# 1. Xtrabackup简介：

MySQL的InnoDB引擎 有个商业的备份工具InnoDB Hotbackup，可以对InnoDB引擎的表实现在线热备。而 percona出品的Xtrabackup，是InnoDB Hotbackup的一个开源替代品，可以在线对InnoDB/XtraDB引擎的表进行物理备份。mysqldump支持在线备份，不过是逻辑备份，效率比较差。xtrabackup是开源的MySQL备份工具，物理备份，效率很不错。

Percona XtraBackup是世界上唯一的开源、免费的MySQL用于非阻塞地备份InnoDB和XtraDB数据库的热备份工具集。

使用Percona XtraBackup，你可以获得以下好处：

**-快速可靠地完成备份**

**-备份期间不间断事务处理**

**-节省磁盘空间和网络带宽**

**-自动备份验证**

**-更高的正常运行时间，由于更快的恢复速度**

Percona XtraBackup支持所有版本的Percona Server、MySQL、MariaDB的流备份、压缩备份、紧凑备份、全量备份、增量增量。

Percona XtraBackup支持非阻塞地备份InnoDB和XtraDB数据库，也支持阻塞式地备份MyISAM等其他存储引擎的数据库。

我们非常建议在生产环境上使用这套工具集用于关键业务的日常备份。

# 2. Percona XtraBackup的特点

这里是一个简短的Percona XtraBackup特征介绍，更多的请参考官方文档。

1. InnoDB不停机不阻塞地创建数据库热备份

2.让MySQL增量备份

3.流压缩MySQL，备份到另一台服务器

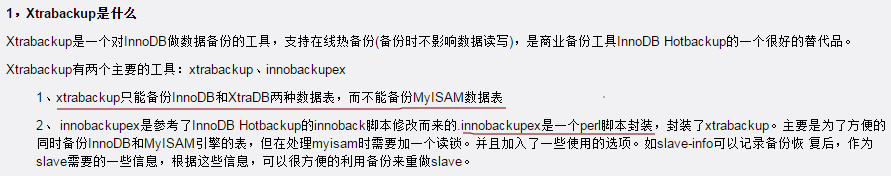
4.在线地在多台MySQL服务器之间复制和移动表

5.更快速地创建MySQL复制服务器(从库)

6.备份MySQL更低的硬件资源负载(CPU、IO)

# 3. Percona XtraBackup2.2版本后一个有趣的改动

现在百度等搜索引擎搜索到的文章都是如何描述XtraBackup工具的:

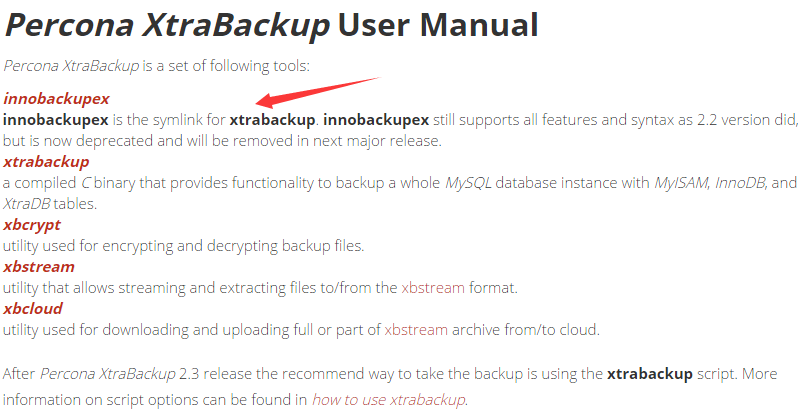


经过我研究这个描述是错的。



我最新下载的2.4.4版，可以清晰地看出innobackupex实际上是xtrabackup的一个软链接，并不是一个perl脚本。

官方是这样描述的:



因为原innobackupex这个perl脚本的功能都封装集成到xtrabackup里了，官方说innobackupex在下一个version将会取消。连软链接都看不到了。所以大家要习惯执行命令改为用xtrabackup了。

而很多博客的教程都是教你如何使用innobackupex，虽然他是xtrabackup的软链接，但他的参数有部分地方不相同，所以还是建议大家到官网学习一下xtrabackup。

