**OceanBase系统自举流程**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **编号** | **文档版本** | **修订章节** | **修订原因** | **修订日期** | **修订人** |
| **1** | 0.1 |  | 新建文档 | 2012/4/20 | 竹翁 |
| **2** |  |  |  |  |  |
| **3** |  |  |  |  |  |
| **4** |  |  |  |  |  |
| **5** |  |  |  |  |  |
| **6** |  |  |  |  |  |

OceanBase的schema和root table改为内部表后，需要为系统自举设计一个特殊的流程。

# 新建table流程

为了支持下面要描述的自举过程，以及支持新的内部表实现，我们需要修改新建表格的流程。为了对比，先说明一下旧的通过switch\_schema建表的流程：

1. RS装载新的schema文件，选出需要新建的表格
2. RS从UPS取得last\_frozen\_version
3. RS从CS列表中选择若干个CS，发送新建空tablet命令，命令包含last\_frozen\_version
4. 只要有一个CS新建tablet成功，则认为建表成功

以上流程的最大问题是，很难保证原子性。

新的建表流程：

1. RS从UPS取得last\_frozen\_version（last\_frozen\_version能否放到内部表all\_sys\_stat中？）
2. RS给UPS发送修改请求，修改first\_tablet\_entry, all\_all\_column, all\_all\_join等完成新建表格。新建的table的schema中有一个特殊属性表示新建表格时刻的frozen\_version

以上两步就是新建表格的全部流程。此时，新表还没有在CS上由对应的空sstable。

为了能够让MS能立即读出这个表的内容，MS在读数据时需要加一个判断：

1. MS在读取数据时，如果schema中显示的该表新建时frozen\_version等于当前系统的frozen\_version，则不需要去查找和读取CS端的SStable上数据。

那么，何时为这些新表新建空sstable呢：

1. RS在做复本复制rereplication检查是，如果发现一个某个表的create\_frozen\_version等于当前系统frozen\_version，并且这个表的tablet还没有在任何CS上新建，则选取若干CS并发送新建命令；
2. RS收到UPS汇报的冻结指令时，也执行上述检查。
3. 对\_\_first\_tablet\_entry, \_\_all\_all\_column, \_\_all\_all\_join三个表不执行上述检查；见下一节的自举流程。

# 自举流程

## 2.1触发时机

RS端持久化存储了first\_tablet\_entry的存储位置。RS启动后，如果这个文件不存在，则启动自举流程。此时，RS只提供机器MS/CS/UPS的管理服务。

## 2.2 流程

所有server都是通过ObSchemaService类读取schema信息的，而这个类已经硬编码了\_\_first\_tablet\_entry, \_\_all\_all\_column, \_\_all\_all\_join三个表的schema信息。

1. RS等待至少有一个CS已经注册
2. RS选定若干CS，发送命令在上面新建\_\_first\_tablet\_entry, \_\_all\_all\_column, \_\_all\_all\_join三个表的空表。
3. RS给UPS发送mutate消息，新建\_\_all\_sys\_param, \_\_all\_sys\_stat两个表。
4. RS选定若干CS，发送命令在上面新建\_\_all\_sys\_param, \_\_all\_sys\_stat两个表的空表。
5. RS给UPS发送mutate消息，写入\_\_all\_sys\_param, \_\_all\_sys\_stat的初始内容。
6. 以上步骤都成功，RS把\_\_first\_tablet\_entry的存储位置等信息持久化。

# 附录

自举完成后，各个系统表的初始状态如下(#开头的行注释了行的schema)。

## \_\_first\_tablet\_entry

# schema := table\_name(16),table\_id,table\_type,meta\_tname,meta\_tid,load\_type,table\_def\_type,rowkey\_column\_num,column\_num,max\_used\_column\_id,replica\_num(26),...

