25 | MySQL是怎么保证高可用的？

2019-01-09 林晓斌

在上一篇文章中，我和你介绍了binlog的基本内容，在一个主备关系中，每个备库接收主库的binlog并执行。

正常情况下，只要主库执行更新生成的所有binlog，都可以传到备库并被正确地执行，备库就能达到跟主库一致的状态，这就是最终一致性。

但是，MySQL要提供高可用能力，只有最终一致性是不够的。为什么这么说呢？今天我就着重和你分析一下。

这里，我再放一次上一篇文章中讲到的双M结构的主备切换流程图。

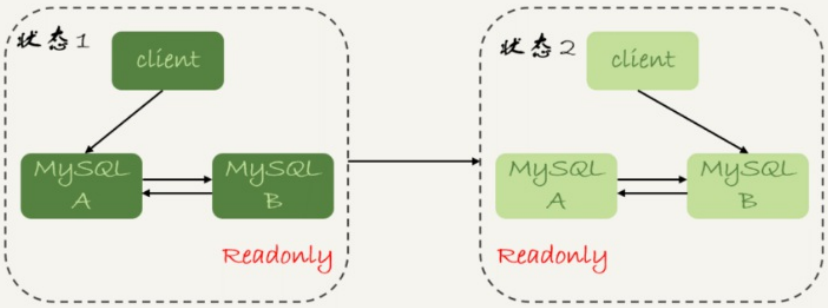

图 1 MySQL主备切换流程--双M结构

主备延迟

主备切换可能是一个主动运维动作，比如软件升级、主库所在机器按计划下线等，也可能是被动操作，比如主库所在机器掉电。

接下来，我们先一起看看主动切换的场景。

在介绍主动切换流程的详细步骤之前，我要先跟你说明一个概念，即“同步延迟”。与数据同步有关的时间点主要包括以下三个：

1. 主库A执行完成一个事务，写入binlog，我们把这个时刻记为T1;
2. 之后传给备库B，我们把备库B接收完这个binlog的时刻记为T2;
3. 备库B执行完成这个事务，我们把这个时刻记为T3。

所谓主备延迟，就是同一个事务，在备库执行完成的时间和主库执行完成的时间之间的差值，也就是T3-T1。

你可以在备库上执行show slave status命令，它的返回结果里面会显示

seconds_behind_master，用于表示当前备库延迟了多少秒。

seconds_behind_master的计算方法是这样的：

1. 每个事务的binlog 里面都有一个时间字段，用于记录主库上写入的时间；
2. 备库取出当前正在执行的事务的时间字段的值，计算它与当前系统时间的差值，得到seconds_behind_master。

可以看到，其实seconds_behind_master这个参数计算的就是T3-T1。所以，我们可以用seconds_behind_master来作为主备延迟的值，这个值的时间精度是秒。

你可能会问，如果主备库机器的系统时间设置不一致，会不会导致主备延迟的值不准？

其实不会的。因为，备库连接到主库的时候，会通过执行SELECT UNIX_TIMESTAMP()函数来获得当前主库的系统时间。如果这时候发现主库的系统时间与自己不一致，备库在执行seconds_behind_master计算的时候会自动扣掉这个差值。

需要说明的是，在网络正常的时候，日志从主库传给备库所需的时间是很短的，即T2-T1的值是非常小的。也就是说，网络正常情况下，主备延迟的主要来源是备库接收完binlog和执行完这个事务之间的时间差。

所以说，主备延迟最直接的表现是，备库消费中转日志（relay log）的速度，比主库生产binlog的速度要慢。接下来，我就和你一起分析下，这可能是由哪些原因导致的。

主备延迟的来源

首先，有些部署条件下，备库所在机器的性能要比主库所在的机器性能差。

一般情况下，有人这么部署时的想法是，反正备库没有请求，所以可以用差一点儿的机器。或者，他们会把20个主库放在4台机器上，而把备库集中在一台机器上。

其实我们都知道，更新请求对IOPS的压力，在主库和备库上是无差别的。所以，做这种部署时，一般都会将备库设置为“非双1”的模式。

但实际上，更新过程中也会触发大量的读操作。所以，当备库主机上的多个备库都在争抢资源的时候，就可能会导致主备延迟了。

当然，这种部署现在比较少了。因为主备可能发生切换，备库随时可能变成主库，所以主备库选用相同规格的机器，并且做对称部署，是现在比较常见的情况。

追问1：但是，做了对称部署以后，还可能会有延迟。这是为什么呢？

这就是**第二种常见的可能了，即备库的压力大**。一般的想法是，主库既然提供了写能力，那么

备库可以提供一些读能力。或者一些运营后台需要的分析语句，不能影响正常业务，所以只能在备库上跑。

我真就见过不少这样的情况。由于主库直接影响业务，大家使用起来会比较克制，反而忽视了备库的压力控制。结果就是，备库上的查询耗费了大量的CPU资源，影响了同步速度，造成主备延迟。

这种情况，我们一般可以这么处理：

1. 一主多从。除了备库外，可以多接几个从库，让这些从库来分担读的压力。
2. 通过binlog输出到外部系统，比如Hadoop这类系统，让外部系统提供统计类查询的能力。

其中，一主多从的方式大都会被采用。因为作为数据库系统，还必须保证有定期全量备份的能力。而从库，就很适合用来做备份。

> 备注：这里需要说明一下，从库和备库在概念上其实差不多。在我们这个专栏里，为了方便描述，我把会在HA过程中被选成新主库的，称为备库，其他的称为从库。

追问2：采用了一主多从，保证备库的压力不会超过主库，还有什么情况可能导致主备延迟吗？

这就是第三种可能了，即大事务。

大事务这种情况很好理解。因为主库上必须等事务执行完成才会写入binlog，再传给备库。所以，如果一个主库上的语句执行10分钟，那这个事务很可能就会导致从库延迟10分钟。

不知道你所在公司的DBA有没有跟你这么说过：不要**一次性地用delete语句删除太多数据**。其实，这就是一个典型的大事务场景。

比如，一些归档类的数据，平时没有注意删除历史数据，等到空间快满了，业务开发人员要一次性地删掉大量历史数据。同时，又因为要避免在高峰期操作会影响业务（至少有这个意识还是很不错的），所以会在晚上执行这些大量数据的删除操作。

