记录没有变化则不返回 value 字段、记录版本号没有变化则不返回 value 字段，具体的见函数 SetFlags 函数。

## TcaplusDB 支持回档吗？支持的回档粒度如何？

TcaplusDB 支持回档，包括全区全服、单张表、千亿条记录回档N条记录，支持冷备时间回档（最近的凌晨01:05:00）、精确时间回档（秒级别）、模糊回档（您指定回档的规则），精确时间回档的速度请参照300GB数据、200GBUlog流水，耗时2小时左右，回档的原理，冷备时间回档即替换引擎文件，精确时间回档，将冷备的引擎文件 + 执行Ulog流水，执行到需要回档的时间点即可，基于 key 的回档需要您配合，先对这些 key 封号，TcaplusDB 回档完毕后，再对这些 key 解封。

## TcaplusDB 按照部分 key 查询（索引）的效率怎么样？

推荐在业务场景1：N（N > 1024）下使用部分 key（索引）查询，单个索引键下的主键个数 = 10GB/单条记录主键的大小，单次读写索引操作耗时100ms左右（单条索引键下有10w+以上的数据记录）。

## TcaplusDB API 的超时机制是什么？it is timeout 错误什么意思？

TcaplusDB API 对每个请求分配个 ID，发送成功后，将 ID 压入判断超时数据结构里，如果该请求的响应包回来后，则从数据结构里删除该 ID，如果在3秒内该请求的响应包还没有被应用层处理，则会显示打印错误日志，关键字包括 it is timeout，此时需要看下 gameserver 是否被阻塞，可能是 tcaproxy（接入层）在向 gameserver 端回包时丢包，可能是响应包已经到达了 gameserver 端，但是 gameserver 端没有及时处理 TcaplusDB 推荐您自己实现超时机制，这样可以对超时的请求进行重试等处理。

## TcaplusDB API 的 SendRequest、OnUpdate、ReceiveResponse 函数具体的作用是什么？

SendRequest 发送请求，本次的发送请求可能已经发送到网络上，也可能被压在 gameserver 与某个 tcaproxy（接入层）的发送通道里；ReceiveResponse 是从本地接收队列里获取响应包；OnUpdate 是负责将发送队列的请求发送到网络和从网络接收响应包到接收队列。在消息驱动编程模式下，OnUpdate 调用推荐的调用频率是1ms一次。

## TcaplusDB 如何实现字段全局自增的功能？

您需要定义单张表，设置单个 value 字段类型设置为 int64\_t（多个 value 字段可以实现计数器数组），多个 gameserver 可以并发对单个 value 进行 increase 操作（不需要设置版本保护规则），并且得到本次返回的 increase 结果，则该次 increase 的结果是全局自增的。

## 关于部分 key 查询（索引）定义有哪些规则？

索引键必须是主键的一部分，索引键必须包含分表因子，索引键不能是主键。

## 单个表 id 和 name 互查怎么实现？

采用第三个 key 字段 x 作为分表因子，在 id 和 x、name 和 x 上建立索引实现该功能，例如存储玩家信息时，可以加上玩家的省份地址，即主键是 uin、name、省份地址，索引键是 uin、省份地址，name、省份地址。

## tcaplus\_client 支持二级字段及以上字段显示吗？

是支持的，请执行 help select，查看 select \* into a.xml 的使用方法。

## TcaplusDB 表变更时，需要注意什么？

1. 只能增加新字段，不能修改原有字段类型、名称，更不能删除原有字段，如果需要修改这些，请采用动态修改表结构技术方案。
2. value 字段中的数组或字符串类型可以加长度，但不能减短。如果需要修改这些，请采用动态修改表结构技术方案。
3. 新增字段对应的 version 要在原有版本号上加1，相应地整个 xml 文件的版本号（文件最上面定义的）也要加1（与新加出来的字段版本号一致）。需要先对 TcaplusDB 的表做变更，再对 gameserver 端的表做变更。  
您如果需要动态修改表结构，TcaplusDB 是支持的，请 [提交工单](#) 选择“其他腾讯云产品”申请。

