

Generic表和List表

TcaplusDB · 自研 NoSQL 数据库

Exported on 02/07/2021

Table of Contents

1 数据表及相关规则	4
1.1 数据表规则	4
1.2 变更规则	4
1.3 Generic表.....	4
1.4 List表.....	4
2 Generic表.....	5
2.1 TDR-Generic表示例	5
2.2 PB-Gneric表示例.....	5
3 List表.....	7
3.1 TDR-List表示例.....	7
3.2 PB-List表示例.....	8
4 SortList表	9
4.1 TDR-Sortlist表示例	10
4.2 PB-SortList表示例.....	11

TcaplusDB需要预设Schema，同时提供两种表类型：Generic表和List表，Generic表一个Key(1-8个字段)记录对应一个Value记录，List表一个Key(1-7个字段)记录对应多个Value记录。List表根据是否进行排序分为普通list表和sortlist表，对于普通List表，Tcaplus内部是无序插入的，用户取出一个Key下的所有Value时，Value内部是无序的。SortList表按照某个指定的Value字段进行排序插入，用户可以按照从大到小或者从小到大的顺序进行取值。在PB和TDR表中都存在这三种schema模式，用户可以根据需求选择合适的模式。

1 数据表及相关规则

1.1 数据表规则

- 数据表名称不超过32字节(包含'\0'结尾)
- 字段名不超过32字节(包含'\0'结尾)
- 每个Key字段不超过1024字节
- 每个Value字段不超过10MB
- 单条记录Pack后长度不超过10MB

1.2 变更规则

- 不允许对primarykey、splitkey、index做任何变更；
- Value字段可以增加字段但不能减少字段，并且增加字段时要注意版本号的变更；
- Value字段默认值可更改，Value字段最大长度只能改大不能改小；
- 不允许删除Value字段，不允许更改Value字段名称和类型；
- List类型表的最大元素个数只允许改大不允许改小，且最大元素个数不能超过10000；

1.3 Generic表

- 可以建索引（见本地索引和全局二级索引）
- 最多支持8个Key字段，256个Value字段

1.4 List表

- 不能建索引
- Key字段最多只能有7个，Value字段最多255个，各有一个要预留给系统使用。
- List支持方便的头尾操作, 适合邮件，留言板，战斗记录等场景
- List表需要指定单个Key记录下的最大元素个数（目前最大10000），超过最大元素个数时，可指定从头部或尾部删除老元素。默认采用FIFO方式淘汰
- 可以一次取出单个Key下的所有Value，Value内部的排列无序。

2 Generic表

2.1 TDR-Generic表示例

```
<struct name="PLAYERONLINECNT" version="1" primarykey="TimeStamp,GameSvrID" splittablekey="TimeStamp">
  <entry name="TimeStamp"           type="uint32"      desc="单位为分钟" />
  <entry name="GameSvrID"            type="string"     size="64" />
  <entry name="GameAppID"            type="string"     size="64" desc="gameapp id" />
  <entry name="OnlineCntIOS"         type="uint32"     defaultvalue="0" desc="ios在线人数" />
  <entry name="OnlineCntAndroid"     type="uint32"     defaultvalue="0" desc="android在线人数" />
  <entry name="BinaryLen"            type="smalluint" defaultvalue="1" desc="数据来源数据长度；长度为0时，忽略来源
检查"/>
  <entry name="binary"               type="tinyint"    desc="二进制" count= "1000" refer="BinaryLen" />
  <entry name="binary2"              type="tinyint"    desc="二进制2" count= "1000" refer="BinaryLen" />
  <entry name="strstr"               type="string"     size="64" desc="字符串"/>
  <index name="index_id" column="TimeStamp"/>
</struct>
```