结果，负责的DBA同学半夜就会收到延迟报警。然后，DBA团队就要求你后续再删除数据的时候，要控制每个事务删除的数据量，分成多次删除。

另一种典型的大事务场景，就是大表DDL。这个场景，我在前面的文章中介绍过。处理方案就是，计划内的DDL，建议使用gh-ost方案（这里，你可以再回顾下第13篇文章《为什么表数据删掉一半，表文件大小不变？》中的相关内容）。

追问3：如果主库上也不做大事务了，还有什么原因会导致主备延迟吗？

造成主备延迟还有一个大方向的原因，就是**备库的并行复制能力**。这个话题，我会留在下一篇文章再和你详细介绍。

其实还是有不少其他情况会导致主备延迟，如果你还碰到过其他场景，欢迎你在评论区给我留言，我来和你一起分析、讨论。

由于主备延迟的存在，所以在主备切换的时候，就相应的有不同的策略。

可靠性优先策略

在图1的双M结构下，从状态1到状态2切换的详细过程是这样的：

1. 判断备库B现在的seconds_behind_master，如果小于某个值（比如5秒）继续下一步，否则持续重试这一步；
2. 把主库A改成只读状态，即把readonly设置为true；
3. 判断备库B的seconds_behind_master的值，直到这个值变成0为止；
4. 把备库B改成可读写状态，也就是把readonly 设置为false；
5. 把业务请求切到备库B。

这个切换流程，一般是由专门的HA系统来完成的，我们暂时称之为可靠性优先流程。

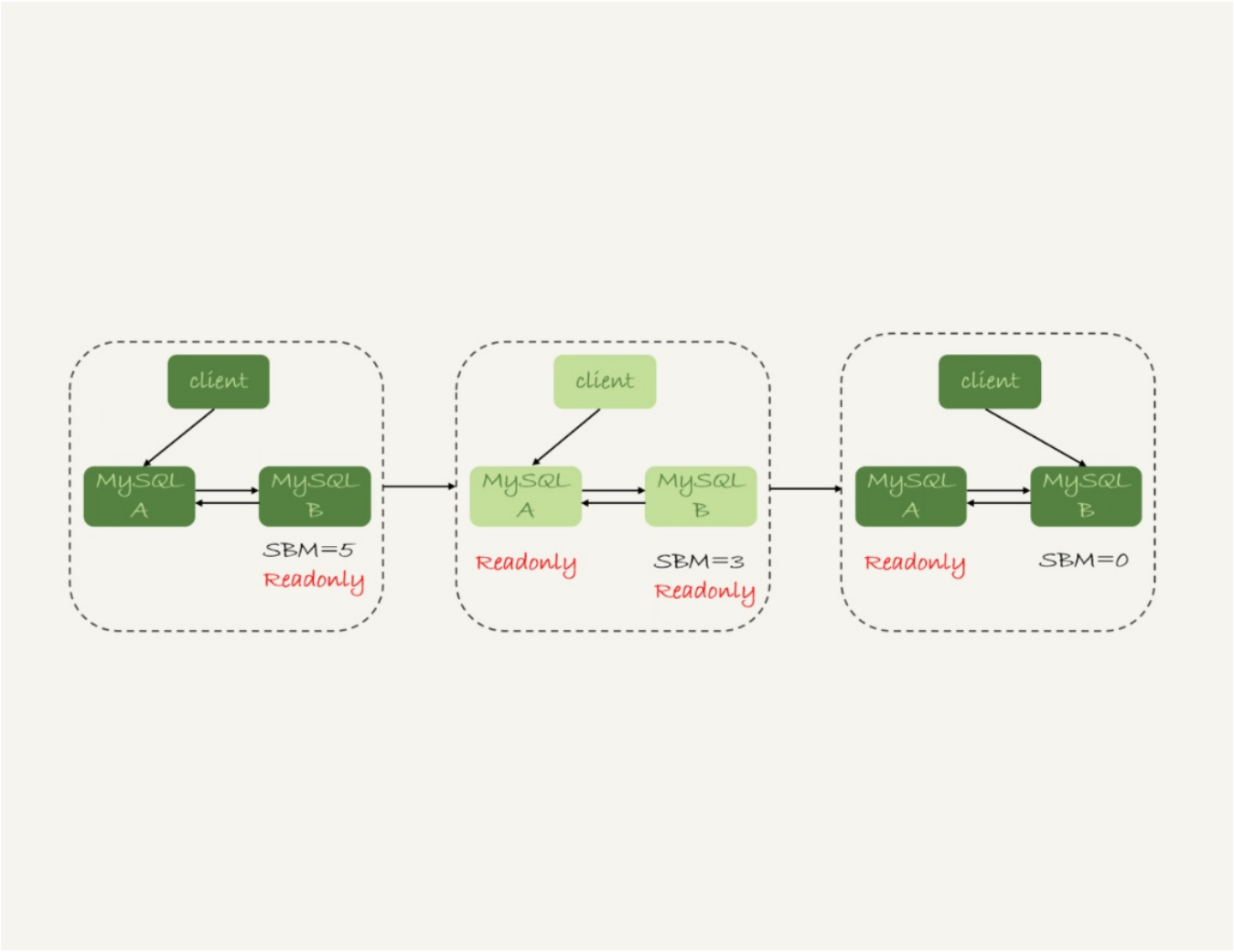

图2 MySQL可靠性优先主备切换流程

备注：图中的SBM，是seconds_behind_master参数的简写。

可以看到，这个切换流程中是有不可用时间的。因为在步骤2之后，主库A和备库B都处于readonly状态，也就是说这时系统处于不可写状态，直到步骤5完成后才能恢复。

在这个不可用状态中，比较耗费时间的是步骤3，可能需要耗费好几秒的时间。这也是为什么需要在步骤1先做判断，确保seconds_behind_master的值足够小。

