



链滴

MySQL 数据热备工具 XtraBackup 的介绍及使用

作者: [GeekBoyDqz](#)

原文链接: <https://ld246.com/article/1562078219377>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



XtraBackup介绍

- Percona XtraBackup是一款基于MySQL的服务器的开源热备份实用程序，在备份过程中不会锁定数据库。它可以备份来自MySQL5.1, 5.5, 5.6和5.7服务器上的InnoDB, XtraDB和MyISAM表的数据以及带有XtraDB的Percona服务器。

- Percona XtraBackup为所有版本的Percona Server, MySQL和MariaDB进行MySQL热备份。它行流，压缩和增量MySQL备份。

- 功能：

在不暂停数据库的情况下创建热的InnoDB备份

进行MySQL的增量备份

将压缩的MySQL备份传输到另一台服务器

在MySQL服务器之间移动表格

轻松创建新的MySQL复制从站

在不增加服务器负载的情况下备份MySQL

Xtrabackup特点：

(1) 备份过程快速、可靠

(2) 备份过程不会打断正在执行的事务

(3) 能够基于压缩等功能节约磁盘空间和流量

(4) 自动实现备份检验

(5) 还原速度快

- 下载地址: <https://www.percona.com/downloads/XtraBackup/LATEST/>
- 官方文档: <https://www.percona.com/doc/percona-xtrabackup/LATEST/index.html#user-s-manual>
- 参数说明:
https://www.percona.com/doc/percona-xtrabackup/LATEST/xtrabackup_bin/xbk_option_reference.html

XtraBackup安装部署

1、下载源码包

- 下载官网: <https://www.percona.com/downloads/Percona-XtraBackup-2.4/LATEST/>

说明: Percona XtraBackup 8.0版本是针对MySQL8.0版本使用的

注意: 下载前首先查看使用系统的libgcrypt版本, 下载对应的版本

```
[root@localhost ~]# yum list installed|grep libgcrypt
```

- 如果不是对应系统libgcrypt的版本, 会出现如下报错:

innobackupex: error while loading shared libraries: libgcrypt.so.20: cannot open shared object file: No such file or directory

2、上传软件解压安装

2.1: 上传

- 创建文件存储目录

```
[root@localhost ~]# mkdir /soft
```

- 使用xftp或者lrzsz命令等工具上传

2.2: 解压

```
[root@localhost soft]# tar -zxvf percona-xtrabackup-2.4.14-Linux-x86_64.libgcrypt153.tar.gz
```

2.3: 拷贝文件

```
[root@localhost ~]# cd /soft/percona-xtrabackup-2.4.14-Linux-x86_64/bin/  
[root@localhost bin]# cp innobackupex /usr/bin/  
[root@localhost bin]# cp xtrabackup /usr/bin
```

#查看版本

```
[root@localhost bin]# ./innobackupex --version
```

XtraBackup命令介绍

- xtrabackup有两个主要的工具: xtrabackup、innobackupex

xtrabackup只能备份InnoDB和XtraDB 两种数据表

innobackupex则封装了xtrabackup,同时可以备份MyISAM数据表

- Innobackupex完整备份后生成了几个重要的文件:

xtrabackup_binlog_info: 记录当前最新的LOG Position

xtrabackup_binlog_pos_innodb:innodb log position

xtrabackup_checkpoints: 存放备份的起始位置beginlsn和结束位置endlsn,增量备份需要这个lsn[增量备份可以在这里面看from和to两个值的变化]

- innobackupex

xtrabackup命令只备份数据文件,并不备份数据表结构(.frm), 所以使用xtrabackup恢复的时候必须对应表结构文件(.frm)。用innobackupex命令, 此命令相当于冷备份, 复制数据目录的索引, 数据, 构造文件, 但会有短暂的锁表(时间依赖于MyISAM大小, 对MyISAM进行锁表; MySQL的5.7默认是InnoDB引擎, 但是系统表还是默认的MyISAM引擎, 所以会对系统表进行短暂锁表操作)

- 参数介绍

--defaults-file: 指定my.cnf参数文件的位置[my.cnf配置文件里必须指定datadir]

--apply-log: 同xtrabackup的--prepare参数,一般情况下,在备份完成后, 数据尚且不能用于恢复操作, 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。因此, 此时数据文件仍处理不一致状态。--apply-log的作用是通过回滚未提交的事务及同步已经提交的事务数据文件使数据文件处于一致性状态。

--copy-back: 做数据恢复时将备份数据文件拷贝到MySQL服务器的datadir

--remote-host=HOSTNAME: 通过ssh将备份数据存储在远程服务器上

--stream=[tar]: 备份文件输出格式, 该文件可在XtarBackup binary文件中获得. 在使用参数stream=ar备份的时候,你的xtrabackup_logfile可能会临时放在/tmp目录下,如果你备份的时候并发写入较大话,xtrabackup_logfile可能会很大(5G+),很可能会撑满你的/tmp目录,可以通过参数--tmpdir指定目录来解决这个问题。

--tmpdir=DIRECTORY: 当有指定--remote-host or --stream时, 事务日志临时存储的目录, 默认采用MySQL配置文件中所指定的临时目录tmpdir

--redo-only --apply-log: 强制备份日志时只redo,跳过rollback,这在做增量备份时非常必要

--use-memory=: 该参数在prepare的时候使用,控制prepare时innodb实例使用的内存

--databases=LIST: 列出需要备份的databases,如果没有指定该参数,所有包含MyISAM和InnoDB表数据库都会被备份

--slave-info: 备份从库, 加上--slave-info备份目录下会多生成一个xtrabackup_slave_info 文件, 这会保存主日志文件以及偏移, 文件内容类似于:CHANGE MASTER TO MASTER_LOG_FILE=", MASTER_LOG_POS=0