## TcaplusDB 表定义时需要注意什么？

1. count 字段的需要增加 refer 字段。
2. 分表因子需要高度离散。
3. 索引键不能和主键一样。

## TcaplusDB 表结构发生变更，后台如何做版本控制？

TcaplusDB 根据表的版本号(Version No.)解决多版本兼容性问题，即 TcaplusDB 存储层会存储同一个表不同版本号的数据；因此 gameserver 或客户端进行数据读写操作时，需要指定表的版本号。

当不同客户端对同一张表进行交替写，并且客户端各自使用不同版本的表定义的时候，pb 表和 tdr 表对于差异字段的处理方式是不同的。

假设 gamesvr1：表有 abc 字段

假设 gamesvr2：表有 ab 字段

假设 tcaplusDb：表有 abc 字段

对于 PB 表：

1、假设 abc 是 pb 表的字段，两个 svr 交替写，c 字段会清空。

对于 TDR 表：

1、假设 abc 是 tdr 表的一级字段，两个 svr 交替写，c 字段不会给清空。

2、假设 abc 是 tdr 表的二级字段，两个 svr 交替写，c 字段会清空。

## TcaplusDB 如何处理请求延时？

请求在3s内没有响应时，TcaplusDB 会打印包含关键字 timeout 的错误日志；建议客户端请求代码中也增加超时逻辑或重试次数。

## TcaplusDB 如何保证读写请求稳定性？

TcaplusDB 默认采用 Master-Slave 模式，Master 负责用户请求的读写，Slave 充当热备节点。当出现 QPS 突增，读写分流会自动开启，即 Master 承当写操作，Slave 承担读操作直至后台扩容完毕。

## TcaplusDB 是否支持保存策略(retention policy)？

支持，TcaplusDB 会默认保留所有数据，支持用户设定保存时间，数据过期会自动淘汰，当前工单支持。

## TcaplusDB 是否支持跨地域迁移？

支持，TcaplusDB 同时支持热迁移(采用跨区热备的方式)和冷迁移，支持数据导出为 Json 文件，也支持 MySQL 数据导入 TcaplusDB，当前均为工单支持。

## TcaplusDB 如何保证热数据？

TcaplusDB 采用 LRU 缓存算法保证数据热度。

## 对于游戏玩家数据，TcaplusDB 如何存储？

一条 TcaplusDB 记录(Record)即为单个玩家的数据，按照主键字段或索引进行数据操作。

## TcaplusDB 表创建后可否增加索引或修改主键？

不可以，如有修改需求请重建表。

## TcaplusDB 能否支持事务？

暂不支持，建议将相关的字段定义在同一个表里进行操作，获得类事务体验。

## 是否会保证一个线程里的非同步请求相关应答会按照顺序处理？是否需要用 syncid 来进行同步处理？（以功能为主，不使用同步 api 的情况下）

同一个线程里，同一个 key 的读写操作是按照顺序处理的，即保序。不同的 key 的读写操作顺序不保证，如果项目组采用 asyncid 来标志请求和响应的映射关系，是可以的，这种是透传字段，包括 asyncid、sequence、userbuff。

## delete 性能如何？是否需要 update 命令来替代 delete？

delete 就是一次普通的写操作，性能上没有问题，项目组可以随时进行 delete 操作。

## 线程与并发

TcaplusDB API 是单线程实现且不支持多线程，一个 tcaplus api 对象对应一个 request 和 response，由于 TcaplusDB API 是非线程安全的，所以不能多个线程使用同一个 TcaplusDB API 对象。如果需要实现并发场景，推荐单个线程使用单个 TcaplusDB API 对象。从线上运营来说，TcaplusDB API 实例在单个机器上保持 1-2 个已经足够了，一个 TcaplusDB API 实例的 qps 可达到 7-10 万。

## SDK 内存占用情况如何？

一个 TcaplusDB API 实例将占用 200M 左右物理内存，实例采用静态分配的方式，内存消耗主要在协议上，收发包各自一个结构体，故需要预分配一些内存。注：不同的 TcaplusDB API 版本，因为支持的单条记录大小不一样，预分配的内存可能有差别。