试想如果一开始主备延迟就长达30分钟，而不先做判断直接切换的话，系统的不可用时间就会长达30分钟，这种情况一般业务都是不可接受的。

当然，系统的不可用时间，是由这个数据可靠性优先的策略决定的。你也可以选择可用性优先的策略，来把这个不可用时间几乎降为0。

可用性优先策略

如果我强行把步骤4、5调整到最开始执行，也就是说不等主备数据同步，直接把连接切到备库B，并且让备库B可以读写，那么系统几乎就没有不可用时间了。

我们把这个切换流程，暂时称作可用性优先流程。这个切换流程的代价，就是可能出现数据不一致的情况。

接下来，我就和你分享一个可用性优先流程产生数据不一致的例子。假设有一个表 t：

```
mysql> CREATE TABLE `t` (
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `c` int(11) unsigned DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB;

insert into t(c) values(1),(2),(3);
```

这个表定义了一个自增主键id，初始化数据后，主库和备库上都是3行数据。接下来，业务人员要继续在表t上执行两条插入语句的命令，依次是：

```
insert into t(c) values(4);
insert into t(c) values(5);
```

假设，现在主库上其他的数据表有大量的更新，导致主备延迟达到5秒。在插入一条c=4的语句

后，发起了主备切换。

图3是**可用性优先策略，且binlog_format=mixed**时的切换流程和数据结果。

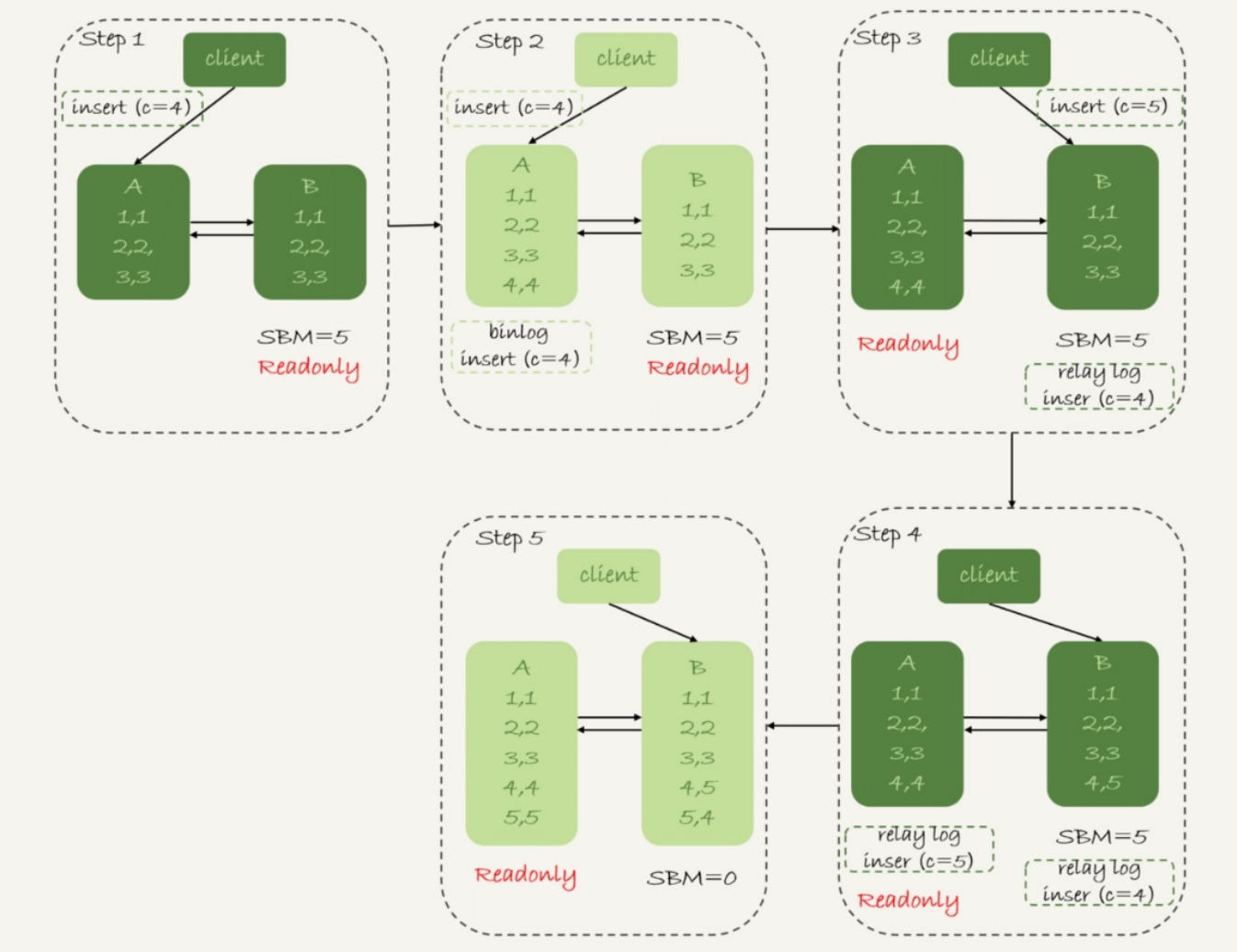

图3 可用性优先策略，且binlog_format=mixed

现在，我们一起分析下这个切换流程：

1. 步骤2中，主库A执行完insert语句，插入了一行数据（4,4），之后开始进行主备切换。
2. 步骤3中，由于主备之间有5秒的延迟，所以备库B还没来得及应用“插入c=4”这个中转日志，就开始接收客户端“插入 c=5”的命令。
3. 步骤4中，备库B插入了一行数据（4,5），并且把这个binlog发给主库A。
4. 步骤5中，备库B执行“插入c=4”这个中转日志，插入了一行数据（5,4）。而直接在备库B执行的“插入c=5”这个语句，传到主库A，就插入了一行新数据（5,5）。

最后的结果就是，主库A和备库B上出现了两行不一致的数据。可以看到，这个数据不一致，是由可用性优先流程导致的。

那么，如果我还是用**可用性优先策略，但设置binlog_format=row**，情况又会怎样呢？

因为row格式在记录binlog的时候，会记录新插入的行的所有字段值，所以最后只会有一行不一致。而且，两边的主备同步的应用线程会报错duplicate key error并停止。也就是说，这种情况下，备库B的(5,4)和主库A的(5,5)这两行数据，都不会被对方执行。

