

Cedar 项目组测试文档	
文档名称	可配置拓展 2MB 事务 Bug List
作者（测试人员）	张春熙
功能模块	可配置拓展 2MB 事务
开发人员	屈兴、朱涛
日期	20170116
负责老师	测试：张蓉；开发：钱卫宁

修订记录：

日期	修改描述
20161231	提交代码

## Bug List:

编号	1	时间	20170118	git 分支	Dev+2mb
配置	90 (ups、rs、lms)、92 (cs、ms)				
测试输入	<pre> delimiter // drop table if exists ptest// create table ptest(c1 int, c2 bool, c3 float, c4 double, c5 varchar(10000), c6 timestamp, primary key(c1))// drop procedure if exists ptest// create procedure ptest() begin     declare @c1 int default 1;     declare @c2 bool default false;     declare @c3 double default 10.0000000001;     declare @c4 double default 5.0001;     declare @c5 varchar(10000) default 'asd312dsd';     declare @c6 timestamp default '1993-03-07 10:08:24';     for @f1 in 1 to 509220 loop         insert into ptest values (@f1, @c2, @c3, @c4, @c5, @c6);     end loop; end// set @@session.ob_query_timeout=9000000000// set @@session.ob_tx_timeout = 9000000000// call ptest() /*+ NO_GROUP LONG_TRANS */ // select count (*) from ptest// </pre>				
错误现象	报错：ERROR 14 (HY001): OB-14: Memory overflow				
原因	For 语法问题				
解决情况	<p><b>已解决</b></p> <p>Memory overflow 的错误原因是，RWSession 会话使用了 session 工厂对象默认的 16MB 大小的缓冲区内存分配器页，无法继续执行。解决方式是设置内存分配器页的大小与日志缓冲区 log_buff_max_size 一致。</p> <p>解决时发现 For loop 还存在的一个问题：          存储过程 for...loop 语法会被成组优化，于是 for...loop 内的与 ups 有关的 sql 的物理计划会被发送到 ups，但是循环次数过多的时候，成组的 sql 的物理计划的大小超过 2mb，但是发送到 ups 的物理计划只有 2mb 略大。所以 ups 反序列化时出现错误，导致事务回滚。          暂时没有很好的解决方式，一种方式调整 ups 的存储物理计划的大小的缓冲区，但是这样做代价大，修改范围也比较大，而另一种调整成组的大小，也没有很好地方式计算每次应该成组多少条 SQL。</p> <p>所以，目前 for loop 语句只能 添加 hint，声明不成组执行。一条条的将 sql 发送给 ups 执行。</p>				

	mysql 登陆时 -c 使用 hint 存储过程执行时 call ptest() /*+ NO_GROUP */ 而一旦循环次数过多，事务执行时间过长，会导致回滚，因此需要设置超时时间避免事务回滚 set @@session.ob_query_timeout=9000000000; SET @@SESSION.ob_tx_timeout = 9000000000;				
编号	2	时间	20170118	git 分支	Dev+2mb
配置	90 (ups、rs、lms)、92 (cs、ms)				
测试输入	<pre> delimiter // drop table if exists ptest// create table ptest(c1 int, c5 varchar(10000), primary key(c1))// drop procedure if exists ptest// create procedure ptest() begin     declare @c1,@d1 int default 1;     declare @c5 varchar(10000) default 'aaaaaaaaaa';     declare @c6 timestamp default '1993-03-07 10:08:24';     while @c1 &lt; 500 do         replace into ptest values (@c1, @c5);         set @c1 = @c1 + 1;         set @d1 = 1;         while @d1 &lt; 100 do             replace into ptest values (@d1, @c5);             set @d1 = @d1 + 1;         end while;     end while; end// set @@session.ob_query_timeout=9000000000// call ptest()// select count(*) from ptest// </pre>				
错误现象	Ups 挂掉				
原因	<p>当前版本没有很好的计算 replace 相同主键在同一个事务时的序列化的大小，序列化出来的实际的大小与 get_serialize_size 返回的大小不一致，导致主动 kill ups 程序语句的执行</p> <p>没有计算正确的原因是 ob_mutator.cpp 中的 copy_cell_函数判断 cell 是否是另一行的方式依赖于主键是否是上一个</p> <p>解决方法是将原来是否是另一行的标志符号 is_row_changed 传递进去来设置 is_row_changed? CHANGED:NOCHANGED</p>				
解决情况	已解决				
编号	3	时间	20170118	git 分支	Dev+2mb
配置	90 (ups、rs、lms)、92 (cs、ms)				
测试输入	<pre> delimiter // drop table if exists ptest// create table ptest(c1 int, c5 varchar(10000), primary key(c1))// </pre>				

