OceanBase测试工具

# 测试概述

* 验证整个OceanBase系统和各个server在一定压力下是否工作正常，数据是否一致。
* 连续测试数天，add cell，update cell，insert row，delete row达到100亿次，get，scan达到1000亿次，总数据量达到T级别。
* 完全随机测试，对多个table，多线程随机add cell，update cell，insert row，delete row，多线程随机get或scan单列、多列、单行、或多行。
* 持续不断地对get和scan的结果根据规则进行验证。
* 即使CS在做合并或是迁移，或是schema发生改变，测试工具正常工作。

# 功能介绍

* 将OceanBase系统作为一个黑盒进行数据一致性测试，包括功能测试和高并发，高负载压力测试。
* 即可以对OceanBase系统进行测试，也可以对单个服务器进行测试。
* 对OceanBase系统主要测试add cell，update cell，insert row，delete row，get和scan接口，这些测试都是以验证数据一致性为基础。
* 测试的规则可以配置，即可以测试复杂的规则集合，也可以选择测试单个简单规则，同时也可以配置相对复杂的规则用于长时间压力测试。
* 当不需要数据一致性验证时，该工具可以用于产生benchmark测试的负载。
* 该工具可以同时在多个客户机上运行多个进程，确保能提供足够负载。

# 概要设计

该工具的主要设计思想是根据规则和辅助列来验证高并发，高压力下的数据一致性。我们规定每个table的原始列和辅助列成对出现，利用OceanBase处理每个请求的原子性，在一个请求中更新原始列和辅助列，保证多线程更新原始列和辅助列的原子性，辅助列与原始列为互为相反数的关系，多线程随机在同一请求中读取原始列和辅助列，并比较二者是否互为相反数，如果互为相反数，校验成功，否则数据不一致。

## schema规则

schema中原始列和辅助列必须一一对应，数据类型相同，column id相差1000，当schema发生变化时，也必须保证该规则成立。原始列用于存储实际的列数据，辅助列用于存储原始列数据（int）或是hash值（int以外的类型）的相反数。测试工具并不关心每列的数据类型，测试工具能处理在ObObj中定义的所有数据类型。

注意：两个特例，create time和modify time类型的列没有辅助列，因为这两种类型不允许修改。Float和double类型的列的辅助列仍为float或double类型，辅助列的值与原始列相同，应为float和double的精度问题，不能使用相反数进行校验。

为了简化测试工具的设计，限制一个schema中只能包含一个宽表和一个join表，如有支持多个宽表和多个join表的需求，以后再做扩展。

Schema举例：

[jtable]

table\_id=1001

table\_type=1

column\_info=0,2, join\_int,int

column\_info=0,3,join\_str,varchar

column\_info=0,1002,expected\_join\_int,int

column\_info=0,1003,expected\_join\_str,int

rowkey\_split=0

rowkey\_max\_length=8

[wtable]

table\_id=1002

table\_type=2

column\_info=1,2,wtable\_int,int

column\_info=1,3,join\_int,int

column\_info=1,4,wtable\_str,vchar

column\_info=1,5,join\_str,vchar

column\_info=1,1002,expected\_wtable\_int,int

column\_info=1,1003,expected\_join\_int,int

column\_info=1,1004,expected\_wtable\_str,int

column\_info=1,1005,expected\_join\_str,int

join=rowkey[8,15]%jtable:join\_int$joint\_int, expected\_join\_int$expected\_joint\_int, join\_str$ join\_str, expected\_join\_str$expected\_join\_str,

rowkey\_split=8

rowkey\_max\_length=16

该schema中定义了两张表，wtable和jtable，wtable中有4个原始列，分别为wtable\_int，wtable\_str，join\_int和join\_str，其中join\_int和join\_str来自jtable。Wtable中有4个辅助列，分别为expected\_wtable\_int，expected\_wtable\_str，expected\_join\_int和expected\_join\_str，其中expected\_join\_int和expected\_join\_str来自jtable。Jtable中有2个原始列，分别为join\_int和join\_str，另外jtable中还有2个辅助列，分别为expected\_join\_int和expected\_join\_str。Wtable的rowkey长度为16字节，jtable的rowkey为8字节，wtable的rowkey的后8个字节作为join的外键。

CS中会存放两张静态表，一张宽表，一张join表，这两张表的静态数据由数据生成工具生成，数据生成工具用gen\_sstable。

## rowkey规则

宽表的rowkey共16字节，由两部分组成，分别为perfix和suffix，各占8个字节，suffix为join表的rowkey，占8个字节。Prefix和suffix都用一个从0开始递增的int64\_t来表示，假设宽表中rowkey全为0的行中的前两列分别存储了perfix当前已写入的最大值和suffix当前已写入的最大值。通过prefix和suffix就可以知道在CS中存储的所有数据的rowkey范围。

当测试工具启动时，会首先读取宽表中rowkey全为0的行的前两列，获得当前已写入的prefix和suffix并存为全局变量，当向宽表中insert了新的prefix，会原子增加prefix的值，同样，如果向join表中添加了新的行，也需要原子递增suffix。从而使测试工具在任何时候都能得到当前系统的宽表和join表的rowkey范围。在测试工具退出时，可以将宽表中rowkey全为0的行更新到宽表中，以便下次接着测试。

注意：原子递增prefix或是suffix，并不是每次都是递增1，而是根据启动的测试工具的数量来决定的，比如需要启动10个测试工具，则原子递增的步长为10，这是因为宽表中存储的prefix和suffix是静态CS中表的rowkey范围，不会被修改，每个测试工具进程都会读取到相同的静态prefix和suffix，为了多个测试工具进程不会产生相同的prefix和suffix，所以每个测试工具进程都有一个初始编号，假设需要启动10个测试工具进程，编号为1的进程添加的第一个prefix为org\_prefix+1，第二个prefix为org\_prefix+1+10，第n个prefix为org\_prefix+1+10\*（n-1）。