--socket=SOCKET: 指定mysql.sock所在位置, 以便备份进程登录mysql。

--no-timestamp: 不建立以时间命名的目录

innobackupex使用

1、对某个数据库进行全部备份的命令介绍

```
innobackupex --defaults-file=/etc/my.cnf --user=root --password=123456 --database=dbn  
me --stream=tar /tmp/data/ 2>/tmp/data/error.log|gzip 1>/tmp/data/test.tar.gz
```

- 说明：

```
--database=dbname 单独对指定数据库做备份， 若是不添加此参数那就是对全库做备份  
2>/tmp/data/error.log    标准错误输出信息写入日志中  
1>/tmp/data/test.tar.gz  打包压缩存储到该文件中
```

2、对数据库的全部备份与恢复

- 下面开始测试xtrabackup的全部备份

2.1：备份MySQL全部数据库

```
[root@localhost ~]# innobackupex --defaults-file=/etc/my.cnf --user=root --password=1234  
6 --socket=/data/mysql/mysql.sock /mysql_data_bak/
```

- 注意：

```
--defaults-file必须放置命令行的第一个位置，不然报以下错误  
xtrabackup: Error: --defaults-file must be specified first on the command line
```

2.2：查看备份数据

指定的备份目录下会生成一个以时间命名的文件夹，备份数据备份在此目录下

```
[root@localhost mysql_data_bak]# ll  
total 20  
drwxr-x--- 6 root  root  4096 Jul  1 16:15 2019-07-01_16-15-32  
  
[root@localhost mysql_data_bak]# cd 2019-07-01_16-15-32/
```

2.3：数据文件一致性

- 利用 --apply-log的作用是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处于一致性状态；把已提交的事务合并到ibdata文件

```
[root@localhost ~]# innobackupex --apply-log /mysql_data_bak/2019-07-01_16-15-32/
```

- 查看数据是否有变化，对比之前多几个文件

2.4：全库恢复

2.4.1：关闭数据库

```
[root@localhost ~]# systemctl stop mysqld
```

2.4.2: 将原数据目录备份

```
[root@localhost ~]# mv /data/mysql{,bak}
```

```
[root@localhost ~]# ls /data/  
mysqlbak      #已经更改，此时数据库无法使用
```

2.4.3: 创建新的Datadir

```
[root@localhost ~]# mkdir /data/mysql
```

注意： 与/etc/my.cnf文件定义的datadir路径一致，不然数据库起不来

2.4.4: 恢复数据

```
[root@localhost ~]# innobackupex --defaults-file=/etc/my.cnf --user=root --copy-back /mysql_data_bak/2019-07-01_16-15-32/
```

2.4.5: 重新授权Datadir目录

```
[root@localhost ~]# chown -R mysql. /data/mysql
```

2.4.6: 启动数据库

```
[root@localhost ~]# systemctl start mysqld
```

2.5: 登入数据库查看

```
[root@localhost ~]# mysql -uroot -p
```

```
mysql> show databases;  
mysql> use yptms;  
mysql> show tables;
```

3、对某一个库进行全备与恢复

3.1: 备份数据库

先进入mysql里创建一个测试数据库，然后导入一些数据或者自己创建

3.2: 备份yptms整个数据库

```
[root@localhost ~]# innobackupex --defaults-file=/etc/my.cnf --user=root --password=12346 --host=11.18.198.91 --databases=yptms /mysql_data_bak/
```

3.3: 查看备份文件


```
[root@localhost ~]# cd /mysql_data_bak/
```

```
[root@localhost mysql_data_bak]# ll -h
total 16K
drwxr-x--- 3 root root 149 Jul 1 16:00 2019-07-01_16-00-28
```

注意：指定的备份目录下会生成一个以时间命名的文件夹，备份数据备份在此目录下

```
[root@localhost mysql_data_bak]# cd 2019-07-01_16-00-28/
```

3.4: 数据文件一致性

- 利用 --apply-log的作用是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处于一致性状态; 把已提交的事务合并到ibdata文件

```
[root@localhost ~]# innobackupex --apply-log /mysql_data_bak/2019-07-01_16-15-32/
```

3.5: 恢复数据库

注意：innobackupex不能通过命令恢复指定的数据库，只能在恢复时停止数据库，然后将当前的datadir目录重新命名，然后新建一个数据目录(必须是/etc/my.cnf文件中指定的datadir路径和名称)；最后数据恢复至新建的datadir中后，再将旧的datadir目录下的其他数据库和文件拷贝至新建的datadir目录下，然后在启动数据库(不建议这么操作，数据量大时建议全备后然后全库恢复，保证数据的一致)

3.5.1: 停止数据库

```
[root@localhost ~]# systemctl stop mysqld
```

3.5.2: 重命名原数据目录

```
[root@localhost ~]# mv /data/mysql /data/mysql_bak
```

3.5.3: 恢复数据库

```
[root@localhost ~]# innobackupex --defaults-file=/etc/my.cnf --user=root --copy-back /mysql_data_bak/2019-07-01_16-00-28/
```

执行上面的命令需要保证数据目录是空的，否则会失败，因为目录里有binlog日志，要么移走，要么用手动复制需要的文件。所以说日志不要和数据放在一起！

3.5.4: 数据目录授权

```
[root@localhost ~]# chown -R mysql. /data/mysql
```

3.6: 启动数据库

```
[root@localhost ~]# systemctl start mysqld
```

3.7: 检查数据

```
[root@localhost ~]# mysql -uroot -p
```