

平安好医生数据库升级 8.0 项目总结

作者	茹憶
时间	2022-01-01

项目背景

2021 年,数据库升级作为重点项目推荐,全年完成了线上 1000 多个应用的数据库中间件升级,400 多个生产数据库升级到了 8.0。整个升级方案和流程经过了大量的验证和测试,整个升级未影响线上业务。在此将升级的一些流程和经验进行整理分享。

升级准备阶段

2021 年整个上半年,数据库升级项目基于处于准备阶段,主要包括 DBA 相关运维工具的改造开发和 MySQL 上下游业务组件改造开发两个部分。

第一部分工作是 DBA 相关运维和工具的改造开发,包括 MySQL8.0 的小版本选项、性能压测、升级流程验证、自动化部署平台适配、自动化升级工具开发、备份和监控工具改造、高可用工具的调研测试等等。

第二部分工作是 MySQL 上下游业务组件适配 MySQL8.0 的开发和验证工作。MySQL 使用的上游主要是开发同学,首先开发同学使用了我们的 Phenix 数据库中间件(底层是 druid)连接数据库,如果要升级数据库到 8.0,就需要确保 phenix 支持 8.0 的版本,另外还有 MySQL 的下游组件,比如 Binlog 订阅组件 canal/roma,大数据离线抽数组件 datax,都需要在 MySQL8.0 环境下进行升级开发和验证工作。

整体升级阶段

2021年6月份的时候，升级准备的阶段的工作已经全部完成，升级准备阶段的工作主要是内部团队工作，再加上中间件团队和大数据团队一起合作，基本没什么太大压力，因为毕竟都是改造开发和验证阶段，没有什么很大的风险。但是到真正准备升级数据库的时候，压力还是比较大。首先是风险问题，MySQL5.6升级到MySQL8.0，中间跨越了MySQL5.7版本，本身升级就可能因为版本变更带来很大的业务风险，比如SQL不兼容，数据库中间件phenix，下游canal、roma等组件未升级或者存在未知的问题都可能引起业务故障。其次是我们目前有1600多个数据库，超过1000多个应用使用数据库，升级时，需要将这个服务器上的所有实例和数据库全部升级掉，我们的数据库部署环境是一个服务器多实例部署，每个实例包含多个数据库，这些数据库都有不同的业务方应用来访问，简单来说每个升级的服务器上都包含多个实例和许多数据库，这些数据库都的访问者可能来自不同的业务团队，如果要升级这个服务器，就要确保这台服务器上面涉及数据库的业务全部升级数据库中间件，否则数据库升级后，如果有数据库中间件没有升级，那将导致业务访问异常。最后涉及1000多个应用升级数据库中间件phenix，几乎所有的业务方都要配合我们来进行升级和验证，如何推进升级，让数百开发完全配合我们完成升级也是需要考虑的问题。

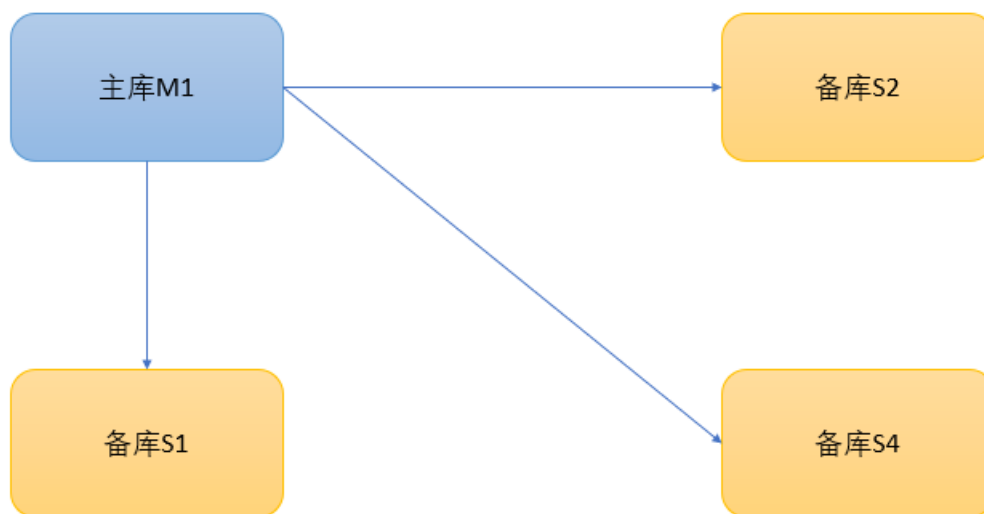
针对上述风险和问题，我定制了如下策略方案，将整个升级工作划分为4个阶段，定制了一套升级计划，通过流程将风险降低到最低，并邀请公司架构委员会负责人进行项目评审，评审之后每周会进行项目进度的汇总和跟进。通过项目管理和周报的形式促使业务方配合我们完成数据库中间件的升级和其他升级事项配合。