## Tcaplus 主键可以是结构体？还是只能是基础类型？

不建议使用结构体，理论上结构体是没问题的，不过用结构体后，不便于后期用各种工具或问题定位、回档、提供 key。

## 请求读取数据报错 525

一般是后端超时了，如果并发量大，可能是 gameserver 与 tcaproxy 之间通信问题，可以看下：

tcaproxy 是否正常

TcaplusDB API 版本号是否大于了接入层版本号

tcaproxy 的 TCP 连接数目是否超限

## 分布式索引进行 delete 有支持 where 的方法吗？

可以采用分布式索引，读出来，再单条单条的删除。

## 重新添加的表格组，对应表新增全局索引，回档恢复数据进去，索引都还在吗，还是会重做索引？

分布式索引是在数据插入时进行自动维护的，当我们采用数据导入方式进行时，数据会同步到全局索引去。

## 如果一个表有多个 primary\_key，通过指定其中一个 primary\_key 并删除对应的数据吗？

主键，按照delete删除。本地索引，按照 deletebypartkey 删除。

## telnet ip port 可以通 tcapdir 地址，但是连接数据库集群失败如何排查？

telnet 9999端口

需确认是腾讯云环境还是普通集群环境，腾讯云需要连客户自己的 vip+9999端口，普通集群连真实 dirip+9999端口。

因为一个集群有多个 dir，最好拿到 api 连接日志，看连哪个 dir 失败，或者连上 dir了但 proxy 版本比 api 版本高导致失败。

## Tcaplus 是否支持单个 app 下不同的 zone 能采用不同的协议

Tcaplus 支持单个 app 下不同的 zone 能采用不同的协议，即 zone1为 PB 协议，zone2为 TDR 协议。

## 表重建或删除再新建会影响缓存的配置吗？

重建可以保留，删除再新增，缓写就没有了，删表的时候就会删除缓写。

## 增加列需要重建表格吗？

增加列只用执行在线变更就行。

## 表结构里的 count 扩大是否需要重建表？

不需要，可以直接修改，并且不用加版本号，count 只能改大不能改小。

## 删除某一张表或者删除表中的某一行，需要变更 tables.xml 的版本号吗？

是已经存在的表是不可以删除字段，只能追加字段，如果需要删除字段需要整表删除然后重建。

## 业务侧扩大 blob 的长度，是否可以支持业务侧无感的动态调整？

可以的，count 定义了 refer，count 值可以修改大，且不需要增加版本号。

## 表变更修改了一级字段之后是否需要更新缓写呢？

如果这些字段不需要配置缓写，那么就不用动，如果需要配置则要更新下缓写配置。

## Redis 替代场景

### 1. TcaplusDB 是否能代替 redis 的 pub/sub 功能？

目前 TcaplusDB 没有 pub/sub 功能，可以采用 list 表来模拟实现，类似进程 A 去 list 表写数据，进程 B 从 list 表读数据这种，会增加代码改造成本。

### 2. 是否支持 redis 的 expired（自动到期功能）？

目前 TcaplusDB 支持数据淘汰，例如账号表设置了30天淘汰，那么会淘汰掉30天内没有任何写操作的数据，是物理删除，即该功能是支持的。

### 3. TcaplusDB Client 有 settl，想问下是否还支持 api?

单条记录设置 settl，是 TcaplusDB API 支持的，但是项目组的使用场景可以设置成全表数据淘汰，即单个表上亿条记录，淘汰标准是1个，例如按照30天内没有任何写操作来淘汰。

### 4. 是否可以使用 tcaplus 的 list table 来代替 redis 的 stored set 功能?

如果是 TopN，采用 list 表即可完成，即该功能是支持的。

### 5. 是否有像 redis 的 pipeline 之类的，用一次请求 (request) 就能处理多个 query? 例如，有10个 query 的话，用1个请求就能处理10个 query。

TcaplusDB 提供了批量操作来满足类似场景的需求，具体可参考 SDK:

