

# 【MySQL 5.7.17】从主从复制到Group Replication

原创：刘伟 数据和云 2016-12-15

时值双十二之际，MySQL官方献上了大礼，Group Replication(后文简称GR)终于正式宣布GA，组合在MySQL 5.7.17版本内部发布出来。

MySQL 5.7.17有很多修正，但GR的发布，却是最值得说的一个事情。



在很久之前，MySQL只是一个采用statement格式作为复制格式，纯异步化复制MyISAM作为存储引擎的，可以运行SQL语句的文件管理器。

## ◆ InnoDB的改进

在InnoDB出现之前，MySQL在数据安全以及性能上是很难保证的：

MyISAM的读写表级锁

宕机不安全

著名的永远跑不完的repair table

这些问题都说明MySQL当时只能作为数据库的一个补充角色，而不能作为任何有持久化，安全要求的数据库需求的用品。

InnoDB为MySQL带来了redo，undo，事务，行级锁等关系数据库DBA这些熟悉的概念，也是从InnoDB开始，MySQL正式作为生产业务数据库进入人们的视线。

## ◆ 主从同步

MySQL若作为一个单机数据库时候，需要解决的问题有两个：

1、ACID问题

2、主从同步

很显然，InnoDB已经解决了第一个问题，接下来是第二个问题。

Statement格式的复制，虽然保证主从运行的SQL语句一致，但这并不能保证主从数据一致，尤其是一些操作系统依赖的函数调用或者一些特殊环境依赖的使用，于是，在此之上，MySQL出现了基于InnoDB的row格式复制，ROW格式可以保证数据的变更记录到日志，为保证主从数据一致性给出了巨大的贡献。

MySQL 5.1版本可以说是一个非常重要的版本，这个版本发布于2008年，适逢新时代互联网大潮发展，MySQL开始被广泛使用于互联网，其读写分离，从库基本上线性扩展读能力的方式，很快普及到整个行业。

但是，**row格式复制提供的是异步复制**。由于互联网技术发展对业务连续性的要求，主库宕机之后，及时实例可以恢复，大多数时候也会使用从库重新构建主从后直接提供服务（典型的例子为MHA），这种操作的背后，由于MySQL的异步复制，即使是在最好情况下，仍然会有临界点事务丢失的问题。

### 一些尝试性改进

---

Google为了解决这个问题引入了半同步技术，作为自己的独立发布版本在内部使用，之后，sun被Oracle收购，在随后发布的MySQL 5.5版本中，半同步被吸收入官方，为MySQL主从复制的数据安全，画上了一个不太完整的句号。

对于半同步技术，从MySQL 5.5到5.6，再到5.7，每个版本都会对此做做修正，半同步技术也一直在不断完善和强大的过程中，在MySQL内部，也逐渐演变出并行复制的方案。一般我们会建议使用MySQL 5.7的最新版本，保障数据安全。

盖技术的更新除了在数据安全上有了更大的保障之外，也让主从复制的另外一个问题-**SQL**线程得到了相当大的缓解。

其实主从复制-->半同步-->并行复制这条路本身，是可以一直走下去的，但是，也会有人问，主从复制是否是唯一的一条路？



还有一个方案，听说过的人应该不在少数，但是用过的人并不多：**MySQL NDB Cluster**。其工作原理是：

把MySQL作为SQL解析以及命令转发的Proxy，后端用NDB Cluster提供的分布式数据库提供服务

这个思路本身没有问题，可惜生不逢时，NDB出现得太早，管理的思路 and 使用的思路，都大异于传统的数据库，相比较当代的各路NoSQL的牛鬼蛇神，NDB足称可靠，但作为企业级产品上，运维以及开发的学习

代价太大；而用于互联网场景，又不足以与各路NoSQL拼比专项，到目前为止，仍然没有被作为主要使用方案。



## 官方之外，Galera Cluster另辟蹊径

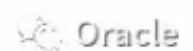
官方的道路一度到此为止，但MySQL作为开源产品，社区里面永远不乏高手，在官方之外，走出了另外一条路，**Galera Cluster**。

Galera Cluster的思路，是在尽量不改变MySQL的运维思路的基础上，保障数据库的安全。最终出现的，是一个乍一看比较奇怪的东西：**Galera是多节点可写的，节点之间share nothing，每个节点都保存当前数据库所有数据，commit发生在单个节点，节点间锁冲突延后到commit阶段处理的集群。**

很多人，包括我在内，认为Galera这种方式才是一个“真正的集群”，节点之间通过分布式协议沟通，节点失败自动踢出，节点加入自动同步，这些才是一个集群应该干，并且应该干到的事情。而传统的主从复制方式，无论如何美化描述，也都需要诸多外围脚本支持才能实现这些功能，并不是一个“真正的集群”。

从理论上讲，虽然有一定的限制条件，但**Galera**所描绘的**MySQL**集群也已经足够漂亮。但是，为了做到这些功能，**Galera对MySQL数据库本身做了不少修改，这点让很多有“官方”洁癖的人，比较担心Galera的引入对MySQL稳定性造成的影响**，从如今的趋势来看，Galera方案几乎与NDB方案一样的结局，最后沦为漂亮的花瓶。

虽然前面说Galera听起来多么美好，但MySQL官方由于种种缘由，没有合并Galera的工作进入官方版本。



但第三方的开发版本则没有这么多忌讳，MySQL世界的两个发行版-Percona以及MariaDB很快结合Galera方案

Percona给出了的是Percona Xtra Cluster（PXC）方案，MariaDB在新版本（现在已经是稳定版本）直接原生组合Galera进去，Galera的问题，由Percona与MariaDB分别按照自己的思路处理解决，为人们的使用创造方便。

