Cedar 项目组测试文档	
文档名称	Auto_Increment 测试案例
作者（测试人员）	李捷荧
功能模块	Auto_Increment
开发人员	刘柏众、黄建伟
日期	20161206
负责老师	测试：张蓉　　开发：张召

修订记录：

日期	修改描述
20161206	创建文档
20170320	完善文档
20170629	回归测试

硬件配置：

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

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 192-198：RS、UPS、LMS + 6 * (MS + CS) 190、191 为备集群、193 为主集群
测试目的	自增后再插入主键有间隔的数据后，自增情况		

测试输入	单集群和三集群分别测试：
	create table t1 (c1 int primary key auto_increment, c2 int);
	insert into t1(c2) values(1),(2),(3);
	insert into t1 values(6,6);
	insert into t1 values(4,4);		

	insert into t1(c2) values(1),(1),(1);
	select * from t1;
	
	create table t2 (c1 int primary key auto_increment, c2 int);
	replace into t2(c2) values(1),(2),(3);
	replace into t2 values(6,6);
	replace into t2 values(4,4);
	replace into t2(c2) values(1),(1),(1);
	select * from t2;		

测试结果	单集群：通过（修复 bug1、bug2、bug3） 高压自增单集群：通过（修复 bug7） 三集群：通过 高压自增三集群：通过 回归测试：通过，（修复 bug10）		
编号	2	配置	190：RS、UPS、MS、CS 191：RS、UPS、MS、CS 192-198：RS、UPS、LMS + 6 * (MS + CS) 190、191 为备集群、193 为主集群
测试目的	自增列插入重复值		

测试输入	单集群和三集群分别测试：
	create table t1 (c1 int primary key auto_increment, c2 int);
	insert into t1(c2) values(1),(1),(1);
	insert into t1(c1,c2) values(1,1);
	insert into t1(c1,c2) values(3,3);
	replace into t1 values(2,2);
	replace into t1 values(4,4);
	insert into t1(c2) values(5);
	select * from t1;		

测试结果	单集群：通过，insert 能够报主键重复，replace 正常 高压自增单集群：通过（修复 bug7） 三集群：通过 高压自增三集群：通过 回归测试：通过		
编号	3	配置	190：RS、UPS、MS、CS 191：RS、UPS、MS、CS 192-198：RS、UPS、LMS + 6 * (MS + CS) 190、191 为备集群、193 为主集群
测试目的	将原有主键值更新为下一条新插入记录的主键值，是否会报重复主键错误		
测试输入	单集群和三集群分别测试： create table t1(c1 int primary key auto_increment, c2 int); insert into t1(c2) values(1),(1);		

	update t1 set c1=3 where c1=1; insert into t1(c2) values(23333);		
测试结果	单集群：错误，出现 bug4（此版本不打算支持） 三集群：错误，出现 bug5（此版本不打算支持）		
编号	4	配置	190：RS、UPS、MS、CS 191：RS、UPS、MS、CS 192-198：RS、UPS、LMS + 6 * (MS + CS) 190、191 为备集群、193 为主集群
测试目的	自增支持的数据类型		

测试输入	单集群和三集群分别测试：
	create table t2(c1 float primary key auto_increment, c2 int);
	insert into t2(c2) values(1),(1),(1);
	replace into t2(c2) values(1);
	replace into t2 values(5,5);
	insert into t2(c2) values(1);
	
	create table t3(c1 double primary key auto_increment, c2 int);
	insert into t3(c2) values(1),(1),(1);
	replace into t3(c2) values(1);
	replace into t3 values(5,5);
	insert into t3(c2) values(1);		

测试结果	单集群：通过 高压自增单集群：通过（修复 bug7） 三集群：通过 高压自增三集群：通过 回归测试：通过		
编号	5	配置	190：RS、UPS、MS、CS 191：RS、UPS、MS、CS 192-198：RS、UPS、LMS + 6 * (MS + CS) 190、191 为备集群、193 为主集群
测试目的	分别删除一张含有自增表的全部数据和部分数据，自增计数起点		

测试输入	单集群和三集群分别测试：
	create table t1(c1 int primary key auto_increment, c2 int);
	insert into t1(c1,c2) values(1,1),(2,2),(3,3);
	delete from t1;
	insert into t1(c2) values(1),(1);
	delete from t1 where c1=5;
	insert into t1(c2) values(1);
	select * from t1;
	