[https://github.com/tencentyun/tcaplusdb-go-sdk/blob/v0.0.10/tdr/request/batchget\\_req.go](https://github.com/tencentyun/tcaplusdb-go-sdk/blob/v0.0.10/tdr/request/batchget_req.go)。

## List 表 index 的增长规则

### 1. idx 从0开始

2. idx 按照步长1从0开始自增，直到自增到 int 类型的最大值，然后会选择一个空闲的 idx 使用，即不管什么情况下，正在使用的 idx 不会重复。

3. 如果 key 删除，idx 从0开始，也可以保持从上次 idx 开始 默认从0开始，也可以通过 SetFlags 设置从上次 idx 开始。

```
/*TCAPLUS_FLAG_LIST_RESERVE_INDEX_HAVING_NO_ELEMENTS
```

```
* 设置此标志后，List 表删除最后一个元素时需要保留 index 和 version。
```

```
* ListDelete ListDeleteBatch ListDeleteAll 操作在删除 list 表最后一个元素时，
```

```
* 设置此标志在写入新的 List 记录时，版本号依次增长，不会被重置为1。
```

```
*
```

```
* 适用场景:
```

```
* 业务需要确定某个表在删除最后一个元素时是否需要保留 index 和 version
```

```
* 主要涉及 List 表的使用体验
```

```
*
```

```
*/ int SetFlags(int32_t flag);
```

index 是不连续的，只会增加例如：1 2 3 4 5 删除了3变成了 1 2 4 5，如果AddRecord(2)就变成了 1 2 6 4 5。

如果 key 下的 list 元素都被删除，等同于 key 也被删除。

listaddafter 支持队头删除，队尾插入；或者队头插入，队尾删除。

listdelete，删除 idx。

TCAPLUS\_API\_LIST\_PRE\_FIRST\_INDEX：新元素插入在第一个元素之前

TCAPLUS\_API\_LIST\_LAST\_INDEX：新元素插入在最后一个元素之后

**increase 相关接口的上限问题，假如当前值是97，这时候 increase5同时上限设置为100会增加成功一部分吗？**

不会更新成功，会返回 SVR\_ERR\_FAIL\_OUT\_OF\_USER\_DEF\_RANGE 错误码。

## tcaplus 对表进行 insert 操作，如果先后2次 insert 操作的主键相同，第二次是覆盖第一次，还是第二次操作失败？

第二次会报-1293 SVR\_ERR\_FAIL\_RECORD\_EXIST insert的记录已存在

## protoc 文件中可以定义的类型，tcaplus 都能使用吗？例如 any 类型

可以，但条件更新没有对 any、union 支持。

## go sdk 怎么设置遍历备表？

通过 SetOnlyReadFromSlave 接口调用设置

## ListGetAllRequest 返回报错 SVR\_ERR\_FAIL\_INVALID\_SUBSCRIPT

按照 offset 和 limit 拉取时，传的参数不对，请检查 offset 和 limit 参数设置。

## 本地索引和分布式索引（全局索引）的创建方式

本地索引在表结构文件中定义，一旦定义无法修改删除；分布式索引在对应的控制台页面（腾讯云控制台页面）创建，可以修改重建。

## 分布式索引（全局索引）支持字符串的分词模糊搜索吗？

支持like模糊查询，还不支持分词检索。

## 表定义中的字段是否可以删除？

已存在的表不能删除字段

## PB 表的 add 操作，add 因为数据已存在时，有没有办法返回已有数据？

可以，在 int SetMessageOption(const ::google::protobuf::Message &msg, int32\_t item, const std::string &option) 函数中 option 设置变更记录的所有字段旧数据。

## TcaplusDB Go SDK 支持通过域名的方式连接目录服务器（Tcapdir）吗？

支持通过域名的方式连接目录服务器（Tcapdir）。

## TcaplusDB 的 list 结构，有接口只查询 list 内元素数量吗？

目前不能查询一个 list 的 size 大小

## Tcaplus list 表元素满时默认的淘汰方式是哪种？

有三种淘汰方式，默认 TCAPLUS\_LIST\_SHIFT\_HEAD 移除最前面的元素。  
TCAPLUS\_LIST\_SHIFT\_NONE: 不允许删除元素，若 LIST 满，插入失败；