例如:

innobackupex --redo-only 这个参数实际上执行的是[xtrabackup --apply-log-only](https://www.percona.com/doc/percona-xtrabackup/2.4/xtrabackup_bin/xbk_option_reference.html" \l "cmdoption-xtrabackup--apply-log-only)

# 4. 使用XtraBackup备份和恢复

## 4.1 XtraBackup的安装

### 4.1.1 安装方法一: rpm的方式安装

1.环境:

CentOS6.5 x86\_64

2.下载

#到此页面可以下载

<https://www.percona.com/downloads/XtraBackup/>

#online文档

https://www.percona.com/doc/percona-xtrabackup/2.2/index.html

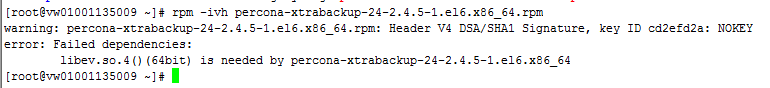
#也可以直接到服务器下wget下载

wget <https://www.percona.com/downloads/XtraBackup/Percona-XtraBackup-2.4.5/binary/redhat/6/x86_64/percona-xtrabackup-24-2.4.5-1.el6.x86_64.rpm>

3.安装

rpm -ivh percona-xtrabackup-24-2.4.5-1.el6.x86\_64.rpm

发现安装报错。



yum install  libev.so.4后依然不能解决。最后百度得知，这个是一个bug，见

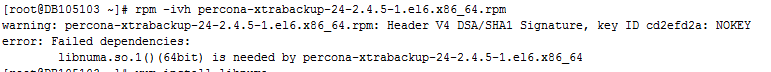
<http://www.cnblogs.com/kerrycode/p/5308719.html>

解决办法:

要用这个rpm装libev才OK

rpm -ivh http://rpmfind.net/linux/dag/redhat/el6/en/x86\_64/dag/RPMS/libev-4.15-1.el6.rf.x86\_64.rpm

有一次遇到过这个报错



解决办法:

yum install numactl.x86\_64

### 4.1.2 安装方法二: 二进制安装

1、下载地址  
https://www.percona.com/downloads/XtraBackup/Percona-XtraBackup-2.4.4/binary/tarball/percona-xtrabackup-2.4.4-Linux-x86\_64.tar.gz  
   
2、安装依赖包  
yum install libaio  
   
3、解压安装  
tar zxvf percona-xtrabackup-2.4.4-Linux-x86\_64.tar.gz  
cd percona-xtrabackup-2.4.4-Linux-x86\_64  
cp -p bin/innobackupex /usr/bin/  
cp -p bin/xtrabackup\* /usr/bin/

## 4.2 Xtrabackup的原理

在备份的时候，备份工具主要执行两个任务来完成备份：  
① 在后台启动一个日志拷贝线程。这个线程会监视InnoDB日志文件，当日志文件发生改变时，这个线程会将发生变化的数据块拷贝到备份目录下一个名为xtrabackup\_logfile的文件中。这个操作是必要的，因为备份可能会持续很长时间，在数据库恢复时，需要所有从备份开始到结束的这些日志文件。  
② 拷贝InnoDB数据文件到指定备份目录下。这不是一个简单的拷贝，备份工具打开并读取文件的方式类似InnoDB，通过读取文件目录并以页（page）为单位进行拷贝。  
  
当数据文件拷贝结束时，xtrabackup会停止日志拷贝线程，并在指定备份目录中创建一个名为xtrabackup\_checkpoints的文件，这个文件包含备份的类型、开始备份的日志序列号和结束备份的日志序列号。

## 4.3 Xtrabackup****常用参数****

**--apply-log-only** 此选项只会导致在准备备份时执行重做阶段。增量备份非常重要。

**--backup** 做一个备份，并将其放置在目标目录。

**--compact** 跳过次要索引页创建紧凑备份

**--compress** 这个选项告诉XtraBackup压缩所有输出数据，包括事务日志文件和元数据文件，使用指定的压缩算法。当前唯一支持的算法quicklz。生成的文件有qpress档案格式，即每\*。QP文件产生的基本上是一一qpress XtraBackup文件存档和可提取的qpress文件归档压缩。

**--compress-threads=#** 此选项指定线程的并行数据压缩的XtraBackup的使用数量。此选项默认为1。并行压缩（压缩线程）可以与并行文件复制（并行）一起使用。例如，并行= 4压缩压缩线程= 2将创建4个IO线程，将读取数据并将其压缩到2个压缩线程。