	~~create table t1(c1 int primary key auto_increment, c2 int);~~		

	~~insert into t1(c1,c2) values(1,1),(2,2),(3,3);~~ ~~truncate table t1;~~ ~~insert into t1(c2) values(1),(1);~~ ~~select * from t1;~~		
测试结果	单集群：第一个通过（修复 bug6）；第二个 CEDAR 不支持 truncate 语法 高压自增单集群：第一个通过（修复 bug7），第二个语法 CEDAR 不支持 三集群：第一个通过，第二个语法 CEDAR 不支持 高压自增三集群：第一个通过，第二个语法 CEDAR 不支持 回归测试：第一个通过，第二个语法 CEDAR 不支持		
编号	6	配置	190：RS、UPS、MS、CS 191：RS、UPS、MS、CS 192-198：RS、UPS、LMS + 6 * (MS + CS) 190、191 为备集群、193 为主集群
测试目的	指定自增初始位置和自增步长		

测试输入	三集群和单集群：
	create table t1(c1 int primary key auto_increment, c2 int) auto_increment=2;
	set @@auto_increment_increment=3;
	insert into t1(c2) values(1),(1);
	select * from t1;		

测试结果	此版本不打算支持		
编号	7	配置	190：RS、UPS、MS、CS 191：RS、UPS、MS、CS 192-198：RS、UPS、LMS + 6 * (MS + CS) 190、191 为备集群、193 为主集群
测试目的	删除表结构后重建表，自增计数起点		

测试输入	三集群和单集群分别测试：
	create table t1(c1 int primary key auto_increment, c2 int);
	insert into t1(c2) values(1),(1),(1);
	drop table t1;
	create table t1(c1 int primary key auto_increment, c2 int);
	insert into t1(c2) values(2),(2);
	select * from t1;		

测试结果	单集群：通过，正确地重新从 1 开始自增 高压自增单集群：通过 三集群：通过 高压自增三集群：通过 回归测试：通过		
编号	8	配置	190：RS、UPS、MS、CS 191：RS、UPS、MS、CS

			192-198：RS、UPS、LMS + 6 * (MS + CS) 190、191 为备集群、193 为主集群
测试目的	自增表建立二级索引，插入 300 万条数据，计数是否正常，表中记录数目是否正确		

测试输入	三集群和单集群分别测试：
	create table t0(c1 int primary key auto_increment, c2 int);
	for(int i=0;i<3000000;i++)
	{
	replace into t0(c2) values(1);
	}
	select count(*) from t1;
	insert into t0(c2) values(2);
	select * from t0 where c1=3000001;		

测试结果	单集群：通过，计数正确，二级索引可用 三集群：通过，计数正确，二级索引可用 回归测试：通过（修复 bug11）		

边界 & 压力案例

编号	9	配置	190：RS、UPS、MS、CS 191：RS、UPS、MS、CS 192-198：RS、UPS、LMS + 6 * (MS + CS) 190、191 为备集群、193 为主集群
测试目的	自增达到 int 数据类型最大值时		

测试输入	单集群和三集群分别测试：
	create table t1(c1 int primary key auto_increment, c2 int);
	insert into t1(c1,c2) values(9223372036854775807, 1);
	insert into t1(c2) values(1);
	
	(2^63) – 1 = 9223372036854775807		

测试结果	单集群：Cedar 新插入值变成负数最大值，MySQL 直接报主键重复（需在模块功能说明书中告诉别人） 高压自增单集群：结果同单集群 三集群：结果同单集群 高压自增三集群：结果同单集群 回归测试：结果同单集群		
编号	10	配置	190：RS、UPS、MS、CS 191：RS、UPS、MS、CS 192-198：RS、UPS、LMS + 6 * (MS + CS) 190、191 为备集群、193 为主集群
测试目的	三集群，高压自增中，强制主 UPS 下线后，计数是否正常		
测试输入	create table t0(c1 int primary key auto_increment, c2 int);		

	for(int i=0;i<3000000;i++)
	{
	replace into t0(c2) values(1);
	}
	在 300 万条自增过程中强制主集群 UPS 下线，插入完成后继续以下操作：
	select count(*) from t0;
	insert into t0(c2) values(2);
	select * from t0 where c1=3000001;		

测试结果	通过，（修复 bug9） 回归测试：通过		
编号	11	配置	190：RS、UPS、MS、CS 191：RS、UPS、MS、CS 192-198：RS、UPS、LMS + 6 * (MS + CS) 190、191 为备集群、193 为主集群
测试目的	三集群，高压自增中，重新选主，计数是否正常		