TCAPLUS\_LIST\_SHIFT\_HEAD: 移除最前面的元素;

TCAPLUS\_LIST\_SHIFT\_TAIL: 移除最后面的元素

## Tcaplus 单行记录的淘汰假如设置了300秒淘汰，是插入后300秒淘汰吗，这段时间怎么续租约呢？

表级别数据淘汰，是根据记录最后修改时间淘汰；即 update 或 replace 会自动续期。精确度大概为天级别。适合场景：如保留最近30天的邮件或战斗记录。记录级别通过 setttl 设置和续期，适合对 ttl 时间比较精确或不同记录的淘汰时间不同的场景。精确到毫秒。

## 一个 TcaplusServer 对象只能访问一个 set 内的不同 zone

1、多个 TcaplusServer 对象是不是可以实现跨 set 访问--只能访问一个 set，不能跨 set；跨 set 可以用多个 api，即多个 TcaplusServer 对象。

2、多个 TcaplusServer 对象内存占用如何--每个 api 实际内存占用在40M左右。

## UpdateNetwork api 的这个接口，内部会调用类似 sleep 或 epoll\_wait 的接口吗？

UpdateNetwork 是用户线程，不会 sleep，通过网络线程收发包，只有网络线程会 epollwait，网络线程用户不用关注。

## 同时调用了 SetResultFlag(2) SetResultFlagForFail(0) 会怎样表现呢？

建议不要一起用，SetResultFlagForFail 和 SetResultFlagForSuccess 推荐这两个函数搭配使用。如果同时调用了 SetResultFlag(2) SetResultFlagForFail(0)，会以最后一次设置为准。

## Tcaplus 可以直接使用 RepeatedPtrField 作为表字段做存储操作吗？proto 基础类型和结构类型作为基础 tcaplus表字段存储方式有什么区别吗？

可以，无区别，注意结构嵌套深可能更消耗 cpu。

## 咨询下 Tcalpus 有原子操作的 api 吗？

Tcaplus 有乐观锁，支持单 key 的事务，TCAPLUS\_CMD\_INCREASE\_REQ 自增自减操作、单 key 下的条件更新操作支持，其他操作不支持事务。

## FieldGet 接口里面的 dottedpaths 字段，如果为空，是返回所有字段呢？还是仅返回 key 字段，其他普通字段全部为空呢？

FieldGet 是返回需要的字段，如果为空的话只有 key 字段，想要返回全量字段可以使用 Get。

## getbypartkey 情形下，业务只想获得符合条件的记录条数，可以只返回记录条数吗？

可以通过调用 SetResultLimit(1) 来获取一条记录，但本次可以通过 GetRecordMatchCount 来拉取总数。

## list 类型表的 ListNum 支持修改吗？

list 类型表的 ListNum 支持改大，不能改小；ListNum 为必填属性，一般由业务预估大小在表定义中设置；

### **表结构里的 size 扩大是否需要修改字段的 version?**

不需要，可以直接修改，size 只能改大不能改小。

### **在不知道 list 表的 index 的情况下，有没有方式能获取到最新的 n 条记录?**

如果本身插入就是按时间顺序 addafter 的话，可以直接从尾部 listgetall，SetResultLimit 可以指定数量参数。或者使用 sortlist 表，加个时间字段，按时间排序。

### **API 访问报错：dir sign up fail, ret:-279，如何处理?**

检查业务密码是否正确；检查 dir 地址是否正确。

### **Tcaplus 能否支持模糊查询，以及忽略大小写?**

支持模糊查询，需要创建分布式索引查询。区分大小写。

### **同一个进程里可以有多个不同的 TcaplusService::TcaplusServer 吗?**

可以的，持有多个 service api 的 handle，分别初始化。

# 数据库原理类

最近更新时间：2024-03-06 10:09:31

- [gameserver 如何剔除某个无效的 tcaproxy（接入层）节点？](#)
- [gameserver 是怎么选择 tcaproxy（接入层）节点的？](#)
- [TcaplusDB 有压缩功能吗？](#)
- [TcaplusDB API 是线程安全的吗？](#)
- [tcapsvr（存储层）的容灾是怎么做的？](#)
- [tcaproxy（接入层）的容灾是怎么做的？](#)
- [TcaplusDB 有过载保护吗？](#)
- [TcaplusDB 的冷热数据交换原理是什么？](#)
- [tcaproxy（接入层）是怎么选择 tcapsvr（存储层）的？](#)
- [TcaplusDB 锁的级别是什么？](#)

