

MySQL数据库的备份与恢复之xtrabackup工具的使用

原创

karony1

2018-03-16 16:42:42

评论(1)

720人阅读

一. 为什么需要备份

在实际生产环境中误删操作，硬件损坏都会导致数据不可用，这个时候我们就需要备份来实现灾难恢复。还需注意的是硬件级的冗余，例如raid，也不能取代备份操作，因为数据误删之后数据也是没有的，所以备份必不可少。

二. 备份的分类

1.物理备份和逻辑备份

物理备份：它是直接拷贝mysql的存储在磁盘上的数据文件。这种备份使用的空间大，特别是innodb存储数据使用的表空间，因为表空间分配大小是按一定的比例增长，所以存在没有使用的空间。但是恢复时使用的时间少。

逻辑备份：将数据以sql语句的形式导出。导出数据和恢复数据所用的时间多，但占用空间小。

2.热备、温备、冷备

热备：对线上环境不见任何锁，不影响线上任何数据库操作。

温备：在备份之间要获取锁，并施加锁。在加锁期间会影响某些操作。

冷备：备份之前停掉服务器，然后在备份。

三. XtraBackup

XtraBackup是Percona旗下的一款开源热备工具，能够在不锁数据库的情况下备份MySQL数据。它能够备份InnoDB，XtraDB，MyISAM表，MySQL版本支持5.1，5.5，5.6，5.7。

XtraBackup具有如下功能：

能够对InnoDB实现热备，无需暂停数据库

能够对MySQL进行增量备份

对MySQL备份能够实现流式压缩并传输给其他服务器
MySQL服务运行时能够在MySQL 服务器之间进行表的迁移
能够很容易创建一个MySQL从服务器
备份MySQL时不会增加服务器负担

XtraBackup能够带InnoDB引擎创建的表实现热备，对MyISAM引擎实现温备。

四. XtraBackup如何实现备份

XtraBackup在备份数据是先备份InnoDB数据，再备份MyISAM数据，最后再记录此次备份的信息。

1.XtraBackup如何对InnoDB实现备份

在解释原理之前我们需要介绍InnoDB引擎的两个重要特性，一个事务特性，一个是存储数据所使用的表空间

1) 事务

在InnoDB引擎中，一旦用户的操作涉及到修改MySQL数据，他并不会直接写入磁盘，就会产生一个事务，并将事务是记录在事务日志中的，而记录日志是使用的round-robin方式，事务日志空间大小一定，也就说后面的事务日志会覆盖前面。只有当用户提交事务后，数据才会持久化到磁盘。

2) 表空间

InnoDB存储数据使用的是表空间，在这个空间中，InnoDB自行组织数据，所有的数据放在page上，每个page的大小固定默认是16KB，你可以通过innodb_page_size将其调大调小。一个page中除了存储数据库数据，还存储了这个表的元数据，用于描述此page。其中一个元数据就是log sequence number日志序列号，它是进行增量备份的基础。

3) 备份原理

XtraBackup在开始的时候会记录日志序列号(LSN)，然后开始拷贝磁盘上的数据文件。在此期间如果数据改变了，那么数据就处于不同的位置。数据改变了，我们不一定要拷贝变化的数据，我们可以记录在此期间的改变的事务日志即可，因为事务日志也记录了数据的改变。所以在拷贝数据期间，XtraBackup会启动一个后台进程监控事务日志，并拷贝在此期间写入的事务日志。这个过程是一个持续过程，因为事务日志会覆盖前面的。

4) 增量备份实现原理

在前面提到过增量备份的实现依赖于LSN，它是一个page所具有的属性。那么在备份时如何利用他实现增量备份呢？首先我们对数据进行完全备份，完全备份时数据有四个页，页的编号为1, 2, 3, 4。完全备份后如果LSN编为1的page数据改变了，此时他的LSN的编号在最大的LSN编号加1，也就是说他的编号会变为5,而不是2。同理如果LSN为3的变化了，他的LSN编号会变为6。增量备份的时候我们只需要备份lsn编号为5,6的page即可。

由于MyISAM引擎存储数据没有LSN所以在物理备份下实现增量备份。

2.XtraBackup备份MyISAM表

XtraBackup是在备份完InnoDB表之后，会运行LOCK TABLES FOR BACKUP拷贝MyISAM表和.frm文件。这个锁是在InnoDB数据拷贝后添加这个锁是一个backup lock他比FLUSH TABLES WITH READ LOCK这个锁更加轻量级。而且在加锁期间并不会影响InnoDB DML操作，这也就是为什么说XtraBackup备份InnoDB是热备，备份MyISAM是温备。

拷贝MyISAM时，并不影响InnoDB数据的操作，也就是说InnoDB数据在改变。这样会导致MyISAM和InnoDB数据在某一时刻并不是一致的。为了能让其数据一致，拷贝完之后还需要事务日志和二进制日志。而这个过程他要施加LOCK BINLOG FOR BACKUP锁，拷贝完成之后才会对二进制日志和table解锁。