**--copy-back** 将备份目录中先前备份的文件复制到原始位置。

**--datadir=DIRECTORY** 备份的源目录。这应该是你的MySQL服务器的datadir相同，所以应该从my.cnf如果存在；否则你必须在命令行上指定。

**--defaults-file=[MY.CNF]**  只读取给定文件的默认选项。必须作为命令行的第一个选项。必须是一个真正的文件，它不能是一个符号链接。

**--extra-lsndir=DIRECTORY** 额外保存在备份目录中的文件xtrabackup\_checkpoints的副本到某目录里

**--incremental-basedir=DIRECTORY**  这是增量备份的上一个基础备份目录。

**--incremental-dir=DIRECTORY** 当准备增量备份时，这是增量备份与完整备份相结合的目录，以创建新的完全备份。

**--incremental-lsn=LSN** 在做增量备份时，如果你知道上一次备份的LSN，你也可以用指定**--incremental-lsn=LSN**来代替，而不需要知道和指定--incremental-basedir=DIRECTORY

**--parallel=#**  此选项指定在创建备份时用于同时复制多个数据文件的线程数。默认值是1（即没有并发传输）。

**--password=PASSWORD**  备份用户的密码

**--prepare** 准备恢复

**--rebuild\_indexes** 恢复时重建二级索引，只在--prepare阶段生效

**--stream=name**  指定流备份的格式。备份将会以指定格式输出到STDOUT。目前支持的格式有tar 和 xbstream。如果指定了这个参数，后面需要接tmpdir目录作为处理流的一个中间目录。

**--use-memory=#** 此选项会影响--prepare阶段和--stats分析统计数据时，内存分配多少。其作用是类似于innodb\_buffer\_pool\_size。这参数名字在oracle官方的的InnoDB Hot Backup工具上也有，但不是一回事。默认值是100MB，如果你有足够的可用内存，推荐设置到1到2GB。支持多种单位格式（例如1MB，1m，1GB，1G）。

**--user=USERNAME** 备份用户的账号

**--version**  输出版本信息

## 4.4 全量备份和恢复实验

#查看版本信息

[root@vw01001135009 data]# xtrabackup --version

xtrabackup version 2.4.5 based on MySQL server 5.7.13 Linux (x86\_64) (revision id: e41c0be)

可以看出我安装的xtrabackup版本为2.4.5，基于5.7.13测试、开发的

下面开始全面备份的实验:

--创建备份账户  
mysql> create user xtrabak@'localhost' identified by 'jishu8cc';  
Query OK, 0 rows affected (0.18 sec)  
mysql> grant reload,lock tables,replication client,super,PROCESS on \*.\* to xtrabak@'localhost';  
Query OK, 0 rows affected (0.00 sec)  
mysql> flush privileges;  
Query OK, 0 rows affected (0.44 sec)