## gameserver 如何剔除某个无效的 tcaproxy（接入层）节点？

TcaplusDB API 在这里对 tcaproxy 异常做了容灾的处理，API 剔除无效的 tcaproxy 进程的方式主要有2个：

1. API 物理上认为某个 tcaproxy 不可用，API 每隔1秒对起链接的所有的 tcaproxy 发送心跳检测包，如果某个 gameserver 在10s内没有从 tcaproxy 收到相应的心跳回包，则 API 会主动断开与 tcaproxy 的 TCP 链接，在下个 onupdate 时主动去链接该 tcaproxy。
2. API 逻辑上认为 tcaproxy 不可用，是每隔10s去计算下某个 tcaproxy 的请求和响应比，作为判断依据，其中 API 为某个请求包超时的阈值是3s，大于3次则认为该 tcaproxy 不可用，请求不会再发给该 tcaproxy，在60s后发送 getmetdata 请求，如果 tcaproxy 能够正确处理 getmetdata 的请求，则 API 再次认为该 tcaproxy 可用，请求会再次发送给该 tcaproxy。

从现象上看，gameserver 在10s内发现某个 tcaproxy 不可用，则不会再向该 tcaproxy 节点发送数据了。

## gameserver 是怎么选择 tcaproxy（接入层）节点的？

gameserver 本地维护了一致性的 Hash 环，凡是某个 tcaproxy（接入层）节点认证通过后即增加到 Hash 环上，如果某个 tcaproxy（接入层）节点扩容后或者由于机器异常导致 gameserver 与 tcaproxy（接入层）之间的 TCP 链接断掉后，gameserver 会从 Hash 环上摘除该 tcaproxy（接入层）节点。gameserver 根据请求里的主键计算 hash 值（如果是 batchget 请求，会随机的选择单个 tcaproxy（接入层）节点），然后在一致性 Hash 环上选择单个 tcaproxy（接入层）节点发送出去。

## TcaplusDB 有压缩功能吗？

TcaplusDB 有压缩功能，采用的压缩算法是 Google snappy 压缩算法，包括协议压缩，即 gameserver <--> tcaproxy（接入层）间的请求包/响应包的压缩功能；数据压缩，即 tcapsvr（存储层）在数据存储时会压缩需要存储的数据，如果您需要节省 gameserver <--> tcaproxy 间的网络流量，推荐开启协议压缩，调用

TcaplusDB API 的函数 SetCompressSwitch，推荐您开启 tcapsvr（存储层）压缩，节省磁盘空间、提高 IO 磁盘性能的同时压缩、解压缩耗费的 CPU 也是可控的问题。

## TcaplusDB API 是线程安全的吗？

TcaplusDB API 是非线程安全的，主要是 tlog、tdr 等组件是非线程安全的，推荐单个线程采用单个 API 对象，单个游戏区采用单个 API 对象，如果需要跨游戏区交互，建议单个 gameserver 维护多个 API 对象。

## tcaproxy（接入层）的容灾是怎么做的？

tcaproxy（接入层）采用对等设计方案，即单个游戏区下面的所有 tcaproxy（接入层）节点都包含了单个游戏区下所有表的路由信息，如果某个 tcaproxy（接入层）故障后，只要其余的 tcaproxy（接入层）节点不会过载，则 gameserver 会剔除异常的 tcaproxy（接入层）节点，不会影响 gameserver 的使用，tcaproxy（接入层）没有单点的风险。

