# MGR监控

设计的表一共有五张：

replication\_group\_members、replication\_group\_member\_stats、replication\_connection\_status、replication\_applier\_status、threads

-- -------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

使用Perfomance Schema表来监视组复制组复制添加以下表：

performance\_schema.replication\_group\_member\_stats

performance\_schema.replication\_group\_members

这些Perfomance Schema复制表还显示有关组复制的信息：

performance\_schema.replication\_connection\_status

performance\_schema.replication\_applier\_status

由组复制插件创建的复制通道命名为：

group\_replication\_recovery - 此通道用于与分布式恢复阶段相关的复制更改。

group\_replication\_applier - 此通道用于组中的传入更改。这是用于直接从组中应用交易的渠道。

以下部分描述了每个表中的信息：

**replication\_group\_members**

存储这组内的所有成员的基本信息，从任何一个尘缘上都能查询的到这些信息。

|  |  |
| --- | --- |
| **字段** | **说明** |
| CHANNEL\_NAME | Group Replication 执行Binlog Event的通到，这个字段的值是‘group\_replication\_applier’ |
| MEMBER\_ID | 成员的UUID，和每个成员上MySQL全局变量server\_uuid的内容一样 |
| MEMBER\_HOST | 成员的hostname，和每个成员上MySQL全局变量hostname或report\_host的内容相同 |
| MEMBER\_PORT | 成员的port，和每个成员上MySQL全局变量port或report\_port的内容相同 |
| MEMBER\_STATE | 成员的状态 |
|  |  |

成员的状态有以下五种：

* OFFLINE：当MySQL服务器的Group Replication插件没有启动时，状态为OFFLINE。
* RECOVERING：当MySQL服务器的Group Replication插件没有启动后，会首先设置成RECOVERING状态，开始复制加入前的数据。
* ONLINE：当RECOVER过程完成后，状态设置为ONLINE，开始对外提供服务。
* ERROR：当本地成员发生错误，Group Replication无法正常进行时，当前成员的状态会变成错误。
* UNREACHABLE：当网络故障或其他成员宕机时，其它成员的状态会被设置为UNREACHABLE。

**replication\_group\_member\_stats**

存储着本地成员的详细信息，每个成员只能查询到自己的详细信息。

|  |  |
| --- | --- |
| **字段** | **说明** |
| CHANNEL\_NAME | Group Replication 执行Binlog Event的通到，这个字段的值是‘group\_replication\_applier’ |
| VIEW\_ID | 组视图的ID |
| MEMBER\_ID | 成员的UUID，和每个成员上MySQL全局变量server\_uuid的内容一样 |
| COUNT\_TRANSACTIONS\_IN\_QUEUE | 队列中等待做全局事务认证的事务数量 |
| COUNT\_TRANSACTIONS\_CHECKED | 做了全局事务认证的事务总数量，从加入组开始累计 |
| COUNT\_CONFLICTS\_DETECTED | 全局事务认证时，有冲突的事务的总数量 |
| COUNT\_TRANSACTIONS\_ROWS\_VALIDATING | 冲突检测数据库的记录总行数 |
| TRANSACTIONS\_COMMITTED\_ALL\_MEMBERS | 在所有成员上已经执行了的事务的GTID集合。相当于所有成员的“gtid\_executed”的交集。不是实时的，每隔一段时间更新一次 |
| LAST\_CONFLICT\_FREE\_TRANSACTION | 最后一个没有冲突的事务的GTID |
|  |  |

**replication\_connection\_status**

当MySQL服务器加入一个组之后，首先要通过异步复制通道（Channel）group\_replication\_recovery把加入组之前组内产生的数据复制过来，这个通道的状态信息和其他异步复制的通道一样可以通过该表进行监控。Group Replication的通道不会在SHOW SLAVE STATUS的结果中显示。

**replication\_applier\_status**

Group Replication通过group\_replication\_applier通道来执行Binlog Event。group\_replication\_applier的状态信息和其它异步复制通道一样可以通过该表进行查询。

## 相关命令：

1.当前主节点的uuid  
select \* from performance\_schema.global\_status where variable\_name = 'group\_replication\_primary\_member';

2.查看节点状态  
SELECT \* FROM performance\_schema.replication\_group\_members;

3.通过以下SQL来直接判断当前节点是否为主节点，得到1表示主节点，0表示不是主节点:

SELECT IF((SELECT @@server\_uuid) = (SELECT VARIABLE\_VALUE FROM performance\_schema.global\_status WHERE VARIABLE\_NAME= 'group\_replication\_primary\_member'), 1, 0) as is\_primary\_node;

4.通过直接查看read\_only变量来判断节点是否为主节点:  
select @@read\_only;