用于insert操作的宽表rowkey前缀使用当前已写入的prefix递增的方式产生，suffix随机选择一个在join表中当前已写入的suffix。用于add，update，delete和get操作的宽表rowkey的prefix从当前已写入的prefix范围中随机选择一个，suffix从当前已写入的suffix范围中随机选择一个。

用于insert操作的join表的rowkey使用当前已写入的suffix递增的方式产生。用于add，update，delete和get操作的join表的rowkey从当前已写入的suffix范围中随机选择一个。

用于scan操作的宽表rowkey范围直接在当前已写入的prefix中随机选取，用于scan操作的join表的rowkey范围直接在当前已写入的suffix中随机选取。

## 操作数据和操作类型生成

支持的操作包括add cell，update cell，insert row，delete row，delete cell，get和scan。根据各个操作的比例来随机选择执行那种操作。如果操作比例没有配置，将会操作随机产生操作。

1. **Add cell**

产生一个随机数作为需要操作的cell数量m，接着产生一个随机数作为每个cell需要add的次数n，然后随机选择宽表或是join表中的int列，将每个int原始列添加n个加1的add操作，并将每个int辅助列添加一个减n的add操作。

注意：随机产生的rowkey可能不存在，OceanBase系统能处理这种情况。

1. **Update cell**

产生一个随机数作为需要操作的cell数量m，然后随机选择宽表或是join表中的任意m个原始列，根据原始列的type随机产生update数据，同时用rowkey的前8个字节转化成int64\_t后与原始列数据（int）或是hash值（int以外的类型）的差值update每个原始列对应的辅助列。

注意：随机产生的rowkey可能不存在，OceanBase系统能处理这种情况。另外宽表中相关的join字段是不能更新的，UPS已经限制了这种操作，这里是故意为之，检查系统是否工作正常。

1. **Insert row**

产生一个随机数作为需要insert row的数量m，然后随机选择宽表或是join表中的任意m个新的rowkey，根据schema中的table信息，根据各列的type随机生成数据填充相应的原始列，使用rowkey的前8个字节转化成int64\_t后与原始列数据（int）或是hash值（int以外的类型）的差值填充辅助列。

注意：insert row必须成功。

1. **Delete row**

产生一个随机数作为需要delete row的数量m，然后随机选择宽表或是join表中的任意m个rowkey，执行删除操作。

注意：随机选择的rowkey可能不存在，尽量控制执行delete row的操作比例，尽量少删除join表中的row。在我们的测试中可能join表的行数量比宽表行数量少得多，所以尽量删除宽表的row。

1. **Delete cell（update cell null）**

产生一个随机数作为需要操作的cell数量m，然后随机选择宽表或是join表中的任意m个原始列，将m个原始列和对应的辅助列设成ObNullType。

注意：随机产生的rowkey可能不存在，OceanBase系统能处理这种情况。

1. **Get**

产生一个随机数作为需要get的cell数量m，然后随机选择宽表或是join表中的任意m个原始列和辅助列，当然get也可以随机选择读取整行，对get成功返回的列，校验原始列数据（int）或是hash值（int以外的类型）加上辅助列的值是否为0。如果原始列为ObNullType，则校验辅助列是否为0。

注意：get的row或是cell可能不存在。

1. **Scan**

产生一个随机数作为需要scan的rowkey数量m，然后随机选择宽表或是join表中的任意连续m个prefix或是suffix组装成rowkey的范围，执行scan操作，见返回的每列数据，校验原始列数据（int）或是hash值（int以外的类型）加上辅助列的值是否为0。如果原始列为ObNullType，则校验辅助列是否为ObNullType。

## 随机混入错误

在以上的操作类型生成中已经随机混入了错误，比如row not exist、null cell，更新宽表中的的join列等等。

## 数据验证

数据验证相对简单，只需要将get或是scan返回的数据，校验原始列数据（int）或是hash值（int以外的类型）加上辅助列的值是否为0。如果原始列为ObNullType，则校验辅助列是否为0。对create time，modify time，float和double类型的列需要做特殊处理。

对每种写操作都分为valid和invalid两种类型，对valid的操作，必须成功，如果操作不成功，则有问题，如果为invalid操作，结果必须失败，如果invalid操作成功了则有问题。

## 线程调度

在测试工具中包括两类线程，读线程和写线程，读线程用于get和scan操作，写线程用于add cell，update cell，insert row，delete row，delete cell，一般情况下，读线程会比写线程多。

该工具可以同时启动多个进程，每个进程可以配置读线程和写线程的数量，不同的配置可以测试不同的服务器。如果配置成全为读线程，用于校验CS的静态数据是否一致。如果配置一定的读线程和写线程，并且CS中不存在静态数据，那么工具用于测试UPS数据是否一致。如果配置一定的读线程和写线程，并且CS中存在静态数据，用于测试整个OceanBase系统是否工作正常。测试中可以根据需要配置读线程和写线程的数量，比如可以只启动写线程运行一段时间，然后停止测试工具，然后只启动读线程来测试系统中的数据是否一致。

# 工具局限

* 工具依赖了每次请求的原子性，将原始列数据（int）或是hash值（int以外的类型）的相反数存入辅助列，校验规则比较简单。更复杂的校验规则准备以后再添加。
* 由于随机产生rowkey可能不存在，多线程并发操作的情况下也不能确定某个rowkey是否一定存在，所以对delete row这样的操作没法验证正确性。
* 测试中加入了许多随机错误，特别是访问和修改不存在的rowkey，更新宽表中join的字段。