## tcapsvr（存储层）的容灾是怎么做的？

tcapsvr（存储层）采用一主（tcapsvr master）一从（tcapsvr slave）的模式运行，tcapsvr master/slave 实时的在同步数据，采用同城异 IDC 部署，确保主从同步时延小于10ms，如果 tcapsvr slave 异常，不会影响 gameserver 的使用（没有开启读分流，gameserver 的请求是 tcapsvr master 处理，如果开启读分流后，tcapsvr slave 会协助处理部分读请求），DBA 进行 tcapsvr slave 重建即可；如果 tcapsvr master 异常，则 tcapsvr slave 会进行故障恢复，DBA 再申请新的机器的 tcapsvr slave 重建即可，tcapsvr（存储层）没有单点的风险。

## TcaplusDB 有过载保护吗？

接入层、存储层都有进程级的过载保护措施，保障业务高峰时服务不雪崩。

## TcaplusDB 的冷热数据交换原理是什么？

TcaplusDB 采用内存 + SSD 盘存储，单个引擎文件，前1GB映射在内存，热数据尽量放在内存，冷数据放在磁盘，采用 LRU 算法进行冷热数据交换，gameserver 的 get 操作触发 LRU 换入操作，tcapsvr（存储层）的 LRU 线程负责 LRU 换出操作，尽量保证热数据存储在内存在里，确保 cache 命中率高、单次读写延时低。

## tcaproxy（接入层）是怎么选择 tcapsvr（存储层）的？

每个表定义的有分表因子，如果没有定义分表，分表因子则默认是主键，tcaproxy（接入层）根据 hash（分表因子）%1w，选择对应的 tcapsvr（存储层），故分表因子离散度要高。

## TcaplusDB 锁的级别是什么？

TcaplusDB 锁的粒度是记录级别。

## 如果 shard 数目过大、过小会对 Tcaplus 的增删改查性能、引擎调优这些有影响吗？

影响可以忽略不计，线上大表都至少分几百个 shard。

## 业务的 shard 数量会考虑一些什么因素？

一般考虑：每个 shard 有1G在内存里，其中约500MB存 Key，300MB存 Value，尽可能多的 Key 在内存里可以提升性能，增加 shard 单 shard 最大256G，文件太大也影响一些恢复操作。

## Tcaplus 高频读数据有没有穿透的问题呢？

Tcaplus 有过载保护,穿透导致服务过载时，在接入层就会拒绝部分请求了。

## 如果瞬间涌入大量请求的话，tcaproxy 和 tcapsvr 会有排队机制吗？

会有排队机制，超过处理能力会大时延、超时等。Tcaproxy 的话，跟请求连接数有关，会按连接分散到 proxy 处理；Tcapsvr 的话，跟请求数据是否分散有关，集中在少数 shard 或 key 的话，会有排队。

## 遍历原理

Tcaplus 支持全表遍历操作,包括 generic 表和 list 表,会根据 sharding 逻辑在各个该表分布的 tcapsvr 上发起遍历任务,因此,整个遍历请求性能相对较低，建议设置从 tcapsvr slave 上遍历数据（从 tcapsvr slave 上遍历数据，不会影响 tcapsvr master 对外提供服务），即接口：SetOnlyReadFromSlave(bool flag)。

如果每次要拉取的数据不多，可以优先考虑本地索引能否满足需求。

遍历器一旦开始以后,Tcaplus 的 tcapsvr 会启动内部迭代器分时间片去遍历读取树上的节点数据,先找 key 后找 value,然后将遍历到的数据打包到 buffer 里面,返回给 service api,service api 内部的遍历逻辑会根据预期 seq 与收到的 seq 是否一致，来判断是否丢包、收到重复的包等，一旦确认收到的包符合预期，则主线程调用 RecvResponse 时可以拿到完整包，然后调用 FetchRecord 可以解析到一条业务记录，遍历可以随时中断和终止，可以有条件的恢复。整体而言，遍历在 service api 内部采用的是 ping-pong 自动触发机制，在收到符合预期的包以后会自动决定是否要继续发起下一个遍历请求(例如后端存储 svr 明确返回了 BUSY 的错误，则遍历会中断)，一直到明确遍历完成或业务停止遍历。