图4中我画出了详细过程，你可以自己再分析一下。

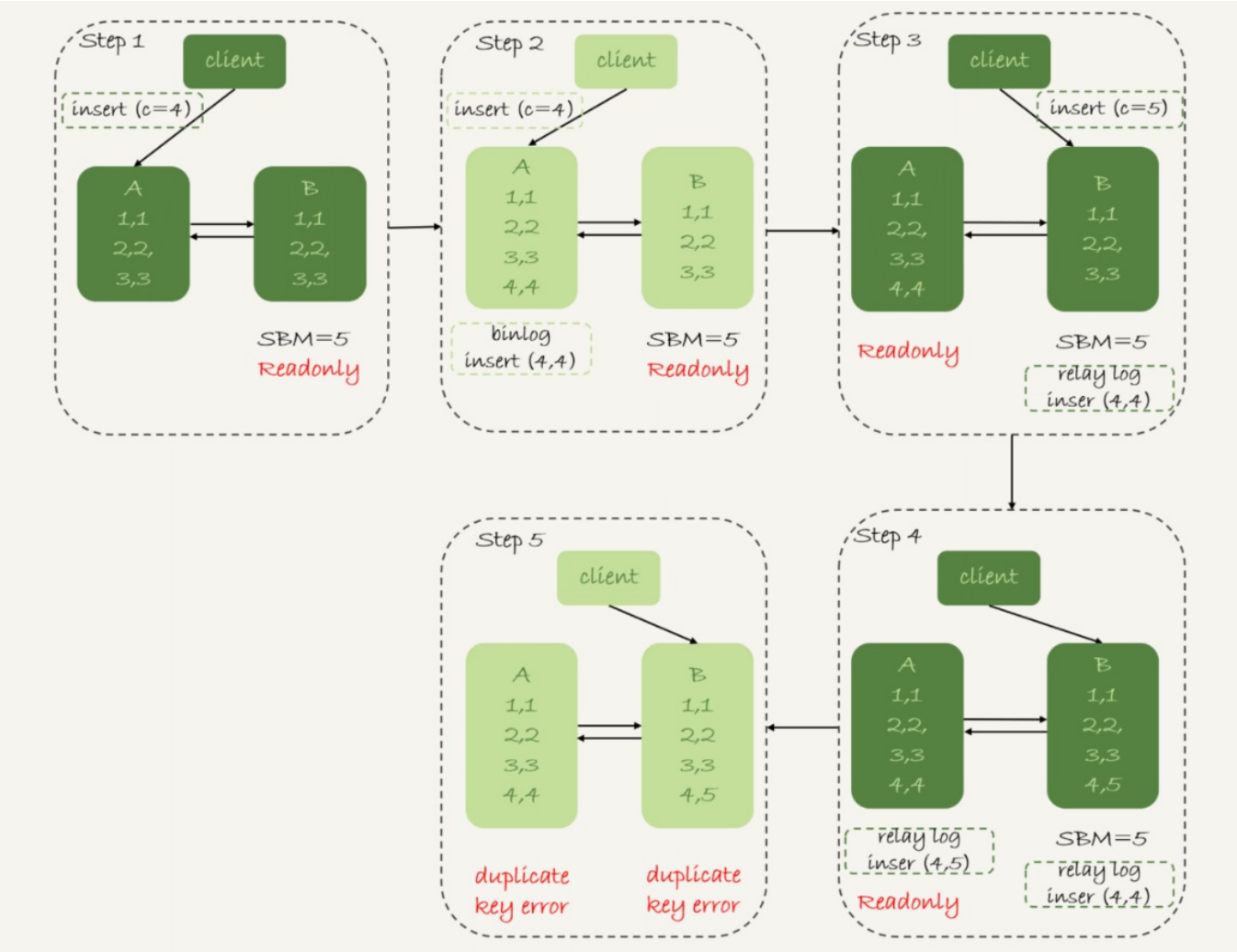

图4 可用性优先策略，且binlog_format=row

从上面的分析中，你可以看到一些结论：

1. 使用row格式的binlog时，数据不一致的问题更容易被发现。而使用mixed或者statement格式的binlog时，数据很可能悄悄地就不一致了。如果你过了很久才发现数据不一致的问题，很可能这时的数据不一致已经不可查，或者连带造成了更多的数据逻辑不一致。

2. 主备切换的可用性优先策略会导致数据不一致。因此，大多数情况下，我都建议你使用可靠性优先策略。毕竟对数据服务来说的话，数据的可靠性一般还是要优于可用性的。

但事无绝对，**有没有哪种情况数据的可用性优先级更高呢？**

答案是，有的。

我曾经碰到过这样的一个场景：

- 有一个库的作用是记录操作日志。这时候，如果数据不一致可以通过binlog来修补，而这个短暂的不一致也不会引发业务问题。
- 同时，业务系统依赖于这个日志写入逻辑，如果这个库不可写，会导致线上的业务操作无法执行。

这时候，你可能就需要选择先强行切换，事后再补数据的策略。

当然，事后复盘的时候，我们想到了一个改进措施就是，让业务逻辑不要依赖于这类日志的写入。也就是说，日志写入这个逻辑模块应该可以降级，比如写到本地文件，或者写到另外一个临时库里面。

这样的话，这种场景就又可以使用可靠性优先策略了。

接下来我们再看看，**按照可靠性优先的思路，异常切换会是什么效果？**

假设，主库A和备库B间的主备延迟是30分钟，这时候主库A掉电了，HA系统要切换B作为主库。我们在主动切换的时候，可以等到主备延迟小于5秒的时候再启动切换，但这时候已经别无选择了。