2.2 PB-Gneric表示例

```
message pb_generic_index_shardingkey {
  option(tcapluservice.tcaplus_primary_key) = "openid,tconndid,timekey,svrid";
  option(tcapluservice.tcaplus_index) = "index_openid(openid)";
  option(tcapluservice.tcaplus_index) = "index_openid_tconndid(openid,tconndid)";
    option(tcapluservice.tcaplus_index) = "index_full_key(openid,tconndid,timekey,svrid)";

    option(tcapluservice.tcaplus_sharding_key) = "openid";

    //4个key
    required uint32 openid = 1; //QQ Uin
    required string timekey = 2[(tcapluservice.tcaplus_size) = 20, (tcapluservice.tcaplus_desc) =
"bingo"];
    required int32 tconndid = 3;
    required string svrid = 4;

    //value
    required string gamesvrid = 5[(tcapluservice.tcaplus_size) = 11];
    repeated property other_property= 6 ;//其他扩展属性
    optional item_bag items = 7;
    repeated int64 lockid = 8 [packed = true];
    optional bytes val = 9;
    optional pay_info pay = 10;
    optional uint32 id_uint32 = 11;
    optional int32 id_int32 = 12;
}
```


3 List表

Tcaplus对List表的存储采用分级机制，包括：

- 索引记录, Fullkey + (idx1,idx2,idx3,.....,idxn)
- 数据记录, [FullKey+idx1,value1][FullKey+idx2,value2][.....][FullKey+idxn,valuen]

3.1 TDR-List表示例

```
<struct name="following_action_list" version="1" primarykey="game,myuin">
  <entry name="game" type="uint64" defaultvalue="0" desc="游戏
ID"/>
  <entry name="myuin" type="uint32"
desc="QQ号"/>
  <entry name="actiontype" type="uint8" defaultvalue="0" desc="1-分享图片, 2-赞
图片, 3-评论图片"/>
  <entry name="uin2" type="uint32"
desc="关注人QQ号"/>
  <entry name="nick2" type="string" size="128"
desc="关注人昵称"/>
  <entry name="sex" type="uint8" defaultvalue="0" desc="关注人
性别男0女1"/>
  <entry name="pid" type="string" size="33"
desc="关注人操作图片ID"/>
  <entry name="time" type="uint32"
desc="时间"/>
  <entry name="content" type="string" size="1024" desc="动态详
细内容"/>
</struct>
```

3.2 PB-List表示例

```
syntax = "proto3";

package myTcaplusTable;

import "tcapluservice.optionv1.proto";

message tb_online_list {
    option(tcapluservice.tcaplus_primary_key) = "openid,tconndid,timekey";
    option(tcapluservice.tcaplus_customattr) = "TableType=LIST;ListNum=1900";

    int32 openid = 1; //QQ Uin
    int32 tconndid = 2;
    string timekey = 3;
    string gamesvrid = 4;
    int32 logintime = 5 ;
    repeated int64 lockid = 6;
    pay_info pay = 7;

    message pay_info {
        uint64 total_money = 1;
        uint64 pay_times = 2;
    }
    map<string, pay_info> projects = 8;
}
```


4 SortList表

最多允许4个排序字段，且在建表的xml（tdr）中指定，具体格式如下：

```
<struct name="following_action_list" version="1" primarykey="game,myuin"
customattr="TableType=SortList;SortRule=INSC;SortFieldNum=1">
    <entry name="game" type="uint64" defaultvalue="0" desc="游戏
ID"/>
    <entry name="myuin" type="uint32"
desc="QQ号"/>
    <entry name="actiontype" type="uint8" defaultvalue="0" desc="1-分享图片, 2-赞
图片, 3-评论图片"/>
    <entry name="uin2" type="uint32"
desc="关注人QQ号"/>
    <entry name="nick2" type="string" size="128"
desc="关注人昵称"/>
    <entry name="sex" type="uint8" defaultvalue="0" desc="关注人
性别男0女1"/>
    <entry name="pid" type="string" size="33"
desc="关注人操作图片ID"/>
    <entry name="time" type="uint32" customattr="sort1" desc="时间"/>
    <entry name="content" type="string" size="1024" desc="动态详
细内容"/>
</struct>
```

如上customattr="TableType=SortList;ListNum=1023;SortFieldNum=1"表示该表格类型为SortList, SortRule=INSC表示升序排列, SortFieldNum=1表示排序字段有1个, customattr="sort1"表示第一个排序字段。

