

---

# xtrabackup 对 MySQL 数据库的备份及恢复

此文档，只是简单地研究了下了，用于数据库全备和增备，效果不错  
因为目前对此软件未有需求，所以，文档并不完善，等有时间了，再进行完善吧  
此文档资料，互联网收集而成  
本人博客：<http://www.linuxyw.com>

## xtrabackup

xtrabackup 是 percona 的一个开源项目，可以热备份 innodb，XtraDB, 和 MyISAM(会锁表)。对 MyISAM 存储引擎会锁表，也是很郁闷的因为线上使用的是 Innodb 和 MyISAM 两种存储引擎，比较头疼!! Xtrabackup 是一个对 InnoDB 做数据备份的工具，支持在线热备份(备份时不影响数据读写)，是商业备份工具 InnoDB Hotbackup 的一个很好的替代品。

## 官方原理

以下内容转载：<http://www.baidu-ops.com/2013/05/26/xtrabackup/>

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

<http://www.linuxyw.com> Email:63780668@qq.com

---

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)、对于完全备份集之后产生的新表，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"
```

<http://www.linuxyw.com> Email:63780668@qq.com

---

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。

### 部分备份和恢复

xtrabackup 可以只备份/恢复部分库表，可以正则模式匹配或者是你想备份库表的列表，但 InnoDB 表必须是独立表空间，同时不能使用流备份功能。

1)、使用正则模式匹配备份部分库表，需要使用参数--include，语句类似如下：

```
$ innobackupex --include='^qb.*' ${BACKUP_DIR}/part-base
```

2)、使用数据库列表备份部分库，需要使用参数--databases，语句类似如下：

```
$ innobackupex --databases=qb0 qb1 qb2 qb3 ${BACKUP_DIR}/part-base
```

3)、使用表列表备份部分表，需要使用参数--tables-file，语句类似如下：

```
$ innobackupex --tables-list=${CONF_DIR}/tab.conf ${BACKUP_DIR}/part-base
```

注：在我们的现实应用中，很少会只备份集群中部分库表，所以只是了解此功能即可，若有现实需要可以参考 percona 官方资料以获取更多信息。

能备份部分库表，也就能根据完全备份集进行部分库表的恢复，在现实中很少会用到，但还是说一下吧。

首先在“准备 prepare”的过程中，使用参数--export 将表导出，这个导出会将每个 InnoDB 表创建一个以 .exp 结尾的文件，这些文件为之后的导入过程服务。

```
$ innobackupex --apply-log --export ${BACKUP_DIR}/base
```

然后将你需要恢复的表的 ibd 和 exp 文件复制到目标机器，在目标机器上执行导入：

```
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

xtrabackup 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

<http://www.linuxyw.com> Email:63780668@qq.com

---

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。

之前我在维护 mysql 数据库的时候, 使用 mysqldump 来进行备份与恢复, 在备份的时候锁住表, 然后全部备份, 在数据少的时候没问题, 但如果数据很多, 不允许锁表, 同时需要恢复数据块的情况, mysqldump 就不适合了, 我在恢复一个 4G 数据文件的数据库的时候, 恢复的数据是使用 mysqldump 的数据, 恢复了 3 个小时还没有反应, 造成的影响很严重, 所以我开始寻找其他的别发软件来满足以上的需求, 幸好找到了, 就是使用 xtrabackup 来进行备份与恢复, 恢复 4G 数据文件的数据库, 仅需要 14 秒, 同时在备份的时候不会锁表, 而且支持增量备份, 所以把我的比较分享给大家, 希望对大家有益!

#### 官方地址:

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

#### 下载:

[http://www.percona.com/edir/downloads/XtraBackup/XtraBackup-2.1.5/binary/Linux/x86\\_64/percona-xtrabackup-2.1.5-680-Linux-x86\\_64.tar.gz](http://www.percona.com/edir/downloads/XtraBackup/XtraBackup-2.1.5/binary/Linux/x86_64/percona-xtrabackup-2.1.5-680-Linux-x86_64.tar.gz)

#### 安装:

```
yum install perl-Time-HiRes -y
yum -y install perl-DBD-MySQL.x86_64
tar xvf percona-xtrabackup-2.1.5-680-Linux-x86_64.tar.gz
cd percona-xtrabackup-2.1.5-Linux-x86_64/
cp bin/* /usr/bin/
```

#### innobackupex 使用参数介绍

由于 innobackupex 能同时备份 InnoDB 和 MyISAM 引擎的表, 这里重点介绍 innobackupex 的备份与恢复使用

通常一般都直接使用 innobackupex, 因为它能同时备份 InnoDB 和 MyISAM 引擎的表。要注意的是 my.cnf 里 datadir 这个参数是必须要指定的, xtrabackup\_55 是根据它去定位 innodb 数据文件的位置。

使用帮助: <http://www.percona.com/docs/wiki/percona-xtrabackup:start>

#### innobackupex 语法及参数说明

```
innobackup [--sleep=MS] [--compress[=LEVEL]] [--include=REGEXP] [--user=NAME]
[--password=WORD] [--port=PORT] [--socket=SOCKET] [--no-timest a mp]
[--ibbackup=IBBACKUP-BINARY] [--slave-info] [--stream=tar]
[--defaults-file=MY.CNF] [--databases=LIST] [--remote-host=HOSTNAME]
```