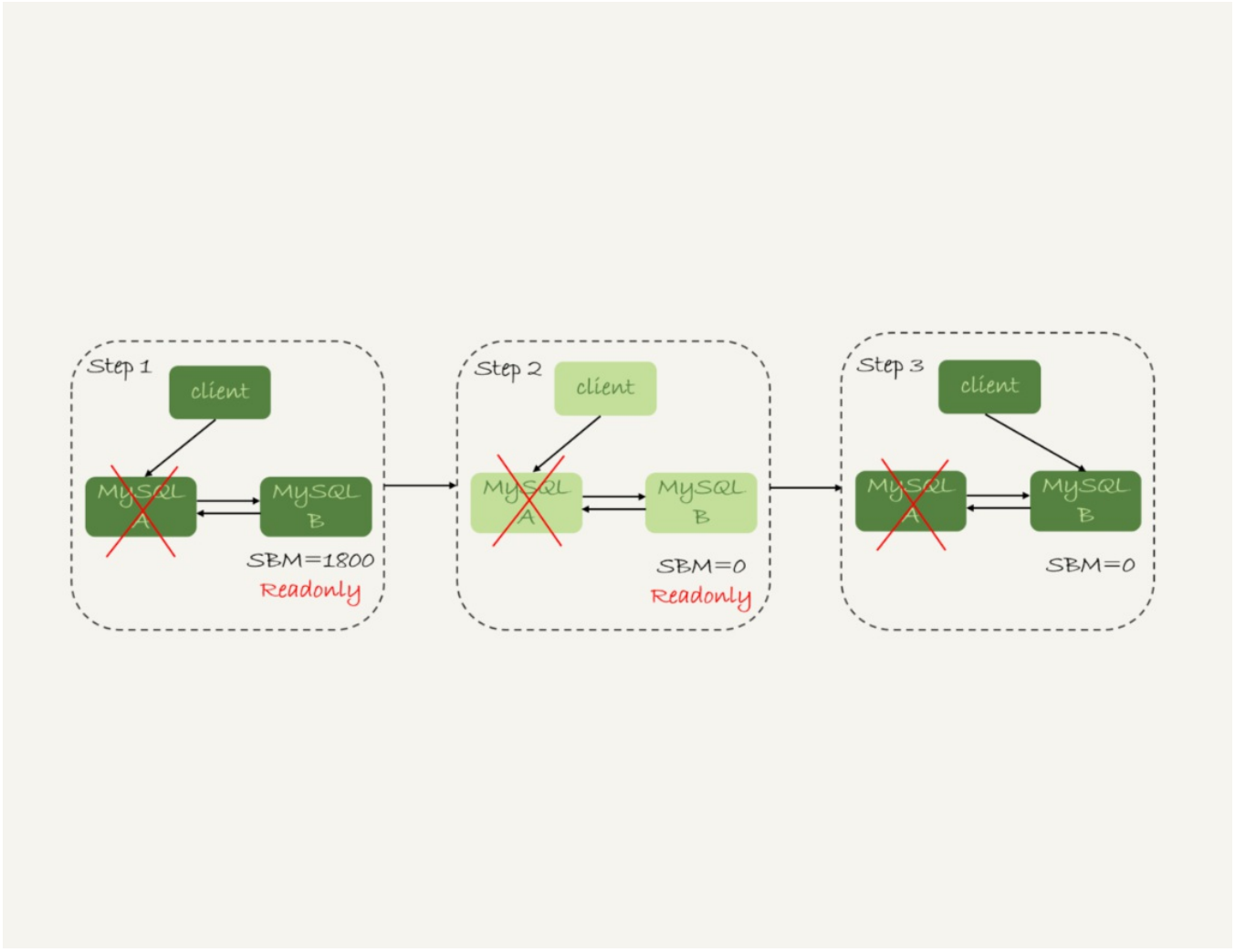

图5 可靠性优先策略，主库不可用

采用可靠性优先策略的话，你就必须得等到备库B的seconds_behind_master=0之后，才能切换。但现在的情况比刚刚更严重，并不是系统只读、不可写的问题了，而是系统处于完全不可用的状态。因为，主库A掉电后，我们的连接还没有切到备库B。

你可能会问，那能不能直接切换到备库B，但是保持B只读呢？

这样也不行。

因为，这段时间内，中转日志还没有应用完成，如果直接发起主备切换，客户端查询看不到之前执行完成的事务，会认为有“数据丢失”。

虽然随着中转日志的继续应用，这些数据会恢复回来，但是对于一些业务来说，查询到“暂时丢失数据的状态”也是不能被接受的。

聊到这里你就知道了，在满足数据可靠性的前提下，MySQL高可用系统的可用性，是依赖于主备延迟的。延迟的时间越小，在主库故障的时候，服务恢复需要的时间就越短，可用性就越高。

