

Cedar 项目组测试文档	
文档名称	truncate 测试案例
作者（测试人员）	李捷荧
功能模块	truncate
开发人员	贺小龙
日期	20170711
负责老师	测试：张蓉 开发：

修订记录：

日期	修改描述
20170711	创建文档
20170804	完善文档

硬件配置：

机器 IP	硬件配置
10.11.1.190-10.11.1.198	<p>CPU：Intel(R) Xeon(R) CPU E5-2620 * 2，2 * 6 * 2 个线程，主频 2000MHz- 2500MHz，L3 缓存 15MB；</p> <p>内存：168GB、152GB、158 GB、168GB、80GB、144GB、128 GB、112 GB、144 GB、128 GB；</p> <p>网络带宽：1000Mb/s（有少数部分机器之间的网络带宽为 100Mb/s）；</p> <p>磁盘 IOPS：76*4=304，磁盘带宽 400MB/s、6500MB/s（读缓存）</p>

功能测试案例

功能 & 语法案例

编号	1	配置	190：RS、UPS、MS、CS 191：RS、UPS、MS、CS 193：RS、UPS、MS、CS 190、191 为备集群、193 为主集群
测试目的	插入少量数据，truncate 功能验证，增删改查		
测试输入	<pre> ALTER SYSTEM SET merge_delay_interval = 1 SERVER_TYPE = CHUNKSERVER; -- 需要等待该时间后才开始合并 ALTER SYSTEM SET min_major_freeze_interval = 1 SERVER_TYPE = UPDATESERVER; -- 两次升级主版本的最小时间间隔 ALTER SYSTEM SET min_merge_interval = 1 SERVER_TYPE </pre>		

	<pre> = UPDATESERVER; -- 两次合并最小时间间隔 create table t1(c1 int primary key, c2 int); insert into t1 values(1,1),(2,2),(3,3); create table t2(c1 int primary key, c2 int); insert into t2 values(1,1),(2,2),(3,3),(4,4); truncate table if exists t1,t2; select * from t1; select * from t2; bin/ups_admin -a 10.11.1.190 -p 14701 -t minor_freeze bin/rs_admin -r 10.11.1.190 -p 14500 stat -o merge insert into t1 values(1,1),(2,2); update t1 set c2=0; delete from t1 where c1=2; insert into t2 values(3,3),(4,4); update t2 set c2=1; delete from t2 where c1=4; select * from t1; select * from t2; drop table t1,t2; </pre>		
测试结果	通过，truncate 后查询结果为空，合并后可插入数据并查询到相应数据 高压下：通过		
编号	2	配置	190: RS、UPS、MS、CS 191: RS、UPS、MS、CS 193: RS、UPS、MS、CS 190、191 为备集群、193 为主集群
测试目的	truncate 功能验证，插入大量数据		
测试输入	<pre> ALTER SYSTEM SET merge_delay_interval = 1 SERVER_TYPE = CHUNKSERVER; -- 需要等待该时间后才开始合并 ALTER SYSTEM SET min_major_freeze_interval = 1 SERVER_TYPE = UPDATESERVER; -- 两次升级主版本的最 小时间间隔 ALTER SYSTEM SET min_merge_interval = 1 SERVER_TYPE = UPDATESERVER; -- 两次合并最小时间间隔 set @@session.ob_query_timeout=9000000000; -- 设置长一 点的查询时间 create table update_test (c1 int primary key, c2 int, c3 float, c4 float, c5 double, c6 double, c7 varchar(20), c8 int); for(int i=0;i<2000000; i++) insert into update_test values(?,?,?,?,?,?,?); </pre>		

	<pre>truncate table update_test; select * from update_test; insert into update_test values(1,1,1,1,1,1,'aaa',1); select * from update_test; drop table update_test;</pre>		
测试结果	通过，truncate 后查询结果为空，合并后可插入数据并查询到相应数据（查询速度慢） 高压下：通过		
编号	3	配置	190: RS、UPS、MS、CS 191: RS、UPS、MS、CS 193: RS、UPS、MS、CS 190、191 为备集群、193 为主集群
测试目的	多次使用 truncate 语法，是否正确		
测试输入	<pre>ALTER SYSTEM SET merge_delay_interval = 1 SERVER_TYPE = CHUNKSERVER; -- 需要等待该时间后才开始合并 ALTER SYSTEM SET min_major_freeze_interval = 1 SERVER_TYPE = UPDATESERVER; -- 两次升级主版本的最 最小时间间隔 ALTER SYSTEM SET min_merge_interval = 1 SERVER_TYPE = UPDATESERVER; -- 两次合并最小时间间隔 create table t1(c1 int primary key, c2 int); insert into t1 values(1,1),(2,2),(3,3); create table t2(c1 int primary key, c2 int); insert into t2 values(1,1),(2,2),(3,3),(4,4); truncate table if exists t1,t2; truncate table if exists t1,t2; truncate table if exists t1,t2; truncate table if exists t1,t2; select * from t1; -- 应为空 select * from t2; -- 应为空 bin/ups_admin -a 10.11.1.190 -p 14701 -t major_freeze bin/rs_admin -r 10.11.1.190 -p 14500 stat -o merge insert into t1 values(1,1); insert into t2 values(3,3); select * from t1; select * from t2; truncate table if exists t1,t2; bin/ups_admin -a 10.11.1.190 -p 14701 -t major_freeze</pre>		