--创建全量备份

[root@vw01001135009 base]# mkdir -p /data/backups/

[root@vw01001135009 base]# xtrabackup --backup --target-dir=/data/backups/base



--查看备份的xtrabackup\_checkpoints

[root@vw01001135009 base]# pwd

/data/backups/base

[root@vw01001135009 base]# cat xtrabackup\_checkpoints

backup\_type = full-backuped

from\_lsn = 0

to\_lsn = 34355296377

last\_lsn = 34355296386

compact = 0

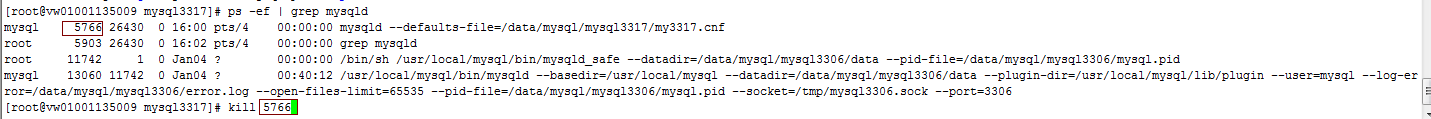
recover\_binlog\_info = 0

可以看出备份类型是全备份，没有经过压缩。开始的lsn是0，结束的lsn是34355296377

--利用此备份恢复出一个3317的mysql实例出来。

#创建mysql3317实例的过程，略。。。

#关闭3317实例的数据库



#删除3317实例的数据库文件 #删库和跑路~

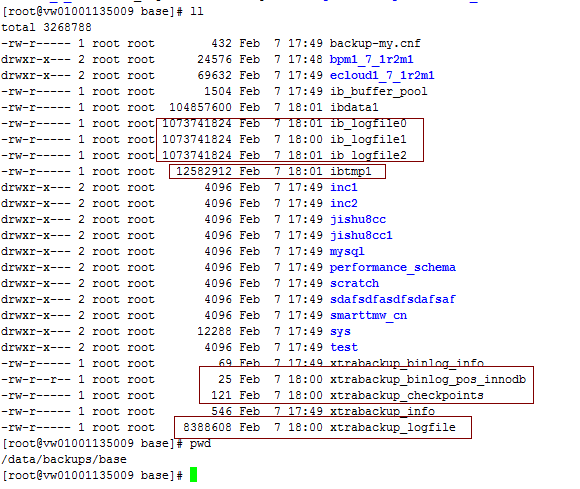
[root@vw01001135009 data]# pwd

/data/mysql/mysql3317/data

[root@vw01001135009 data]# rm -rf \*

#准备数据库

[root@vw01001135009 data]# xtrabackup --defaults-file=/data/mysql/mysql3317/my3317.cnf --prepare --target-dir=/data/backups/base



你可以看出准备好数据库后，这些文件修改时间放生了变化，准备数据库的过程就是利用备份写redolog，用于后面的recovery

#恢复

[root@vw01001135009 data]# xtrabackup --defaults-file=/data/mysql/mysql3317/my3317.cnf --copy-back --target-dir=/data/backups/base

#修改权限(恢复后文件默认权限是root,需要修改)

[root@vw01001135009 mysql3317]# pwd

/data/mysql/mysql3317

[root@vw01001135009 mysql3317]# ls

data error.log logs my3317.cnf tmp

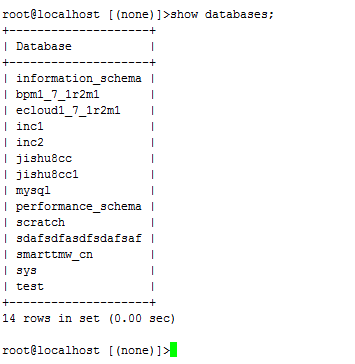
[root@vw01001135009 mysql3317]# chown mysql:mysql data -R

#启动数据库

mysqld --defaults-file=/data/mysql/mysql3317/my3317.cnf &

#进入数据库验证

mysql -uroot -p -S /tmp/mysql3317.sock

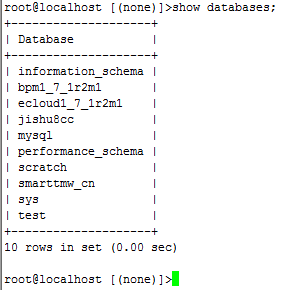


明天会继续写增量备份、紧凑备份、压缩备份、流备份等等的实验过程。

## 4.5 增量备份和恢复实验

--开始增量备份实验

mysql -uroot -p -S /tmp/mysql3306.sock



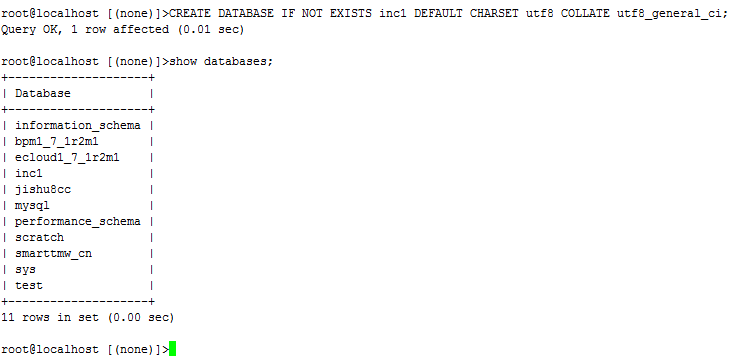
#创建一份基础的全量备份

[root@vw01001135009 base]# mkdir -p /data/backups/

[root@vw01001135009 base]# xtrabackup --defaults-file=/etc/my.cnf --user=xtrabak --password=jishu8cc --backup --target-dir=/data/backups/base