- 排序默认按照从小到大进行排序，在ServiceApi中可以指定是从小到大取还是从大到小取。
- 暂时不允许在表变更时从无序List变为SortList，也不允许从SortList变更为无序List，不允许表变更变换排序字段的顺序以及增减排序字段（这个均采用自己写so走表变更Key-Value方式实现）。
- 排序字段最大字节数8B，排序字段的类型：byte, uint16,uint32,uint64,int16,int32,int64,float,double【string(包含\0最多8B，暂时不支持)】。
- 排序是指Value字段参与排序，而不是Key。

使用说明：

排序

当已有数据结构排好序后，再采用ListAddAfter进行数据插入时，采用插入排序效果最佳。

插入

对于ListAddAfter，流程是：先看是否已经满了，如果满了且不允许淘汰元素，则插入失败，如果满了，允许淘汰，则删掉那个元素，并且获取一个BiggestIndex，将新元素插在对应位置，并挪动其他元素。

而对于SortList，当用户进行ListAddAfter时，需要变更下ServiceApi里面的接口说明，因为我们默认是按照从小到大的顺序进行排序，当用户需要淘汰最后一个元素时，即淘汰最大的元素，当用户需要淘汰队头的元素时，即淘汰最小的元素。

4.1 TDR-Sortlist表示例

```
<struct name="table_Sortlist_single" primaryKey="key, name"
customattr2="TableType=SortLIST;ListNum=1023;SortFieldNum=1;SortRule=DESC" version="5" desc="用于list表遍历测试, 需要4个shard, 建表list最大1023个元素" >
  <entry name="key" type="uint32" desc="单个uint32作为KEY的时候, hashcode = key % 10000"/>
  <entry name="name" type="int16" />
  <entry name="level" type="uint32" />
  <entry name="value1" type="string" size="102400" defaultvalue="" desc="最大长度:100KB"/>
  <entry name="value2" type="string" size="102400" defaultvalue="" desc="最大长度:100KB"/>
  <entry name="type_int8" type="int8" desc="type_int8" />
  <entry name="type_uint8" type="uint8" desc="type_uint8"/>
  <entry name="type_int16" type="int16" desc="type_int16" customattr2="sort1"/>
  <entry name="type_uint16" type="uint16" desc="type_uint16"/>
  <entry name="type_int32" type="int32" desc="type_int32"/>
  <entry name="type_uint32" type="uint32" desc="type_uint32"/>
  <entry name="type_int64" type="int64" desc="type_int64"/>
  <entry name="type_uint64" type="uint64" desc="type_uint64"/>
  <entry name="type_float" type="float" desc="type_float"/>
  <entry name="type_double" type="double" desc="type_double"/>
  <entry name="type_short" type="short" desc="type_short"/>
  <entry name="type_string" type="string" desc="type_string" size="20"/>
  <entry name="type_tinyint" type="tinyint" desc="type_tinyint"/>
  <entry name="type_datetime" type="datetime" desc="type_datetime"/>
</struct>
```

4.2 PB-SortList表示例

```
message pb_sortedlist {
  option(tcapluservice.tcaplus_primary_key) = "openid,tconndid,timekey,svrid";
  option(tcapluservice.tcaplus_customattr) = "TableType=SortList;ListNum=1900;SortField=id_int32";

  //4个key
  required uint32 openid = 1; //QQ Uin
  required string timekey = 2[(tcapluservice.tcaplus_size) = 20, (tcapluservice.tcaplus_desc) =
"bingo"];
  required int32 tconndid = 3;
  required string svrid = 4;

  //value
  required string gamesvrid = 5[(tcapluservice.tcaplus_size) = 11];
  repeated property other_property= 6 ;//其他扩展属性
  optional item_bag items = 7;
  repeated int64 lockid = 8 [packed = true];
  optional bytes val = 9;
  optional pay_info pay = 10;
  optional uint32 id_uint32 = 11;
  optional int32 id_int32 = 12;
}
```