xtrabackup原理及实施

bluesky
软件架构师，从事分布式数据库，云计算等方向研发

收录于 · 分布式系统 >

Xtrabackup是基于InnoDB存储引擎灾难恢复的。它复制InnoDB的数据文件，尽管数据文件在内部是非一致性的，但在执行灾难恢复时可以保证这些数据文件是一致的，并且可用。

官方原理

在InnoDB内部会维护一个redo日志文件，我们也可以叫做事务日志文件。事务日志会存储每一个InnoDB表数据的记录修改。当InnoDB启动时，InnoDB会检查数据文件和事务日志，并执行两个步骤：它应用（前滚）已经提交的事务日志到数据文件，并将修改过但没有提交的数据进行回滚操作。

Xtrabackup在启动时会记住log sequence number（LSN），并且复制所有的数据文件。复制过程需要一些时间，所以这期间如果数据文件有改动，那么将会使数据库处于一个不同的时间点。这时，xtrabackup会运行一个后台进程，用于监视事务日志，并从事务日志复制最新的修改。Xtrabackup必须持续的做这个操作，是因为事务日志是会轮转重复的写入，并且事务日志可以被重用。所以xtrabackup自启动开始，就不停的将事务日志中每个数据文件的修改都记录下来。

上面就是xtrabackup的备份过程。接下来是准备（prepare）过程。在这个过程中，xtrabackup使用之前复制的事务日志，对各个数据文件执行灾难恢复（就像mysql刚启动时要做的一样）。当这个过程结束后，数据库就可以做恢复还原了。

以上的过程在xtrabackup的编译二进制程序中实现。程序innobackupex可以允许我们备份MyISAM表和frm文件从而增加了便捷和功能。Innobackupex会启动xtrabackup，直到xtrabackup复制数据文件后，然后执行FLUSH TABLES WITH READ LOCK来阻止新的写入进来并把MyISAM表数据刷到硬盘上，之后复制MyISAM数据文件，最后释放锁。

备份MyISAM和InnoDB表最终会处于一致，在准备（prepare）过程结束后，InnoDB表数据已经前滚到整个备份结束的点，而不是回滚到xtrabackup刚开始时的点。这个时间点与执行FLUSH TABLES WITH READ LOCK的时间点相同，所以myisam表数据与InnoDB表数据是同步的。类似oracle的，InnoDB的prepare过程可以称为recover（恢复），myisam的数据复制过程可以称为restore（还原）。

Xtrabackup和innobackupex这两个工具都提供了许多前文没有提到的功能特点。手册上有对各个功能都有详细的介绍。简单介绍下，这些工具提供了如流（streaming）备份，增量（incremental）备份等，通过复制数据文件，复制日志文件和提交日志到数据文件（前滚）实现了各种复合备份方式。

自己的理解

Xtrabackup只能备份和恢复InnoDB表，而且只有ibd文件，frm文件它不管，恢复时就需要DBA提供frm。innobackupex可以备份和恢复MyISAM表以及frm文件，并且对xtrabackup也做了很好的封装，所以可以使用innobackupex来备份MySQL数据库。还有一个问题，就是innobackupex备份MyISAM表之前要对全库进行加READ LOCK，阻塞写操作，若备份是在从库上进行的话会影响主从同步，造成延迟。对InnoDB表备份不会阻塞读写。

Xtrabackup增量备份的原理是：

1)首先完成一个完全备份，并记录下此时检查点LSN；

2)然后增量备份时，比较表空间中每个页的LSN是否大于上次备份的LSN，若是则备份该页并记录当前检查点的LSN。

具体来说，首先在logfile中找到并记录最后一个checkpoint（“last checkpoint LSN”），然后开始从LSN的位置开始拷贝InnoDB的logfile到xtrabackup_logfile；然后开始拷贝全部的数据

关于作者

bluesky
软件架构师，从事分布式数...