#数据库建一个新的库inc1

mysql -uroot -p -S /tmp/mysql3306.sock

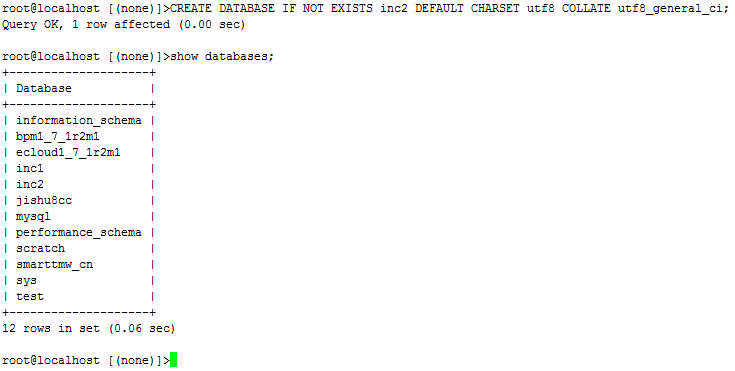


#制作增量备份一

xtrabackup --defaults-file=/etc/my.cnf --user=xtrabak --password=jishu8cc --backup --target-dir=/data/backups/inc1 --incremental-basedir=/data/backups/base

#数据库建一个新的库inc2

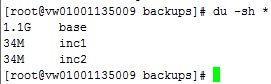
mysql -uroot -p -S /tmp/mysql3306.sock



#制作增量备份二

xtrabackup --defaults-file=/etc/my.cnf --user=xtrabak --password=jishu8cc --backup --target-dir=/data/backups/inc2 --incremental-basedir=/data/backups/inc1

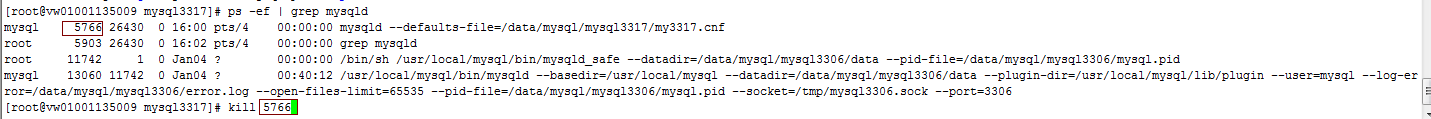
#查看备份的大小，如下



--开始增量备份的恢复实验了

#和之前一样，先创建一个mysql3317实例，过程略。。

#关闭3317实例的数据库(关库)



#删除3317实例的数据库文件(删库)

[root@vw01001135009 data]# pwd

/data/mysql/mysql3317/data

[root@vw01001135009 data]# rm -rf \*

#准备数据库

[root@vw01001135009 data]# xtrabackup --defaults-file=/data/mysql/mysql3317/my3317.cnf --user=xtrabak --password=jishu8cc --prepare --apply-log-only --target-dir=/data/backups/base

[root@vw01001135009 data]# xtrabackup --defaults-file=/data/mysql/mysql3317/my3317.cnf --user=xtrabak --password=jishu8cc --prepare --apply-log-only --target-dir=/data/backups/base --incremental-dir=/data/backups/inc1

[root@vw01001135009 data]# xtrabackup --defaults-file=/data/mysql/mysql3317/my3317.cnf --user=xtrabak --password=jishu8cc --prepare --target-dir=/data/backups/base --incremental-dir=/data/backups/inc2

#恢复数据库

innobackupex --defaults-file=/data/mysql/mysql3317/my3317.cnf --copy-back /data/backups/base

#修改权限(恢复后文件默认权限是root,需要修改)

[root@vw01001135009 mysql3317]# pwd

/data/mysql/mysql3317

[root@vw01001135009 mysql3317]# ls

data error.log logs my3317.cnf tmp

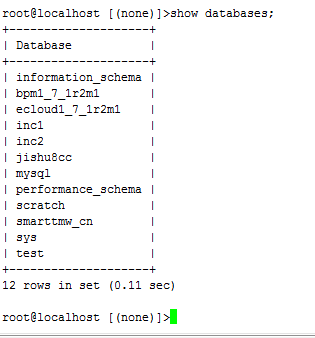
[root@vw01001135009 mysql3317]# chown mysql:mysql data -R