整体升级方案四个阶段如下：

- 1) 中间件业务升级：中间件业务风险较低，这个阶段先把中间件团队的应用 phenix 升级到新版本，数据库暂时不升级，持续观察验证新版本中间件 phenix 使用 MySQL5.6 数据库是否稳定。
- 2) 线上业务 20%打样升级：这个阶段我们选择了大概 20% 的线上数据库和应用，大概 200 多个数据库和应用，这个阶段我们要求每个应用都需要在测试环境升级 phenix 数据库中间件，同时在测试环境分别验证新版本 phenix 在 MySQL5.6 和 MySQL8.0 下业务是否正常，验证完成后将 phenix 发布上线，DBA 升级对应的线上数据库到 8.0。
- 3) 剩余 80%业务全部升级 phenix：这个阶段我们要求所有业务方升级完成 phenix，因为业务验证工作量比较大，前 20%的应用已经验证完成，phenix 升级未发现风险，所以这个阶段我们不要求开发同学做全量业务验证，只需要升级 Phenix 观察无问题后发布到线上环境。
- 4) 数据库逐步升级：这个阶段开发已经完成了测试和线上所有 phenix 版本的升级，phenix 支持连接 MySQL5.6 和 MySQL8.0 两个版本，然后 DBA 将测试环境所有数据库升级到 MySQL8.0 进行观察和收集问题，没有问题后逐步将线上数据库升级到 MySQL8.0 版本。

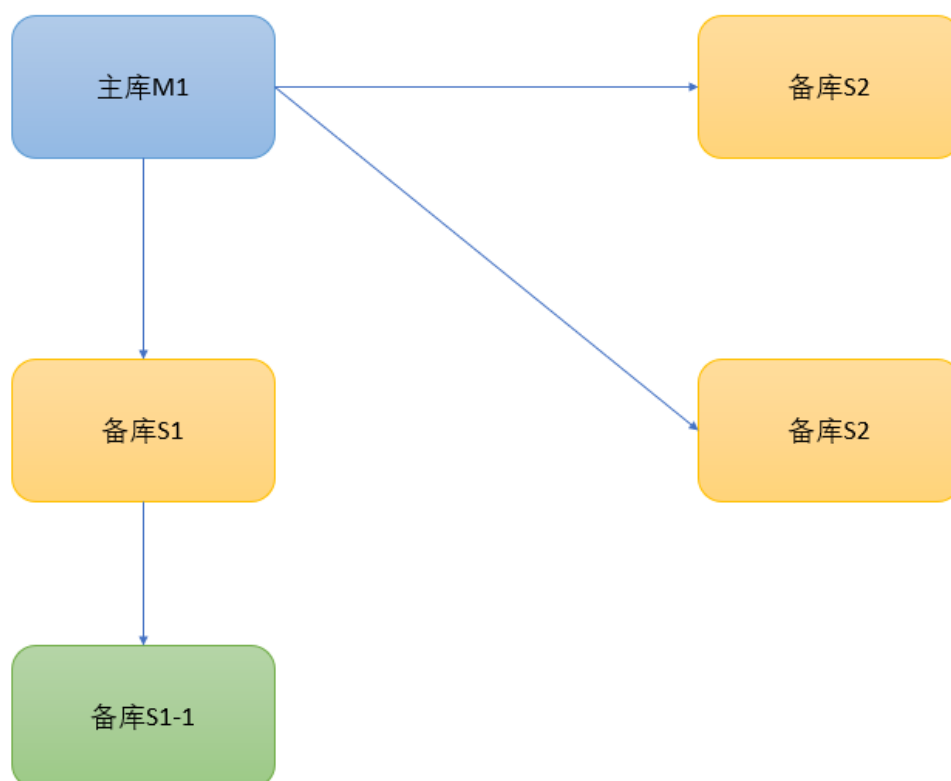
升级流程步骤

下面介绍下数据库升级的整个流程。我们目前 MySQL5.6 的架构如下，使用 MySQL Replication 架构，1 个主库，3 个备库，其中备库 S2 和 S4 在异地机房，都为 5.6 版本。