小结

今天这篇文章，我先和你介绍了MySQL高可用系统的基础，就是主备切换逻辑。紧接着，我又和你讨论了几种会导致主备延迟的情况，以及相应的改进方向。

然后，由于主备延迟的存在，切换策略就有不同的选择。所以，我又和你一起分析了可靠性优先和可用性优先策略的区别。

在实际的应用中，我更建议使用可靠性优先的策略。毕竟保证数据准确，应该是数据库服务的底线。在这个基础上，通过减少主备延迟，提升系统的可用性。

最后，我给你留下一个思考题吧。

一般现在的数据库运维系统都有备库延迟监控，其实就是在备库上执行 show slave status，采集seconds_behind_master的值。

假设，现在你看到你维护的一个备库，它的延迟监控的图像类似图6，是一个45°斜向上的线段，你觉得可能是什么原因导致呢？你又会怎么去确认这个原因呢？

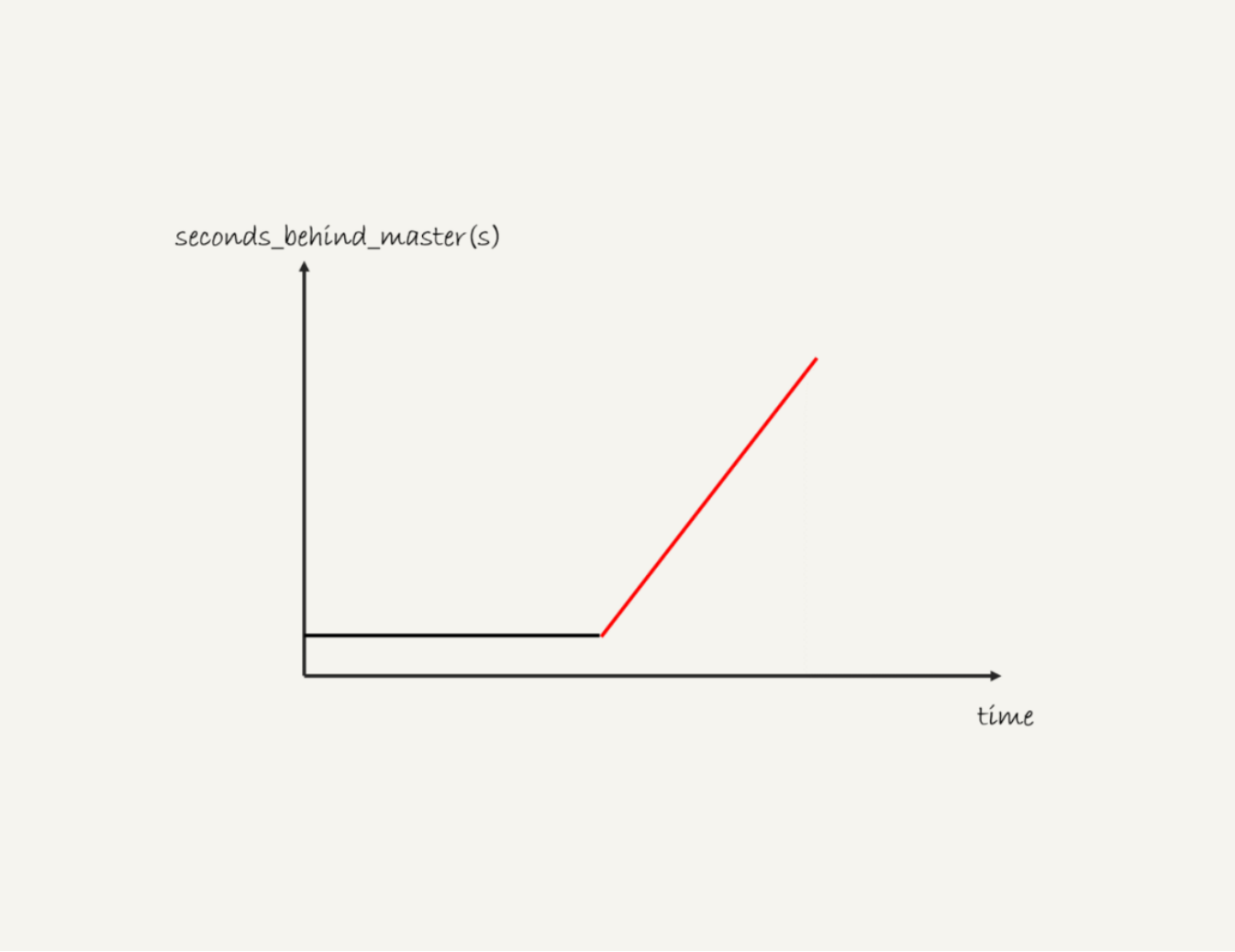

图6 备库延迟

你可以把你的分析写在评论区，我会在下一篇文章的末尾跟你讨论这个问题。感谢你的收听，也欢迎你把这篇文章分享给更多的朋友一起阅读。

上期问题时间

上期我留给你的问题是，什么情况下双M结构会出现循环复制。

一种场景是，在一个主库更新事务后，用命令set global server_id=x修改了server_id。等日志再传回来的时候，发现server_id跟自己的server_id不同，就只能执行了。

另一种场景是，有三个节点的时候，如图7所示，trx1是在节点 B执行的，因此binlog上的server_id就是B，binlog传给节点 A，然后A和A'搭建了双M结构，就会出现循环复制。

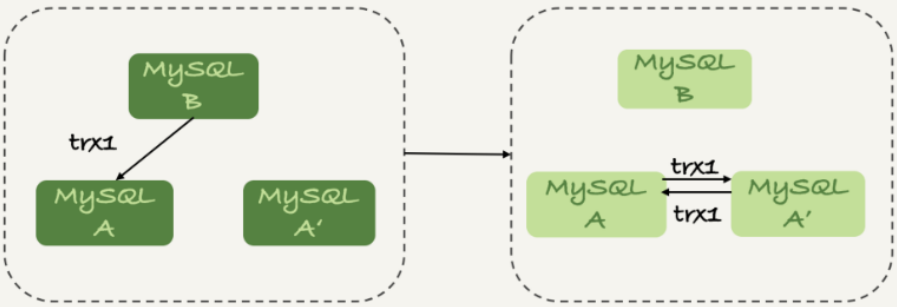

图7 三节点循环复制

这种三节点复制的场景，做数据库迁移的时候会出现。

如果出现了循环复制，可以在A或者A'上，执行如下命令：

```
stop slave;
CHANGE MASTER TO IGNORE_SERVER_IDS=(server_id_of_B);
start slave;
```

这样这个节点收到日志后就不会再执行。过一段时间后，再执行下面的命令把这个值改回来。

```
stop slave;
CHANGE MASTER TO IGNORE_SERVER_IDS=();
start slave;
```

评论区留言点赞板：

> @一大只、@HuaMax 同学提到了第一个复现方法；

> @Jonh同学提到了IGNORE_SERVER_IDS这个解决方法；

> @React 提到，如果主备设置不同的步长，备库是不是可以设置为可读写。我的建议是，只要这个节点设计内就不会有业务直接在上面执行更新，就建议设置为readonly。

精选留言

某、人

33

遇到过下面几种造成主从延迟的情况:
1.主库DML语句并发大,从库qps高
2.从库服务器配置差或者一台服务器上几台从库(资源竞争激烈,特别是io)
3.主库和从库的参数配置不一样
4.大事务(DDL,我觉得DDL也相当于一个大事务)
5.从库上在进行备份操作
6.表上无主键的情况(主库利用索引更改数据,备库回放只能用全表扫描,这种情况可以调整slave_rows_search_algorithms参数适当优化下)
7.设置的是延迟备库
8.备库空间不足的情况下