测试输入	create table t1(c1 int primary key auto_increment, c2 int);
	for(int i=0;i<3000000;i++)
	{
	replace into t1(c2) values(1);
	}
	在 300 万条自增过程中重新选主，插入完成后继续以下操作：
	select count(*) from t1;
	insert into t1(c2) values(2);
	select * from t1 where c1=3000001;		

测试结果	通过 回归测试：通过		
编号	12	配置	190：RS、UPS、MS、CS 191：RS、UPS、MS、CS 192-198：RS、UPS、LMS + 6 * (MS + CS) 190、191 为备集群、193 为主集群
测试目的	自增表插入 300 万条数据，插入过程中多次每日合并，计数是否正常，表中记录数目是否正确		

测试输入	三集群和单集群分别测试：
	create table t1(c1 int primary key auto_increment, c2 int);
	for(int i=0; i < 3000000; i ++)
	{
	replace into t1(c2) values(1);
	}
	select count(*) from t1;		

	insert into t1(c2) values(2); select * from t1 where c1=3000001;
测试结果	单集群：通过，（修复 bug8） 三集群：通过 回归测试：通过

性能测试案例

<table>
<tr><td>编号</td><td>1</td><td>配置</td><td>190：RS、UPS、MS、CS
191：RS、UPS、MS、CS
192-198：RS、UPS、LMS + 6 * (MS + CS)
190、191 为备集群、193 为主集群</td></tr>
<tr><td>测试目的</td><td colspan="3">验证 CEDAR 自增和非自增性能差距是否合理</td></tr>
<tr><td>测试输入</td><td colspan="3">一个是 CEDAR 指定主键插入和利用自增功能插入的性能差，另一个是 MySQL 指定主键插入和自增插入的性能差

MySQL 和 CEDAR 单集群测试：
create table t0(c1 int primary key auto_increment, c2 int, c3 varchar(20));
for(int i=1; i<=3000000; i++)
{
replace into t0(c1, c2) values(i, i, ‘aaaaaaaaaaaaaaaa’);
}

create table t1(c1 int primary key auto_increment, c2 int, c3 varchar(20));
for(int i=1; i<=3000000; i++)
{
replace into t1(c2, c3) values(i, ‘aaaaaaaaaaaaaaaaaaaaaaaa’);
}</td></tr>
<tr><td>测试结果</td><td colspan="3">MySQL（4 次平均）
非主键自增
<table><tr><td>线程</td><td>TPS</td><td>QRS（ms）</td></tr><tr><td>100</td><td>31575</td><td>3.1848</td></tr><tr><td>90</td><td>31959</td><td>2.8261</td></tr><tr><td>80</td><td>31882</td><td>2.5194</td></tr><tr><td>70</td><td>31686</td><td>2.2210</td></tr></table>
主键自增
<table><tr><td>线程</td><td>TPS</td><td>QRS（ms）</td></tr><tr><td>100</td><td>25066</td><td>4.0016</td></tr><tr><td>90</td><td>26312</td><td>3.4111</td></tr><tr><td>80</td><td>27126</td><td>2.9697</td></tr><tr><td>70</td><td>29684</td><td>2.3703</td></tr></table>
MySQL 性能差别：使用自增 TPS 约为原来的 80%

CEDAR 三集群 7MS（4 次平均）
非主键自增</td></tr>
</table>

线程	TPS	QRS（ms）
100	107867	2.8374
90	101070	2.7221
80	94445	2.5767

主键自增（3MS）

线程	TPS	QRS（ms）
100	5400	54.0884
90	5321	49.7806
80	5292	42.9619

CEDAR 性能差别：使用自增 TPS 约为原来的 5%

主键自增（1MS）

线程	TPS	QRS（ms）
100	5000	19.9703
90	4723	18.8607

稳定性测试案例

编号	1	配置	194 主：RS,UPS,MS,CS 195 备: RS,UPS,MS,CS 196 备: RS, UPS, MS, CS
测试目的	三集群下建立自增表，一段时间内不停插入记录，系统是否正常		
测试输入	1. 20170319，11:02-19:19 插入约 1.2 亿条记录 2. 20170321，14:15-21:59 插入约 8500 万条记录		
测试结果	1. 系统正常提供服务，增删改查均正常 2. 系统正常提供服务，增删改查均正常		