	<pre>bin/rs_admin -r 10.11.1.190 -p 14500 stat -o merge insert into t1 values(2,2); insert into t2 values(4,4); select * from t1; select * from t2; truncate table if exists t1,t2; bin/ups_admin -a 10.11.1.190 -p 14701 -t major_freeze bin/rs_admin -r 10.11.1.190 -p 14500 stat -o merge insert into t1 values(5,5); insert into t2 values(5,5); select * from t1; select * from t2; drop table t1,t2;</pre>		
测试结果	通过 高压下：通过		
编号	4	配置	190: RS、UPS、MS、CS 191: RS、UPS、MS、CS 193: RS、UPS、MS、CS 190、191 为备集群、193 为主集群
测试目的	换主之后在新集群上使用 truncate		
测试输入	<pre>ALTER SYSTEM SET merge_delay_interval = 1 SERVER_TYPE = CHUNKSERVER; -- 需要等待该时间后才开始合并 ALTER SYSTEM SET min_major_freeze_interval = 1 SERVER_TYPE = UPDATESERVER; -- 两次升级主版本的最 小时间间隔 ALTER SYSTEM SET min_merge_interval = 1 SERVER_TYPE = UPDATESERVER; -- 两次合并最小时间间隔 create table t1(c1 int primary key, c2 int); insert into t1 values(1,1),(2,2),(3,3); create table t2(c1 int primary key, c2 int); insert into t2 values(1,1),(2,2),(3,3),(4,4); bin/rs_admin -r 10.11.1.190 -p 14500 reelect truncate table t1,t2; select * from t1; select * from t2; bin/ups_admin -a 10.11.1.190 -p 14701 -t major_freeze</pre>		

	<pre> insert into t1 values(5,5),(6,6); update t1 set c2=0; delete from t1 where c1=6; insert into t2 values(5,5),(6,6); update t2 set c2=0; delete from t2 where c1=6; select * from t1; select * from t2; drop table t1,t2; </pre>		
测试结果	通过，结果正确 高压下：通过		
编号	5	配置	190: RS、UPS、MS、CS 191: RS、UPS、MS、CS 193: RS、UPS、MS、CS 190、191 为备集群、193 为主集群
测试目的	下线主 UPS 之后在新集群上使用 truncate		
测试输入	<pre> ALTER SYSTEM SET merge_delay_interval = 1 SERVER_TYPE = CHUNKSERVER; -- 需要等待该时间后才开始合并 ALTER SYSTEM SET min_major_freeze_interval = 1 SERVER_TYPE = UPDATESERVER; -- 两次升级主版本的最 最小时间间隔 ALTER SYSTEM SET min_merge_interval = 1 SERVER_TYPE = UPDATESERVER; -- 两次合并最小时间间隔 create table t1(c1 int primary key, c2 int); insert into t1 values(1,1),(2,2),(3,3); create table t2(c1 int primary key, c2 int); insert into t2 values(1,1),(2,2),(3,3),(4,4); kill master ups truncate table t1,t2; select * from t1; select * from t2; # 新的主集群上每日合并 bin/ups_admin -a 10.11.1.190 -p 14701 -t major_freeze insert into t1 values(5,5),(6,6); update t1 set c2=0; delete from t1 where c1=6; insert into t2 values(5,5),(6,6); update t2 set c2=0; </pre>		

	delete from t2 where c1=6; select * from t1; select * from t2; drop table t1,t2;		
测试结果	通过，修复 bug3 高压下：通过		
编号	6	配置	190: RS、UPS、MS、CS 191: RS、UPS、MS、CS 193: RS、UPS、MS、CS 190、191 为备集群、193 为主集群
测试目的	建立多张表，并插入少量数据，下线所有 UPS，重启所有 UPS 后在主集群上使用 truncate		
测试输入	<pre> ALTER SYSTEM SET merge_delay_interval = 1 SERVER_TYPE = CHUNKSERVER; -- 需要等待该时间后才开始合并 ALTER SYSTEM SET min_major_freeze_interval = 1 SERVER_TYPE = UPDATESERVER; -- 两次升级主版本的最小时间间隔 ALTER SYSTEM SET min_merge_interval = 1 SERVER_TYPE = UPDATESERVER; -- 两次合并最小时间间隔 create table t1(c1 int primary key, c2 int); insert into t1 values(1,1),(2,2),(3,3); create table t2(c1 int primary key, c2 int); insert into t2 values(1,1),(2,2),(3,3),(4,4); kill all ups truncate table t1,t2; select * from t1; select * from t2; ## 每日合并 bin/ups_admin -a 10.11.1.190 -p 14701 -t major_freeze insert into t1 values(5,5),(6,6); update t1 set c2=0; delete from t1 where c1=6; insert into t2 values(5,5),(6,6); update t2 set c2=0; delete from t2 where c1=6; select * from t1; select * from t2; drop table t1,t2; </pre>		