回答 81
文章 1,024
关注者 10,333

+ 关注他
发私信

文件.ibd；在拷贝全部数据文件结束之后，才停止拷贝logfile。

所以xtrabackup_logfile文件在并发写入很大时也会变得很大，占用很多空间，需要注意。另外当我们使用--stream=tar或者远程备份--remote-host时默认使用/tmp，但最好显示用参数--tmpdir指定，以免把/tmp目录占满影响备份以及系统其它正常服务。

因为logfile✦里面记录全部的数据修改情况，所以即使在备份过程中数据文件被修改过了，恢复时仍然能够通过解析xtrabackup_logfile保持数据的一致。

Xtrabackup的增量备份只能用于InnoDB表，不能用在MyISAM表上。采用增量备份MySQL数据库时xtrabackup会依据上次全备份或增量备份目录对InnoDB表进行增量备份，对MyISAM表会进行全表复制。

流备份（streaming✦）可以将备份直接保存到远程服务器上。

当执行恢复时，由于复制是不锁表的所以此时数据文件都是不一致的，xtrabackup使用之前保存的redo log对各个数据文件检查是否与事务日志的checkpoint一致，执行恢复：

1)根据复制数据文件时以及之后已提交事务产生的事务日志进行前滚；

2)将未提交的事务进行回滚。

这个过程就是MySQL数据库宕机之后执行的crash recovery。

增量备份

在InnoDB中，每个page中都记录LSN信息，每当相关数据发生改变，page的LSN就会自动增加，xtrabackup的增量备份就是依据这一原理进行的。Xtrabackup将上次备份（完全备份集或者也是一个增量备份集）以来LSN改变的page进行备份。

所以，要做增量备份第一次就要做一个完全备份（就是将MySQL实例或者说要备份的数据库表做一个完全复制，同时记录LSN），之后可以基于此进行增量备份以及恢复。

增量备份优点：

1)数据库太大没有足够的空间全量备份✦，增量备份能有效节省空间，并且效率高。

2)支持热备份✦，备份过程不锁表（针对InnoDB而言），不阻塞数据库的读写。

3)每日备份只产生少量数据，也可采用远程备份✦，节省本地空间。

4)备份恢复基于文件操作，降低直接对数据库操作风险。

5)备份效率更高，恢复效率更高。

恢复与还原

backup的恢复过程中包括恢复和还原两个部分。

我们前面已经说了xtrabackup只备份InnoDB表的ibd文件，而innobackupex可以备份包括InnoDB表在内的其他存储引擎的表的所有数据文件。由于不同引擎表备份时的不同，也会让恢复过程看起来不一样。

先来看看完全备份集的恢复。

在InnoDB表的备份或者更直接的说ibd数据文件复制的过程中，数据库处于不一致的状态，所以要将xtraback_logfile中尚未提交的事务进行回滚，以及将已经提交的事务进行前滚，

使各个数据文件处于一个一致性状态，这个过程叫做“准备(prepare)”。

如果你是在一个从库上执行的备份，那说明你没有东西需要回滚，只是简单的apply redo log就可以了。另外在prepare过程中可以使用参数--use-memory增大使用系统内存✦量从而提高恢复速度。

之后，我们就可以根据backup-my.cnf中的配置把数据文件复制回对应的目录了，当然你也可以自己复制回去，但innobackupex都会帮我们完成。在这里，对于InnoDB表✦来说是完成“后准备”动作，我们称之为“恢复(recovery)”，而对于MyISAM表✦来说由于备份时是采用锁表方

式复制的，所以此时只是简单的复制回来，不需要apply log，这个我们称之为“还原(restore)”。

注：本文档里之所以使用恢复和还原，也是和其他数据库比如Oracle看起来一样。

对于增量备份的恢复过程，与完全备份集的恢复类似，只是有少许不同：

1)恢复过程需要使用完全备份集和各个增量备份集，各个备份集的恢复与前面说的一样（前滚和回滚），之后各个增量备份集的redo log都会应用到完全备份集中；

2)对于完全备机集之后产生的新表，要有特殊处理方式，以便恢复后不丢表；

3)要以完全备份集为基础，然后按顺序应用各个增量备份集。

流备份和压缩

提到流备份(streaming)就要说远程备份和备份压缩，先说流备份吧。

流备份是指备份的数据通过标准输出STDOUT传输给tar程序进行归档，而不是单纯的将数据文件保存到指定的备份目录中，参数--stream=tar表示开启流备份功能并打包。同时也可以利用流备份到远程服务器上。

举例来说，

$ innobackupex --stream=TAR ${BACKUP_DIR}/base | gzip > ${BACKUP_DIR}/base.tar.gz
$ innobackupex --stream=TAR ${BACKUP_DIR}/base|ssh somebackupaddr “cat > ${DIR}/base.tar”