#启动数据库

mysqld --defaults-file=/data/mysql/mysql3317/my3317.cnf &

#进入数据库验证

mysql -uroot -p -S /tmp/mysql3317.sock



增量备份技术细节讲解:

1.增量备份前需要一个基础全备，没有这个基础全备，你的增量备份等于无效！

2.多次增量备份，可以基于上一次全备做增量备份，也可以基于上一次的增量备份做增量备份。我这里举得例子是举的”基于上一次的增量备份做增量备份”。

3.在增量备份的还原准备步骤时(--prepare)，前面的步骤都要加--apply-log-only，最后一步不要加--apply-log-only

查看/data/backups/base(--target-dir=/data/backups/base)里的xtrabackup\_checkpoints文件，你就会发现，前面准备步骤，只合并log，但不进行事务回滚，而最后一步有。

注意:

如果不使用“--apply-log-only”选项来防止回滚阶段，则增量备份将无用。事务回滚后，无法应用进一步的增量备份。

[root@vw01001135009 base]# cat xtrabackup\_checkpoints

backup\_type = full-backuped #--prepare之前的状态

from\_lsn = 0

to\_lsn = 34355354075

last\_lsn = 34355354084

compact = 0

recover\_binlog\_info = 0

[root@vw01001135009 base]# cat xtrabackup\_checkpoints

backup\_type = log-applied #--prepare --apply-log-only 增量备份inc1后状态

from\_lsn = 0

to\_lsn = 34355358498 #增量备份inc1的to\_lsn被合并过来

last\_lsn = 34355358507

compact = 0

recover\_binlog\_info = 0

[root@vw01001135009 base]# cat xtrabackup\_checkpoints

backup\_type = full-prepared #--prepare (不加--apply-log-only)增量备份inc2后状态，完全准备好，可以用于copy-back 恢复的状态

from\_lsn = 0

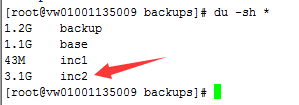
to\_lsn = 34355363033 #增量备份inc2的to\_lsn被合并过来

last\_lsn = 34355363042

compact = 0

recover\_binlog\_info = 0

你可以看出inc2目录增大很多倍，因为这个步骤是full-prepared，准备数据文件时对多余事务进行了事务回滚，生成了redolog。



## 4.6 流备份、压缩备份综合实验

场景:

生产服务器，采用流备份，带压缩参数、紧凑备份参数，备份还原到测试服务器，计算耗时。下面过程并不是全部步骤，是耗时的主要步骤。

#带多线程压缩的流备份

xtrabackup --backup --stream=xbstream --compress --compress-threads=4 --extra-lsndir=/data/backups/lsnbase >/data/backups/base.xbsteam

压缩前106G，压缩后19G，总耗时才15分33秒

#传输到测试服务器

scp base.xbsteam 10.1.134.190:/data/backups

8分1秒

#解压步骤1

xbstream -xv -C xtrabackup\_backupfiles/ < base.xbsteam

解压5分钟

#解压步骤2

cd xtrabackup\_backupfiles/

for bf in `find . -iname "\*\.qp"`; do qpress -d $bf $(dirname $bf) && rm $bf -rf; done