测试结果	通过，结果正确 高压下：通过，结果正确，SERVER 无异常		
编号	7	配置	190: RS、UPS、MS、CS 191: RS、UPS、MS、CS 193: RS、UPS、MS、CS 190、191 为备集群、193 为主集群
测试目的	建二级索引，插入数据，使用 truncate，二级索引表数据是否也清空		
测试输入	<pre> ALTER SYSTEM SET merge_delay_interval = 1 SERVER_TYPE = CHUNKSERVER; -- 需要等待该时间后才开始合并 ALTER SYSTEM SET min_major_freeze_interval = 1 SERVER_TYPE = UPDATESERVER; -- 两次升级主版本的最 小时间间隔 ALTER SYSTEM SET min_merge_interval = 1 SERVER_TYPE = UPDATESERVER; -- 两次合并最小时间间隔 set @@session.ob_query_timeout=9000000000; -- 设置 长一点的查询时间 create table t1(c1 int primary key, c2 int); create index t1_index on t1(c2); create table t2(c1 int primary key, c2 int); create index t2_index on t2(c2); #每日合并确保索引生效 bin/ups_admin -a 10.11.1.190 -p 14701 -t major_freeze insert into t1 values(1,1),(2,2),(3,3); insert into t2 values(1,1),(2,2),(3,3),(4,4); insert into t1 values(5,5); insert into t2 values(5,5); truncate table t1,t2; select * from t1; -- 应为空 select * from t2; -- 应为空 select * from __t1_t1_index; -- 应为空 select * from __t2_t2_index; -- 应为空 bin/ups_admin -a 10.11.1.190 -p 14701 -t major_freeze insert into t1 values(6,6); insert into t2 values(6,6); select * from t1; -- 应只有 6 select * from t2; -- 应只有 6 select * from __t1_t1_index; select * from __t2_t2_index; drop table t1,t2; </pre>		

测试结果	通过，修复 bug1，bug2。 高压下：通过，结果正确，SERVER 正常
------	---

性能测试案例

编号	1	配置	190：RS、UPS、MS、CS 191：RS、UPS、MS、CS 193：RS、UPS、MS、CS 190、191 为备集群、193 为主集群
测试目的	一个是 dev 分支，一个是有 truncate 模块的分支，表结构一致，查询性能对比（truncate 模块多了一次查询交互）		
测试输入	表结构：select_test(c1 int primary key,c2 int,c3 int,c4 float,c5 float,c6 float, c7 double, c8 double,c9 double, varchar(20)) 数据：200 万 负载：主键单点查询		
测试结果	单集群 1MS（5 次平均）		
	Dev 分支		
	线程	QPS	QRS（ms）
	400	13852	14.326
	200	14031	7.113
	100	14191	3.518
	Truncate 模块		
	线程	QPS	QRS（ms）
	400	12022	16.515
	200	12030	8.276
	100	13070	3.824
	性能差别：QPS 下降 1000-2000 左右		
	三集群		
	Dev 分支		
	线程	QPS	QRS（ms）
	400	14137	14.018
200	14214	7.011	
100	14133	3.531	
Truncate 模块			
线程	QPS	QRS（ms）	
400	12183	16.275	
200	12128	8.212	
100	12913	3.867	
性能差别：QPS 下降 1000-2000 左右			
编号	2	配置	190：RS、UPS、MS、CS 191：RS、UPS、MS、CS 193：RS、UPS、MS、CS

			190、191 为备集群、193 为主集群
测试目的	比较 truncate，delete，drop 三种方式的性能差别		
测试输入	表结构：update_test(c1 int primary key,c2 int,c3 float,c4 float, c5 double, c6 double,c7 varchar(20), c8 int) 数据：200 万 分别执行： truncate table update_test; delete from update_test; drop table update_test;		
测试结果	Truncate 花费时间：58.996 毫秒 Delete 花费时间：200 万记录超时，2 万记录花费 9.207 秒 Drop 花费时间：866.141 毫秒 性能：truncate > drop >>> delete 正确，因为 Truncate 直接修改内存表，而 DROP 需要跟 RootServer 交互，流程不一样。		

稳定性测试

编号	1	配置	190: RS、UPS、MS、CS 191: RS、UPS、MS、CS 193: RS、UPS、MS、CS 190、191 为备集群、193 为主集群
测试目的	长时间不断重复 replace 数据，检测 truncate 模块是否引入不稳定因素		
测试输入	表结构: t0(c1 int primary key,c2 int,c3 varchar(20)) 数据: 长时间 replace 200 万		
测试结果	2017-08-03 16:39 开始 2017-08-04 9:10 结束 正常, SERVER 无异常, 能继续提供服务		