3.备份创建的文件解释

- backup-my.cnf：它并不是原始的my.cnf，而是Xtrabackup在备份时获取的InnoDB引擎的相关数据。预还原的时候它会读取此文件的内容，或者从 XtraBackup --defaults-file指明的文件中读取。
- xtrabackup_checkpoints：描述备份的类型（全备或增量备份）、它的状态（例如prepared）和他的LSN范围。可看如下示例

完全备份

```
backup_type = full-backupped
from_lsn = 0
to_lsn = 15188961605
last_lsn = 15188961605
```

增量备份

```
backup_type = incremental
from_lsn = 15188961605
to_lsn = 15189350111
last_lsn = 15189350111
```

- xtrabackup_binlog_info: 在备份的那一刻服务器二进制所处的位置, 通过SHOW MASTER STATUS获取
- xtrabackup_binlog_pos_innodb: InnoDB表当前所处的二进制位置。与InnoDB事务相关
- xtrabackup_logfile: 在备份过程中拷贝的事务日志, 用于还原

五.实验

本实验将会演示备份和还原操作, 使用innobackup命令。

操作系统为centos 7.2

MySQL版本为5.5

1.安装Percona-Xtrabackup

```
~]# wget https://www.percona.com/downloads/XtraBackup/Percona-XtraBackup-2.4.4/binary/redhat/7/x86_64/Percona-XtraBackup-2.4.4-rdf58cf2-el7-x86_64-bundle.tar
展开tarball 其中包含三个rpm,
percona-xtrabackup-24: 包含最新的Percona XtraBackup GA 二进制以及相关文件
percona-xtrabackup-24-debuginfo: 用于调试percona-xtrabackup-24中的二进制程序
percona-xtrabackup-test-24: Percona Xtrabackup测试组件
~]# tar xf Percona-XtraBackup-2.4.4-rdf58cf2-el7-x86_64-bundle.tar
~]# yum install -y percona-xtrabackup-24-debuginfo-2.4.4-1.el7.x86_64.rpm percona-xtrabackup-24-2.4.4-1.el7.x86_64.rpm percona-xtrabackup-test-24-2.4.4-1.el7.x86_64.rpm
```

2.对数据进行完全备份及其还原

1) 创建完全备份

```
~]# innobackupex --defaults-file=/etc/my.cnf --user=root --host=127.0.0.1 --password=123456 /backups/full
```

完全备份成功的话, 最后的输出信息会出现completed OK!

- 参数解释

--defaults-file: 读取mysql默认的参数, 这个要是命令行的第一个参数, 不能是符号链接文件。
--user --host --password 登录主机, 用户名和密码
/backups/full: 备份目录, 如果没有会自动创建

- 生成文件

```
[root@slave ~]# ls -l /backups/full/
total 0
drwxr-x---. 7 root root 227 Mar 16 14:20 2018-03-16_14-20-35
[root@slave ~]# ls -l /backups/full/2018-03-16_14-20-35/
total 18460
-rw-r-----. 1 root root 417 Mar 16 14:20 backup-my.cnf
drwxr-x---. 2 root root 272 Mar 16 14:20 hellodb
-rw-r-----. 1 root root 18874368 Mar 16 14:20 ibdata1
drwxr-x---. 2 root root 4096 Mar 16 14:20 mysql
drwxr-x---. 2 root root 4096 Mar 16 14:20 performance_schema
drwxr-x---. 2 root root 62 Mar 16 14:20 pt
drwxr-x---. 2 root root 20 Mar 16 14:20 test
-rw-r-----. 1 root root 25 Mar 16 14:20 xtrabackup_binlog_info
-rw-r-----. 1 root root 113 Mar 16 14:20 xtrabackup_checkpoints
-rw-r-----. 1 root root 522 Mar 16 14:20 xtrabackup_info
-rw-r-----. 1 root root 2560 Mar 16 14:20 xtrabackup_logfile
```

innobackupex会在指定存放数据的目录下用当前时间创建一个目录，所有生成的备份文件都会这个时间目录下。

- __查看xtrabackup_checkpoints文件__

```
backup_type = full-backupped
from_lsn = 0
to_lsn = 1843549
last_lsn = 1843549
compact = 0
recover_binlog_info = 0
```

可以通过backup_type看到这是一个完全备份，lsn从0到1843549。

2) 完全备份还原

这个备份不能用于恢复，因为备份的数据中可能会包含尚未提交的事务或已经提交但尚未同步至数据文件中的事务，此时数据文件处于不一致的状态。因此，我们现在就是要通过回滚未提交的事务及同步已经提交的事务至数据文件也使得数据文件处于一致性状态。

- 预还原

```
~]# innobackupex --apply-log --redo-only /backups/full/2018-03-16_14-20-35/
```

预还原成功的话，最后的输出信息会出现completed OK!