这期的问题:
看这曲线,应该是从库正在应用一个大事务,或者一个大表上无主键的情况(有该表的更新)

应该是T3随着时间的增长在增长,而T1这个时间点是没变的,造成的现象就是
随着时间的增长,second_behind_master也是有规律的增长

2019-01-10

作者回复

分析的点很准确

2019-01-11

梁中华 7

我有一个比较极端一点的HA问题，假设主库的binlog刚写成功还未来得及把binlog同步到从库，主库就掉电了，这时候从库的数据会不完整吗？
第二个问题，原主库重启加入集群后，那条没有传出去的binlog会如何处理？

2019-01-09

作者回复

1.可能会丢

2. 要看重启之后的拓扑结构了，如果还有节点是这个库的从库，还是会拿走的

2019-01-09

万勇 6

主备同步延迟，工作中常遇到几种情况：
1.主库做大量的dml操作，引起延迟
2.主库有个大事务在处理，引起延迟
3.对myisam存储引擎的表做dml操作，从库会有延迟。
4.利用pt工具对主库的大表做字段新增、修改和添加索引等操作，从库会有延迟。

2019-01-09

作者回复

你是有故事的

2019-01-09

linqw 5

总结下学习完高可用，老师有空帮忙看下
1、主备延迟，就是在同一个事务在备库执行完成的时间和主库执行完成的时间之间的差值，包括主库事务执行完成时间和将binlog发送给备库，备库事务的执行完成时间的差值。每个事务的seconds_behind_master延迟时间，每个事务的 binlog 里面都有一个时间字段，用于记录主库上的写入时间，备库取出当前正在执行的事务的时间字段的值，计算它与当前系统时的差值。
2、主备延迟的来源①首先，有些部署条件下，备库所在机器的性能要比主库所在的机器性能差，原因多个备库部署在同一台机器上，大量的查询会导致io资源的竞争，解决办法是配置”双1“，redo log和binlog都只write fs page cache②备库的压力大，产生的原因大量的查询操作在备库操作，耗费了大量的cpu，导致同步延迟，解决办法，使用一主多从，多个从减少备的查询压力③大事务，因为如果一个大的事务的dml操作导致执行时间过长，将其事务binlog发送给备库，备库也需执行那么长时间，导致主备延迟，解决办法尽量减少大事务，比如delete操作，使用limit分批删除，可以防止大事务也可以减少锁的范围。

④大表的ddl，会导致主库将其ddl binlog发送给备库，备库解析中转日志，同步，后续的dml bin log发送过来，需等待ddl的mdl写锁释放，导致主备延迟。

3、可靠性优先策略，①判断备库 B 现在的 seconds_behind_master如果小于某个值（比如 5 秒）继续下一步，否则持续重试这一步，②把主库 A 改成只读状态，即把 readonly 设置为 true，③判断备库 B 的 seconds_behind_master的值，直到这个值变成 0 为止； 把备库 B 改成可读写也就是把 readonly 设置为 false； 把业务请求切换到备库，个人理解如果发送过来的binlog在中转日志中有多个事务，业务不可用的时间，就是多个事务被运用的总时间。如果非正常情况下，主库掉电，会导致出现的问题，如果备库和主库的延迟时间短，在中转日志运用完成，业务才能正常使用，如果在中转日志还未运用完成，切换为备库会导致之前完成的事务，”数据丢失“，但是在一些业务场景下不可接受。

4、可用性策略，出现的问题：在双m，且binlog_format=mixed，会导致主备数据不一致，使用使用 row 格式的 binlog 时，数据不一致的问题更容易发现，因为binlog row会记录字段的所有值。

5、老师有个问题不太理解，就是主备延迟时，会导致备库在没有运用中转日志时，业务查询时导致”数据丢失“，那如何解决了？

2019-02-17

作者回复

1~4 很好的总结

5. 也是好问题，直接看《28 | 读写分离有哪些坑？》

2019-02-26

7号

5

老师，生产环境有一张表需要清理，该表大小140G。要保留最近一个月的数据，又不能按时间直接用detele删（全表扫描），本来想通过清空分区表删，但是分区表又是哈希的。。有没好的办法呢？

2019-01-09

作者回复

估计下一个月占多少比例，如果比较小就建新表，把数据导过去吧

如果一个月占比高的话，只能一点点删了。

时间字段有索引的话，每个分区按时间过滤出来删除

2019-01-09

崔伟协

4

发生主从切换的时候，主有的最新数据没同步到从，会出现这种情况吗，出现了会怎么样

2019-01-11

作者回复

异常切换有可能的

要根据你的处理策略了，如果不能丢，有几个可选的

1.不切换（等这个库自己恢复起来）

2. 使用semi-sync策略

3. 启动后业务做数据对账（这个一般用得少，成本高）

2019-01-11

John

4

循环复制根本原因是binlog中引入了非当前主机的server id，可以通过ignore server ids过滤，但是一般情况如果出现循环复制，数据的可靠性就值得怀疑了，不管是过滤还是重新找点都很难保证循环的部分完整执行过，最后都要验证数据的状态，属于特别严重故障

2019-01-10

作者回复

你这个方法好

数据问题的话，如果是设置的row 格式的binlog还好，因为insert和delete都会报错，会出现循环的就是update, 然后update都是可重入的

2019-01-10

undifined

4

问题答案：

1. 备库在执行复杂查询，导致资源被占用
2. 备库正在执行一个大事务
3. DML 语句执行

老师我的理解对吗

2019-01-09

作者回复

1不太准确，明天我会提到哈

23对的

2019-01-09

aubrey

3

semi-sync在网络故障超时的情况下会退化成async，这个时候如果刚好主库掉电了，有些binlog还没有传给从库，从库无法判断数据跟主库是否一致，如果强行切换可能会导致丢数据，在金融业务场景下只能＂人工智能＂来做切换，服务中断时间长。AliSQL采用双通道复制更容易判断主备数据是否一致，如果一致可以自动切换，如果不一致才需要人工恢复数据。

2019-02-14

作者回复

内行

2019-02-16

康磊

3

老师你好，现在一般采用读写分离，读的是从库，那么主从如果出现延迟的话，读库就读的不是最新数据，对这种问题有什么好建议吗？

2019-01-11

作者回复

第28篇专门讲这个问题，敬请期待

2019-01-11

cyberty

3

请问老师，如果备库连接主库之后，主库的系统时间修改了，备库同步的时候是否会自动修正？

2019-01-10

作者回复

好问题，不会

2019-01-10

via

3

通过 binlog 输出到外部系统，比如 Hadoop 这类...