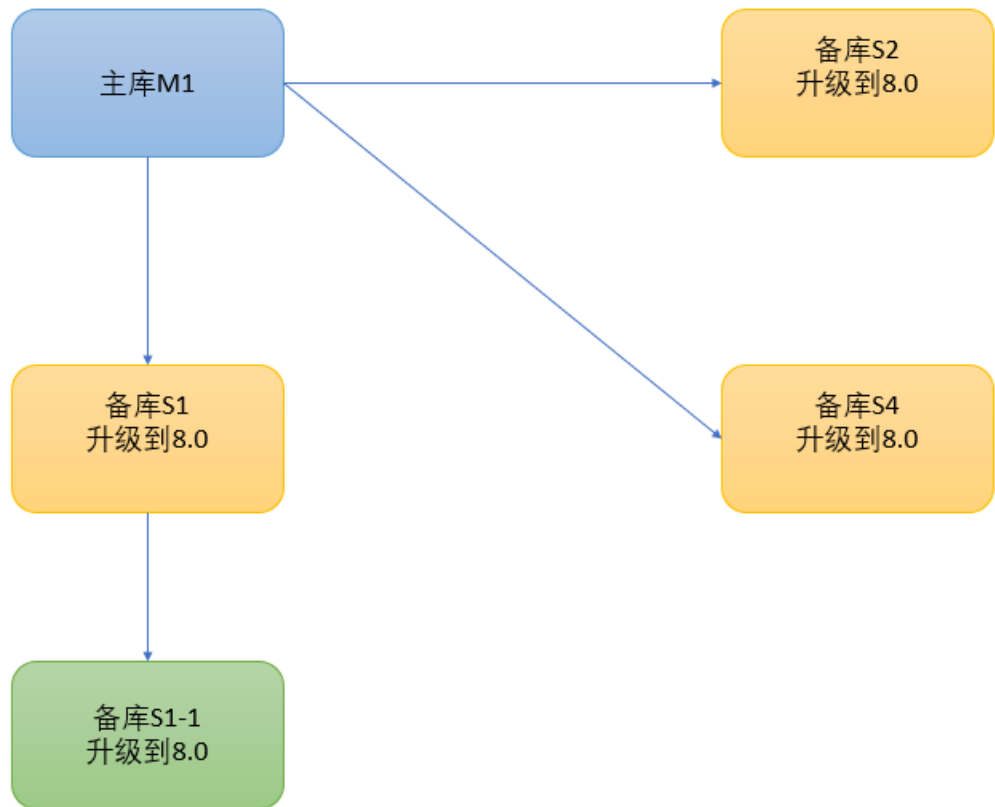


升级到 MySQL8.0 流程如下：

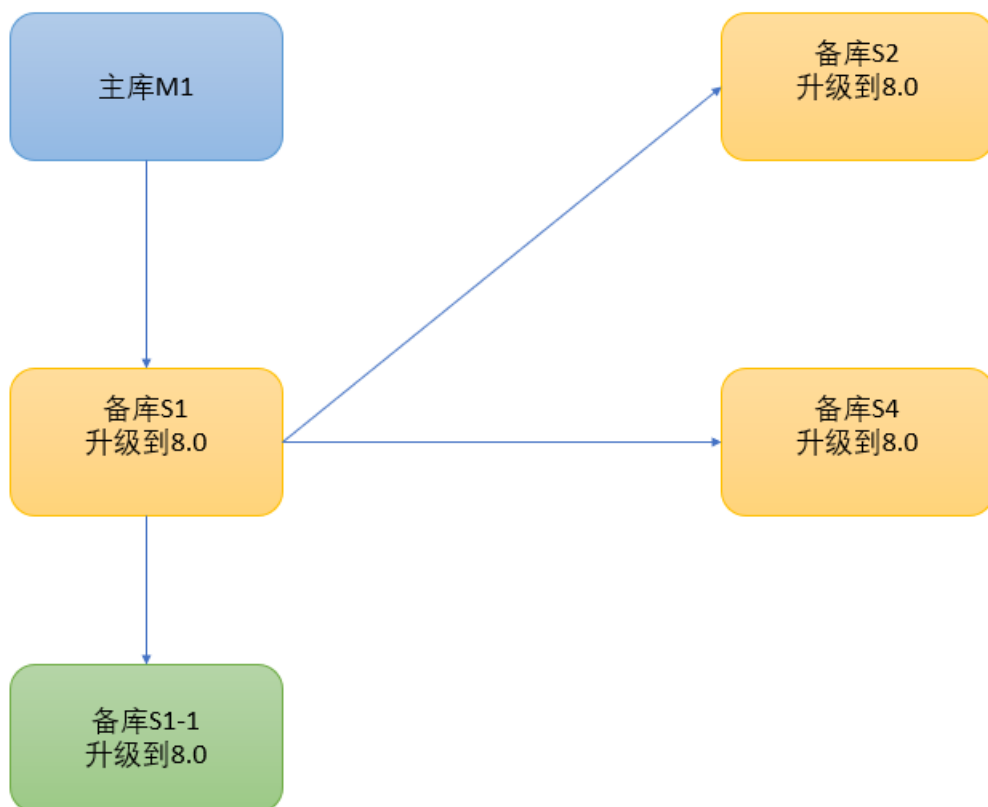
- 1) 首先搭建一个 S1 的级联备库 S1-1



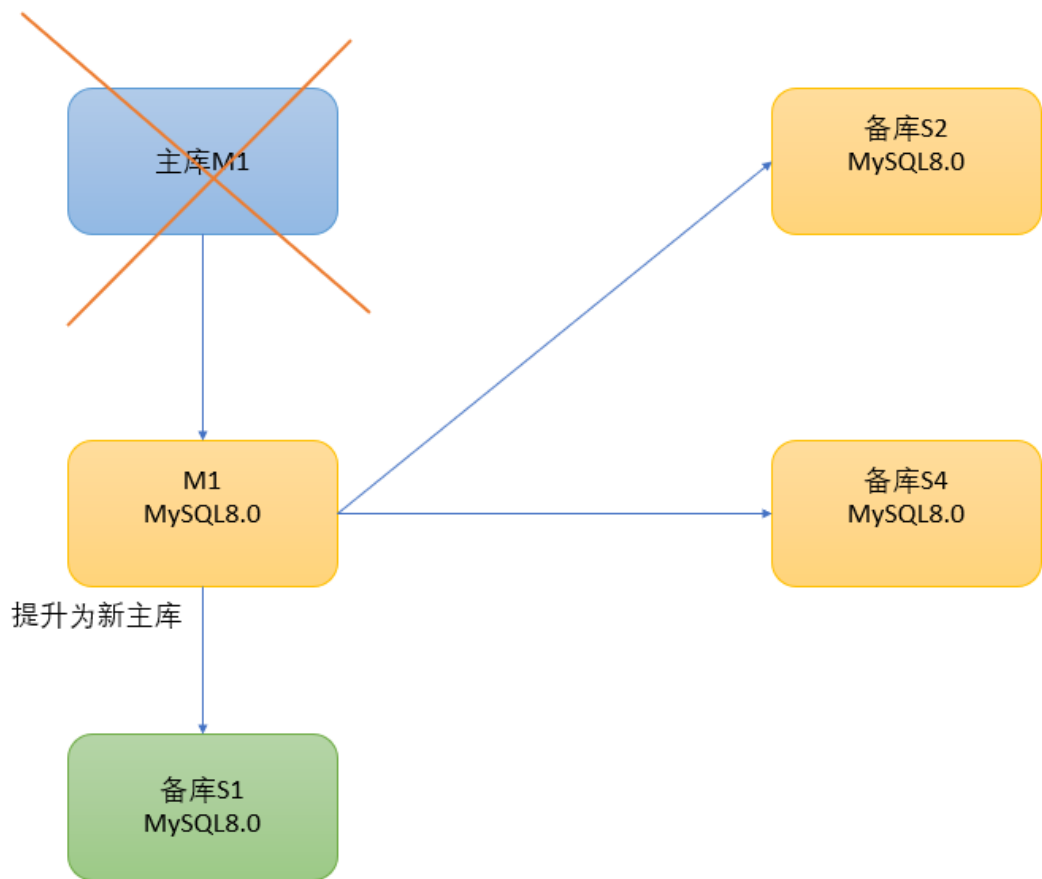
- 2) 将备库按 S1-1, S2, S4 ,S1 的顺序依次升级到 MySQL8.0 (MySQL 低版本可以向高版本同步数据)



3) HA 切换将备库 S2 和 S4 指向 S1



4) 最后进行主库切换,使用切换程序将 S1 切换为 M1, S1-1 变为 S1,完成整个升级流程。



升级注意问题

1. 升级 8.0 后尽量保持 sql mode 和 5.6 一致,减少 SQL 差异的风险。
2. 在 8.0 里面修改系统变量需要授予 session_variables_admin 权限才可以,如果程序需要执行 set variables 操作,则需要授予 session_variables_admin 权限。
3. MySQL8.0 里面 explicit_defaults_for_timestamp 变量默认为 on,需要调整为 off。
4. MySQL8.0 里面 get_lock(key,timeout)函数的key 长度限制最长 64 个字符,如果有使用 get_lock() 函数,需要检查是否超过长度限制。
5. MySQL5.6 里面如果 group by 查询不加 order by 排序默认会按 group 字段排序,在 8.0 里面

是按照数据默认存储方式输出，不会排序。

6. Group by name desc 写法失效 (本身是不规范写法)
7. 注意 SQL 里面不能有 MySQL5.6 之后的新增关键字，否则 SQL 无法执行会引起业务异常。