参数说明

--apply-log：应用在备份期间产生的事务日志。

--redo-only：表明应用日志的方式，将提交的日志重新提交，没有提交的事务不要回滚，因为他很有可

- **还原**

还原时，mysql服务器需要关闭，而且数据目录需要为空。

```
~]# systemctl stop mariadb
```

```
~]# mv /var/lib/mysql /tmp
```

```
~]# mkdir /var/lib/mysql
```

```
~]# innobackupex --defaults-file=/etc/my.cnf --copy-back /backups/full/2018-03-16_14-20-35/
```

```
~]# chown -R mysql:mysql /var/lib/mysql
```

```
~]# systemctl start mariadb
```

2.完全备份+增量备份及其还原

- **备份**

完全备份

```
~]# innobackupex --defaults-file=/etc/my.cnf --user=root --host=127.0.0.1 --password=123456  
/backups/full
```

第一次增量备份

```
~]# innobackupex --defaults-file=/etc/my.cnf --user=root --host=127.0.0.1 --password=123456 -  
-incremental /backups/incr-1 --incremental-basedir=/backups/full/2018-03-16_14-20-35/
```

第二次增量备份

```
~]# innobackupex --defaults-file=/etc/my.cnf --user=root --host=127.0.0.1 --password=123456  
/backups/incr-2 /backups/incr-2 --incremental-basedir=/backups/incr-1/2018-03-16_15-20-32/
```

参数说明

--incremental 表明此备份为增量备份。

--incremental-basedir 指明增量备份一谁为基础。

/backups/incr-1 指明备份所放置的目录

- **预还原**

完全备份预还原

```
~]# innobackupex --apply-log --redo-only /backups/full/2018-03-16_14-20-35/
```

第一次增量备份预还原

```
~]# innobackupex --apply-log --redo-only /backups/incr-1/2018-03-16_15-20-32/ --incremental-  
dir=/backups/full/2018-03-16_14-20-35/
```

第二次增量备份与还原

```
~]# innobackupex --apply-log --redo-only /backups/incr-1/2018-03-16_16-01-44/ --incremental-  
dir=/backups/full/2018-03-16_14-20-35/
```

--incremental-dir：指明以哪个备份作为还原处理。

- 还原

参考备份还原

六. 备份脚本

```
#!/bin/bash
```

```
# Description: innobackupex backup MySQL
```

```
# Date: 2018-01-23
```

```
# Version: 0.1.0
```

```
# Author: lirou
```

#这个备份计划以一周为一个周期，周一为完全备份，周二到周日都是以周一为基础的增量备份。

#以一周还原的时候只需要周一的完全备份和周日的增量备份即可。

```
backup_dir_parent=/var/lib/mysql/backups
```

```
backup_dir=
```

```
full_backup_dir=
```

```
increment_backup_dir=
```

```
backup_times=
```

```
backup_max_times=5
```

```
#登录服务器参数
```

```
mysql_host=127.0.0.1
```

```
mysql_user=root
```

```
mysql_password=123456
```

```
#MySQL服务配置文件
```

```
defaults_file=/etc/my.cnf
```

```
#记录备份是输出的信息
```

```
backup_log=/var/log/innobackup/innobackup.log
```

```
#记录每次备份是否成功
```

```
backup_status=/var/log/innobackup/innobackup.status
```

```
! [ -d $backup_dir_parent ] && mkdir -pv $backup_dir_parent
```

```
! [ -d $backup_log ] && mkdir -pv $backup_log
```

```
week=$(date "+%u")
```

```
if [ $week -eq 1 ]; then
```

```
    backup_times=$(ls -l $backup_dir_parent | grep -i '^d.*backup.*' | wc -l)
```

```
    if [ $backup_times -ge $backup_max_times ];then
```

```
        rm -rf ${backup_dir_parent}/${ls -lt $backup_dir_parent |tail -l)
```

```
    fi
```

```
    backup_dir=${backup_dir_parent}/backup-$(date "+%Y-%m-%d")
```

```
    full_backup_dir=${backup_dir}/full
```

```
    innobackupex --defaults-file=${defaults_file} --host=${mysql_host} --user=${mysql_us
```

```
else
```

```
    backup_dir=${backup_dir_parent}/${ls -lt $backup_dir_parent | grep -i '^d.*backup.*' |  
    increment_backup_dir=${backup_dir}/incr-${week}  
    innobackupex --defaults-file=${defaults_file} --host=${mysql_host} --user=${mysql_us
```

```
fi
```

```
echo "$(date '+%Y-%m-%d'):$?" >> $backup_status
```

此备份脚本在制作定时任务时，一定要确保周一的完全备份执行了，因为周二到周日的增量都以他为基础。

参考

https://www.percona.com/doc/percona-xtrabackup/2.4/how_xtrabackup_works.html

[https://www.percona.com/doc/percona-](https://www.percona.com/doc/percona-xtrabackup/LATEST/innobackupex/innobackupex_option_reference.html)

[xtrabackup/LATEST/innobackupex/innobackupex_option_reference.html](https://www.percona.com/doc/percona-xtrabackup/LATEST/innobackupex/innobackupex_option_reference.html)

<https://dev.mysql.com/doc/refman/5.7/en/innodb-file-space.html>

https://dev.mysql.com/doc/refman/5.7/en/glossary.html#glos_page

<https://www.percona.com/doc/percona-xtrabackup/2.4/xtrabackup-files.html#xtrabackup-files>

版权声明：原创作品，如需转载，请注明出处。否则将追究法律责任