当然了，如果你使用了流备份，那么增量备份也就不能用了，因为增量备份需要参考次备份情况，而上次备份却被打包或者压缩了。

在我们现实使用中，更多的使用增量备份，至于归档压缩我们可以通过脚本自主完成。

式复制的，所以此时只是简单的复制回来，不需要apply log，这个我们称之为“还原(restore)”。

注：本文档里之所以使用恢复和还原，也是和其他数据库比如Oracle看起来一样。

对于增量备份的恢复过程，与完全备份集的恢复类似，只是有少许不同：

1)恢复过程需要使用完全备份集和各个增量备份集，各个备份集的恢复与前面说的一样（前滚和回滚），之后各个增量备份集的redo log都会应用到完全备份集中；

2)对于完全备机集之后产生的新表，要有特殊处理方式，以便恢复后不丢表；

3)要以完全备份集为基础，然后按顺序应用各个增量备份集。

流备份和压缩

提到流备份(streaming)就要说远程备份和备份压缩，先说流备份吧。

流备份是指备份的数据通过标准输出STDOUT传输给tar程序进行归档，而不是单纯的将数据文件保存到指定的备份目录中，参数--stream=tar表示开启流备份功能并打包。同时也可以利用流备份到远程服务器上。

举例来说，

$ innobackupex --stream=TAR ${BACKUP_DIR}/base | gzip > ${BACKUP_DIR}/base.tar.gz
$ innobackupex --stream=TAR ${BACKUP_DIR}/base|ssh somebackupaddr “cat > ${DIR}/base.tar”

当然了，如果你使用了流备份，那么增量备份也就不能用了，因为增量备份需要参考次备份情况，而上次备份却被打包或者压缩了。

在我们现实使用中，更多的使用增量备份，至于归档压缩我们可以通过脚本自主完成。

mysql> create table t() engine=innodb; //此处需要DBA手动创建一个同结构的表或表已存在
mysql> ALTER TABLE t DISCARD TABLESPACE; $ cp t.ibd t.exp ${DATA_DIR}/${DB}/
mysql> ALTER TABLE t IMPORT TABLESPACE;

这样的导出导入就可以保住恢复的表可以与数据库其他表保持一致性了。

并行备份

Xtrbackup还支持并行备份，默认情况下xtrabackup备份时只会开启一个进程进行数据文件的备份，若配置参数--parallel=N可以让xtrabackup开启N个子进程对多个数据文件进行并发备份，这样可以加快备份的速度。当然服务器的IO处理能力以及对服务器的影响也是要考虑的，所以另一个参数--throttle=IOS会与它同时使用，这个参数用来限制备份过程中每秒读写的IO次数，对服务器的IO是一个保护。

这两个参数xtrabackup和innobackupex都支持，举例如下：

$ innobackupex --parallel=4 --throttle=400 ${BACKUP_DIR}/part-base

注意：对同一个数据文件只会有一个进程在备份。

其他

Xtrabackup在备份时主要的工作是做数据文件复制，它每次只会读写1MB的数据（即64个page，不能修改），xtrabackup逐页访问1MB数据，使用innodb的buf_page_is_corrupted()函数检查此页的数据是否正常，如果数据不正常，就重新读取这一页，最多重新读取10次，如果还是失败，备份就失败了，退出。

在复制事务日志的时候，每次读写512KB的数据，同样不可以配置。

转自：xtrabackup原理及实施 - 运维生存时间

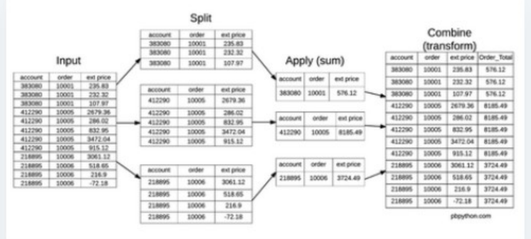
发布于 2021-11-21 12:43

MySQL 备份 Percona

推荐阅读

logistic回归方法的：输入、向前、向后与条件、LR、Wald

数据小兵



如何理解pandas中的transform函数

HiDaD... 发表于数据分析

XtraBackup实现全备&增量备份与恢复

Percona XtraBackup主要是有两个工具，其中一个是xtrabackup，一个是innobackupex，后者是前者封装后的一个脚本。 在针对MySQL的物理备份工具中，大概是最流行也是最强大的工具了，此外著名...

blues... 发表于分布式系统

【DCM-05】Stata应用：多项式Logit模型之条件Logit模型...

暮雪寒泉 发表于离散选择模...