	<pre> drop procedure if exists ptest// create procedure ptest() begin     declare @c1,@d1 int default 1;     declare @c5 varchar(10000) default 'a';     declare @c6 timestamp default '1993-03-07 10:08:24';     while @c1 &lt; 500 do         replace into ptest values (@c1, @c5);         set @c1 = @c1 + 1;         set @d1 = 1;         while @d1 &lt; 1000 do             replace into ptest values (@d1, @c5);             set @d1 = @d1 + 1;         end while;     end while;  end// set @@session.ob_query_timeout=9000000000// set @@session.ob_tx_timeout = 9000000000// call ptest();// select count (*) from ptest// </pre>				
错误现象	事务回滚				
原因	可能是因为 while 次数过多 超时回滚				
解决情况	设置超时时间避免事务回滚 <pre> set @@session.ob_query_timeout=9000000000; SET @@SESSION.ob_tx_timeout = 9000000000; </pre>				
编号	4	时间	20170118	git 分支	Dev+2mb
配置	90 (ups、rs、lms)、92 (cs、ms)				
测试输入	<pre> drop table if exists ptest// create table ptest(c1 int, c2 bool, c3 float, c4 double, c5 varchar(10000), c6 timestamp, primary key(c1))// drop procedure if exists ptest// create procedure ptest() begin     declare @c1,@d1 int default 1;     declare @c2 bool default false;     declare @c3 float default 100.000;     declare @c4 double default 5.0001;     declare @c5 varchar(10000) default 'asdasdsd';     declare @c6 timestamp default '1993-03-07 10:08:24';     replace into ptest values (@c1, @c2, @c3, @c4, @c5, @c6);     while(@c1&lt;1000) do         set @c1 = @c1 + 1;     end while; end// </pre>				

	<pre> replace into ptest values (@c1, @c2, @c3, @c4, @c5, @c6); if(@c1%2 = 1) then     replace into ptest values (@c1, @c2, @c3, @c4, 'sds', @c6); end if; if(@c1 = 100) then     while(@c3 &gt; 0) do         set @d1 = @d1 - 1;         replace into ptest values (@d1, @c2, @c3, @c4, 'xiugai', @c6);         set @c3 = @c3 - 1;     end while; end if; end while; end// set @@session.ob_query_timeout=9000000000// call ptest()// select count(*) from ptest// </pre>				
错误现象	Ups 挂掉				
原因	<a href="#">同 bug2</a>				
解决情况					
编号	5	时间	20170118	git 分支	Dev+2mb
配置	90 (ups、rs、lms)、92 (cs、ms)				
测试输入	<pre> drop table if exists ptest; create table ptest(c1 int, c2 bool, c3 float, c4 double, c5 varchar(10000), c6 timestamp, primary key(c1)); insert into ptest values (@d1, @c2, @c3, @c4, @c5, @c6); </pre>				
错误现象	Ms 挂掉				
原因	<a href="#">Ob 原来的 bug</a> ，由于出错的返回值代码被后来的覆盖，导致执行逻辑错误				
解决情况	<a href="#">已解决</a>				
编号	6	时间	20170118	git 分支	Dev+2mb
配置	三集群：90 (ups、rs、lms)、92 (cs、ms)、93 (ups、rs、cs、ms)				
测试输入	同 bug1				
错误现象	<p>日志报错，集群不可用</p> <pre> ERROR get_log (ob_log_buffer.cpp:251) [140662005540608] get_log(start_id=713, copy_count=2097152, read_count=0): NOT find align pos WARN      trim_log_buffer  (ob_ups_log_utils.cpp:1054)  [140662005540608] trim_log_buffer(offset=797184, align=9, len=2097152, end_pos=0): not found aligned pos WARN      trim_log_buffer  (ob_ups_log_utils.cpp:1054)  [140662026520320] trim_log_buffer(offset=347136, align=9, len=2097152, end_pos=0): not found aligned pos ERROR get_log (ob_log_buffer.cpp:251) [140662026520320] get_log(start_id=713, copy_count=2097152, read_count=0): NOT find align </pre>				

原因					
解决情况	已解决				
编号	7	时间	20170118	git 分支	Dev+2mb
配置	三集群：90（ups、rs、lms）、92（cs、ms）、93（ups、rs、cs、ms）				
测试输入	1、设置 32M 包配置后，测试一组功能保证 32M 包在无压力负载时可用。 2、一个客户端对数据库进行压力测试，500 个线程均匀的连接到 5ms 上。一个客户端连接到数据库，执行 32M 包功能测试，测试案例在>30MB,<32MB，此时服务器的内存消耗巨快，跑不出来结果，一段时间后挂掉 ups，之后重启 ups 后集群不可用。				
错误现象					
原因					
解决情况	已解决				