解压11分钟，解压后106G

#准备

xtrabackup --prepare --target-dir=/data/backups/xtrabackup\_backupfiles

准备只有18秒

#copy-back/move-back

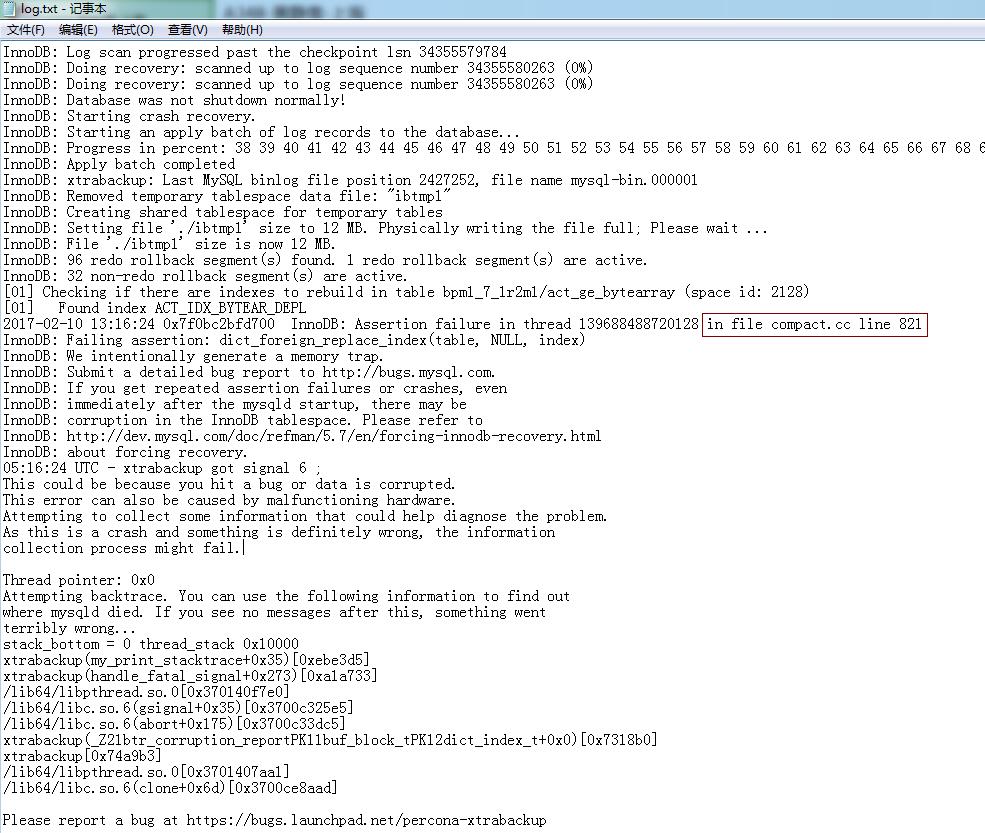
xtrabackup --copy-back --target-dir=/data/backups/xtrabackup\_backupfiles

copy-back的步骤几乎就是拷贝文件的速度，如果用move-back的话就几乎不用时间，我因为测试环境没有足够的硬盘空间，我估计copy大概需要10分钟吧

注:

之前我还测试了紧凑备份，即带--compact参数，紧凑备份的意思是备份时删除二级索引，在还原时重新创建二级索引，这样备份的压缩文件会比正常压缩还再小25%左右。但是准备恢复步骤会慢一些。

猜测总速度应该差不多，但这个参数在准备阶段测试不成功，不知道是不是遇到bug了。



逻辑备份及还原对比

mysqldump --single-transaction --master-data=2 -ER -A >/data/103all.sql

64分钟 74G

压缩

tar zcvf 103all.sql.tgz 103all.sql

26分21秒 5.2G

传输

scp 103all.sql.tgz 10.1.134.190:/data

2分16秒 5.2G

解压

tar zxvf 103all.sql.tgz

14分33秒

导入

mysql </data/103all.sql

306分钟8秒

# 5.使用注意:

## 1.为什么文档和网上的方法有出入？

答:

因为本文档是基于官方最新的英文文档做的实验，网上的方法大多数都是用旧的方法，使用的innobackupex的方法，如果您使用的版本低于5.7，建议安装旧版xtrabackup工具，并使用旧方法。如果您使用的版本大于等于5.7，旧的使用方法(innobackupex)依然兼容，但建议用我的这个新的方法(xtrabackup)，具体见我前面写的” 3.Percona XtraBackup2.2版本后一个有趣的改动”

## 2.为什么我没有使用--defaults-file=/etc/my.cnf

答:

因为xtrabackup和mysqld一样，不指定defaults-file默认第一个加载顺序读的是/etc/my.cnf，作为懒人，肯定能省就省的，能不输入就不输入。如果你的配置文件不在/etc/my.cnf。那么还是需要添加上defaults-file参数的，例如:

xtrabackup --defaults-file=/data/mysql/mysql3317/my3317.cnf --backup --target-dir=/data/backups/base

## 3.为什么我不需要输入数据库账号密码就可以备份？

答:

实际上在--backup阶段是需要输入的。我没有输入是因为我有这样一个配置文件

cat ~/.my.cnf

[mysql]

port = 3306

socket = /tmp/mysql3306.sock

user=dba

password=jishu8cc

所以实际上我是使用dba这个账号备份的。