Percona本身就是最大的开源数据库服务商，MariaDB也有基金会与商业公司支持，这两个公司的方案，在经历了一段时间的试水期之后，很快被业界接受。

国外姑且不论，国内的情况，Percona方案从去哪儿网开始，被广泛使用于互联网类行业，对高可用以及数据丢失敏感的业务。而另外一个MariaDB的方案，则在传统行业的“去IOE”运动中扮演着重要角色。

方案本身的可靠性比较是不必疑虑的，但使用场景的结果如此，MariaDB的用户更多看重的，应该还是MariaDB背后完整的开源基因吧。

MySQL官方呢，在这个潮流中，就只是看着吗？

当然不是，既然没有吸收Galera进入自身，那么剩下的事情，就是自己开发了。



在前段时间发布的整个MySQL InnoDBCluster计划中，MySQL官方的野心很大，包括多主集群，读写分离，读横行扩展，写横行扩展等诸多组件。对，里面提到的，多主集群，就是MySQL原生的，与Galera类似的，“真正的集群”方案。也是整个计划里面，目前第一个可用的。

从规划时间上看，在非常早的时间，GR就已经作为规划方案开始编写，初始于MySQL Lab，最终合并到官方分支宣布GA，历经了多年时间开发，为用户以及社区给出了MySQL自己的多主方案。

本质上，GR是一个与Galera方案类似的多主集群方案，原理上，都是分布式协议沟通，commit阶段处理节点间锁冲突等等。

在Galera方案已经大行其道的现在，GR还有什么优势或者意义呢？

- 最直观的第一点，就是这个是MySQL的官方方案，也就是说，用户可以不必要忌讳Percona以及MariaDB的“非官方感”而直接使用到更好（根据特定的benchmark）的多主集群服务。可以直接享受到多主复制，多线程复制等官方已提供的功能而不必等纠结第三方的可靠性以及学习成本。

- 第二点，Galera由于实现方式限制，只能用于linux平台，但GR是可以用于win以及mac（BSD）平台的，这点无论是对于技术本身的学习，还是小企业环境的部署，都有足够的好处。对运维技能的需求，也在一定程度上降低了不少。
- 第三点，Galera的实现毕竟是外加的组件。比如，由于引入的gcache作为事务的同步缓存，造成主机资源的耗费，而GR方案则直接使用row格式的binlog做这个工作，降低了主机压力。而且，一旦待同步数据库的延迟超过gcache的限制，就会导致数据库重传（SST），GR通过binlog的复用，直接采用传统的数据库备份恢复方式就可以构建节点开始同步，这点上比Galera的实现更适合生产环境。

因此从长期考虑上看，**GR**的实现会是更好的选择！

然而，目前阶段，**GR**还有些问题需要逐步解决或者让人们排除顾虑。

第一点，生产环境的使用。无论是PXC还是MariaDB方案，都已经存在生产环境运行多年的案例，其稳定性，安全性，乃至运行中遇到的种种问题的解决方案手段，都有成熟，众多的积累案例，GR则是刚刚GA，并且只提供给MySQL5.7.17版本（估计后续版本都会集成），在MySQL 5.7开始进入生产环境部署的时候，如果一并纳入GR的部署，带来的运维以及使用问题相信不在少数，这些问题如何解决，最佳实践是什么，这是一个需要持续长时间维护的事情。

第二点，工具支持。到目前为止，MySQL主要使用的运维工具，相当多的程序来源是Percona公司，Percona公司对MySQL官方的态度一直是积极跟进，但是，在已经有PXC方案的现状下，Percona公司有多大兴趣以及人力维护GR相关的工具体系是一个目前存疑的问题。

第三点，使用培训。GR方案作为一个更完美的高可用以及数据安全方案，其实际使用中，DBA以及开发，必须对GR的种种限制有足够多的了解，才能在实际程序开发以及运维中，做到最合适的处理，这一点势必需要社区与官方的大力推进推动才可以。

很幸运，我们可以生活在这个时代，可以看着MySQL从一个“可以跑SQL的文件工具”，逐渐走向为一个高可用高安全的关系数据库系统。在这个过程中，我们未必有能力或者精力去构建栋梁，但添砖加瓦，雕梁画壁的时间，却是谁都有的。

关于**GR**更多知识：

[MySQL Group Replication 学习笔记](#)

想学习更多MySQL的知识，与专家面对面交流，请扫码加群：



## 云和恩墨大讲堂(M)



### 近期文章

[MySQL Group Replication 学习笔记](#)

[利用硬链接和truncate降低drop table对线上环境的影响](#)

[以12c Identity类型示范自我探索式学习方法](#)

[从执行计划洞察ORACLE优化器的“小聪明”](#)

[Oracle安全比特币勒索问题揭秘和防范](#)

[针对Sharding DB的单点故障，合理构建HA架构](#)

### 资源下载

关注本微信（OraNews）回复关键字获取

**CodeSet**, 《Oracle性能优化与诊断案例精选》代码;

**2016OTC**, 第六届Oracle技术嘉年华PPT;

**2016DTCC**, 2016数据库大会PPT;

**DBALife**, "DBA的一天"精品海报大图;

**12cArch**, "Oracle 12c体系结构"精品海报;

**DBA01**, 《Oracle DBA手记》第一本下载;

**YunHe**, "云和恩墨大讲堂"案例文档下载;