\_\_all\_all\_column,2,3,\_\_first\_tablet\_entry,1,1,1,2,13,28,replica1\_ipv4,replica1\_ipv6\_high,replica1\_ipv6\_low,replica1\_port,1,replica1\_row\_count,replica1\_size,replica1\_checksum,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

\_\_all\_join\_info,3,3,\_\_first\_tablet\_entry,1,1,1,4,8,23,replica1\_ipv4,replica1\_ipv6\_high,replica1\_ipv6\_low,replica1\_port,1,replica1\_row\_count,replica1\_size,replica1\_checksum,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

\_\_all\_sys\_param,4,3,\_\_first\_tablet\_entry,1,1,1,9,12,27,replica1\_ipv4,replica1\_ipv6\_high,replica1\_ipv6\_low,replica1\_port,1,replica1\_row\_count,replica1\_size,replica1\_checksum,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

\_\_all\_sys\_stat,5,3,\_\_first\_tablet\_entry,1,1,1,10,14,29,replica1\_ipv4,replica1\_ipv6\_high,replica1\_ipv6\_low,replica1\_port,1,replica1\_row\_count,replica1\_size,replica1\_checksum,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

## \_\_all\_all\_column

# schema := table\_name(16),table\_id,column\_name,column\_id,column\_group\_id,rowkey\_id,join\_table\_id,join\_column\_id,data\_type,data\_length,data\_precision,data\_scale,nullable

\_\_all\_sys\_param,4,name,16,0,1,0,0,6,256,0,0,0

\_\_all\_sys\_param,4,cluster\_role,17,0,2,0,0,1,8,0,0,0

\_\_all\_sys\_param,4,cluster\_id,18,0,3,0,0,1,8,0,0,0

\_\_all\_sys\_param,4,server\_type,19,0,4,0,0,1,8,0,0,0

\_\_all\_sys\_param,4,server\_role,20,0,5,0,0,1,8,0,0,0

\_\_all\_sys\_param,4,server\_ipv4,21,0,6,0,0,1,8,0,0,0

\_\_all\_sys\_param,4,server\_ipv6\_high,22,0,7,0,0,1,8,0,0,0

\_\_all\_sys\_param,4,server\_ipv6\_low,23,0,8,0,0,1,8,0,0,0

\_\_all\_sys\_param,4,server\_ip\_port,24,0,9,0,0,1,8,0,0,0

\_\_all\_sys\_param,4,data\_type,25,0,0,0,0,1,8,0,0,0

\_\_all\_sys\_param,4,value,26,0,0,0,0,6,256,0,0,0

\_\_all\_sys\_param,4,info,27,0,0,0,0,6,256,0,0,0

\_\_all\_sys\_stat,5,name,16,0,1,0,0,6,256,0,0,0

\_\_all\_sys\_stat,5,cluster\_role,17,0,2,0,0,1,8,0,0,0

\_\_all\_sys\_stat,5,cluster\_id,18,0,3,0,0,1,8,0,0,0

\_\_all\_sys\_stat,5,server\_type,19,0,4,0,0,1,8,0,0,0

\_\_all\_sys\_stat,5,server\_role,20,0,5,0,0,1,8,0,0,0

\_\_all\_sys\_stat,5,server\_ipv4,21,0,6,0,0,1,8,0,0,0

\_\_all\_sys\_stat,5,server\_ipv6\_high,22,0,7,0,0,1,8,0,0,0

\_\_all\_sys\_stat,5,server\_ipv6\_low,23,0,8,0,0,1,8,0,0,0

\_\_all\_sys\_stat,5,server\_ip\_port,24,0,9,0,0,1,8,0,0,0

\_\_all\_sys\_stat,5,table\_id,25,0,10,0,0,1,8,0,0,0

\_\_all\_sys\_stat,5,data\_type,26,0,0,0,0,1,8,0,0,0

\_\_all\_sys\_stat,5,value1,27,0,0,0,0,1,8,0,0,0

\_\_all\_sys\_stat,5,value2,28,0,0,0,0,6,256,0,0,0

\_\_all\_sys\_stat,5,info,29,0,0,0,0,6,256,0,0,0

## \_\_all\_all\_join（空）

# schema := left\_table\_name,left\_table\_id,left\_column\_name,left\_column\_id,right\_table\_name,right\_table\_id,right\_column\_name,right\_column\_id

## \_\_all\_sys\_param

# schema := name,cluster\_role,cluster\_id,server\_type,server\_role,server\_ipv4,server\_ipv6\_high,server\_ipv6\_low,server\_ip\_port,data\_type,value,info

ob\_app\_name,0,0,0,0,0,0,0,0,6,my app name,oceanbase app name

## \_\_all\_sys\_stat

# schema := name,cluster\_role,cluster\_id,server\_type,server\_role,server\_ipv4,server\_ipv6\_high,server\_ipv6\_low,server\_ip\_port,table\_id,data\_type,value1,value2,info

max\_used\_table\_id,0,0,0,0,0,0,0,0,0,1,1000,,max used table id