文中这个具体是可采用什么工具呢？

2019-01-09

作者回复

canal 可以了解下

2019-01-10

可可

2

老师，我先来讲个笑话
昨天去面试另一家公司，问mysql的问题，问罢之后。
面试官:我看你对mysql了解的还蛮深得，是不是也看了极客时间的。。。～
我: 没有没有(连忙否认，有一种提前看了考试答案的罪恶感)

所以最后我有个问题，如果后面还有这样的问题，老师您觉得我应该怎么回答？

2019-03-27

作者回复

额。。这个不好说，得看面试官是什么类型的

最好的情况是那种，面试官问你的专栏讲到的知识点你都回答了，
然后面试官又把问题延伸了，然后你还顺利回答出来了。

这样就大大方方说学过了， 面试官一定觉得你又好学又勤于思考哟

（先预祝一波顺利拿到offer哈）

2019-04-01

aliang

2

老师，我有一个问题：（1）seconds_behind_master的计算方法是通过从库的系统时间戳减去

sql_thead线程正在执行的binlog_event上的时间戳的差值。当从库系统时间不准时也不会影响seconds的值，因为从库连接到主库时会通过select unix_timestamp（）查询主库的系统时间，若发现和从库不一致会在计算seconds这个值时作调整（2）我的疑惑是在主从网络正常时，select unix_timestamp执行的频率和触发条件是怎样的（换句话说（1）中描述的从库连接到主库这个行为是一直存在的还是有其他触发条件？）。如果这个频率不高，那在两次select unix_timestamp期间从库系统时间发生变化，seconds的值岂不是不准了？

2019-02-23

作者回复

嗯，取时间这个动作只发生在“主从建立连接”的过程中，

如果已经连着的时候，时间戳改掉，是会不准的

2019-02-24

Long

2

老师，您好：

一直追这个课程，解决了我自己的很多知识盲点，或者更加深入的了解一些知识点，已经在试读留言下推荐了这个课程。

但是有的时候在分析问题的时候，看很多日志，比如error log中的死锁日志，报错日志中每个字段是什么意思，以及show innodb engine status中每个日志段的意思，比如在将redo log的时候innodb status就会记录，这样就可以结合课程中的log write，page cahce和fsync逻辑到数据库中实际感受到学到的原因，在应用中怎么一一对应。

比如，show innodb engine status中：

FILE I/O

***

***

Pending normal aio reads: 0 [0, 0, 0, 0, 0, 0, 0, 0] , aio writes: 0 [0, 0, 0, 0, 0, 0, 0, 0] ,

ibuf aio reads: 0, log i/o's: 0, sync i/o's: 0

Pending flushes (fsync) log: 1; buffer pool: 0

14321192292 OS file reads, 120057595 OS file writes, 60413577 OS fsyncs

10 pending preads, 1 pending pwrites

4648.01 reads/s, 16383 avg bytes/read, 48.09 writes/s, 34.98 fsyncs/s

---

LOG

Log sequence number 3893849611607

Log flushed up to 3893849603096

Pages flushed up to 3893705803837

Last checkpoint at 3893705803837

1 pending log writes, 0 pending chkp writes
28053287 log i/o's done, 10.15 log i/o's/second

之前看过网上的一些分析，由于和原理脱离，所以理解的都不深。
非常期待老师能结合error log一些常见问题分析比如dead lock，常见crash啊之类的，
以及show innodb engine status中的重点内容！

多谢

2019-01-09

作者回复

嗯，这这个建议很好，我会考虑加的哈

2019-01-09

EAGLE

2

文中提到“如果一个主库上的语句执行 10 分钟，那这个事务很可能就会导致从库延迟 10 分钟”。这个延迟是针对当前delete事务，还是所有的事务都延迟。

2019-01-09

作者回复

要看有没有开并行复制，默认是串行，一个堵全部堵

2019-01-09

Sr7vy

2

问题1：T3的解释是：备库执行完这个事物。则：Seconds_Behind_Master=T3-T1。如T1=30min，主执行完成，备没有执行。猜测1：那么Seconds_Behind_Master=30min吗？猜测2：备执需要先把这个30min的事务执行完后，Seconds_Behind_Master=30min？
问题2：很多时候是否能把Seconds_Behind_Master当作真正的延迟时间（面试常被问）？如果能，pt-heartbeat存在还有啥意义啊？

2019-01-09

作者回复

问题1:

1.备库没收到，还是收到没执行，前者0，后者30

2. 第二问没看懂

问题2:

类似的，主库把日志都发给备库了吗

2019-01-09

WilliamX

2

大事务导致主从延时的问题，我修改下。
主库上即使有大事务，但只要影响行数不多，传送到从库时间为t1，完成时间为T3，那这个时间gap和是否大事务没关系吧？

2019-01-09

作者回复

嗯，你这里说的大事务，是那种“查很多，更新少”，这种没关系的；

一般我们在说主备延迟的大事务，是指“更新很多行”的

2019-01-10

JJ 2

请问老师，主库断电了，怎么把binlog传给从库同步数据，怎么使的SBM为0主从切换呢？

2019-01-09

作者回复

等应用完就认为是SBM=0

如果不能接受主库有来不及传的，就使用semi-sync

2019-01-09

Sinyo 2

老师，在 binlog row模式下，insert 到表中一条记录，这条记录中某个字段不填，该字段在表中有设置默认值，利用canal解析binlog出来，这个不填的字段会不存在；难道 binlog 只记录有插入的字段数据，表字段的默认数据就不会记录么？mysql版本5.7.22 canal版本1.0.3

2019-01-09

作者回复

不会啊

insert记录的时候肯定都记录的

你的默认值是什么？

2019-01-10