<http://www.linuxyw.com> Email:63780668@qq.com

---

```
BACKUP-ROOT-DIR
```

```
innobackup --apply-log [--use-memory=MB] [--uncompress] [--defaults-file=MY.CNF]
[--ibbackup=IBBACKUP-BINARY] BACKUP-DIR
```

```
innobackup --copy-back [--defaults-file=MY.CNF] BACKUP-DIR
```

各参数说明:

#### **--defaults-file**

指定 mysql 的配置文件 my.cnf 的位置, 如--defaults-file=/etc/my.cnf

如果不该参数, xtrabackup 将从依次从以下位置查找配置文件/etc/my.cnf、/etc/mysql/my.cnf、/usr/local/etc/my.cnf、~/my.cnf, 并读取配置文件中的[mysqld]和[xtrabackup]配置段。[mysqld]中只需要指定 datadir、innodb\_data\_home\_dir、innodb\_data\_file\_path、innodb\_log\_group\_home\_dir、innodb\_log\_files\_in\_group、innodb\_log\_file\_size6 个参数即可让 xtrabackup 正常工作

#### **--apply-log**

对 xtrabackup 的--prepare 参数的封装

#### **--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 将备份数据存储到进程服务器上, HOSTNAME 是远程 IP 地址;

#### **--stream=[tar]**

备份文件输出格式, tar 时使用 tar4ibd , 该文件可在 XtarBackup binary 文件中获得. 如果备份时有指定--stream=tar, 则 tar4ibd 文件所处目录一定要在\$PATH 中(因为使用的是 tar4ibd 去压缩, 在 XtraBackup 的 binary 包中可获得该文件)。

在使用参数 stream=tar 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的话 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 配置文件中指定的临时目录 tmpdir

#### **--redo-only --apply-log 组,**

<http://www.linuxyw.com> Email:63780668@qq.com

---

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 实例使用的内存量

`--throttle=IOS`

同 xtrabackup 的 `--throttle` 参数

`--sleep=`是给 ibbackup 使用的, 指定每备份 1M 数据, 过程停止拷贝多少毫秒, 也是为了在备份时尽量减小对正常业务的影响, 具体可以查看 ibbackup 的手册 ;

`--compress[=LEVEL]`

对备份数据进行压缩, 仅支持 ibbackup, xtrabackup 还没有实现;

`--include=REGEXP`

对 xtrabackup 参数 `--tables` 的封装, 也支持 ibbackup。备份包含的库表, 例如:  
`--include="test.*"`, 意思是要备份 test 库中所有的表。如果需要全备份, 则省略这个参数; 如果需要备份 test 库下的 2 个表: test1 和 test2, 则写成:  
`--include="test.test1|test.test2"`。也可以使用通配符, 如: `--include="test.test*"`。

`--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 表的 database 都会被备份;

`--uncompress`

解压备份的数据文件, 支持 ibbackup, xtrabackup 还没有实现该功能;

`--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, 这里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.

### 操作:

对数据库全库备份:

```
innobackupex --defaults-file=/etc/my.cnf --user=root --password=111111
--port=3306 /tmp/
```

### 解释:

`--defaults-file=/etc/my.cnf` #指定 my.cnf 位置

<http://www.linuxyw.com> Email:63780668@qq.com

--user=root #指定 mysql 帐号  
--password=111111 #指定 mysql 密码  
--port=3306 #指定 mysql 端口  
/tmp/ #指定备份好的文件存放目录，我这放在/tmp/下面，  
执行完成后，如下多了个时间为名的目录，里面就是备份文件了：

```
[root@drfdai-17 tmp]# ls
2013-10-29_16-06-41
[root@drfdai-17 tmp]# ls 2013-10-29_16-06-41/
aa                                ibdata1                                test
xtrabackup_checkpoints
backup-my.cnf  mysql                                xtrabackup_binary                       xtrabackup_logfile
drfdai        performance_schema  xtrabackup_binlog_info
```

```
[root@drfdai-17 tmp]# ls
2013-10-29_16-06-41  mysql.sock
[root@drfdai-17 tmp]# ls 2013-10-29_16-06-41/
aa                                ibdata1                                test                                xtrabackup_checkpoints
backup-my.cnf  mysql                                xtrabackup_binary                       xtrabackup_logfile
drfdai        performance_schema  xtrabackup_binlog_info
```

#### 备份 drfdai 库：

```
innobackupex --defaults-file=/etc/my.cnf --user=root --password=111111
--port=3306 --database=drfdai /tmp/
```

#### 解释：

--database=drfdai #指定需要备份的数据库名

```
[root@drfdai-17 2013-10-29_16-23-53]# ls
backup-my.cnf  ibdata1                                xtrabackup_binlog_info  xtrabackup_logfile
drfdai        xtrabackup_binary  xtrabackup_checkpoints
```

```
[root@drfdai-17 2013-10-29_16-23-53]# ls
backup-my.cnf  ibdata1                                xtrabackup_binlog_info  xtrabackup_logfile
drfdai        xtrabackup_binary  xtrabackup_checkpoints
```

#### 备份多个库：

```
innobackupex --defaults-file=/etc/my.cnf --user=root --password=111111
--port=3306 --database='drfdai mysql' /tmp/
```

#### 解释：

--database='drfdai mysql' #指定你需要备份的那几个库名，用单引号把这些库名引起来，每个库中间用空格隔开。

```
[root@drfdai-17 2013-10-29_16-33-09]# ls
backup-my.cnf  ibdata1  xtrabackup_binary                       xtrabackup_checkpoints
drfdai        mysql    xtrabackup_binlog_info  xtrabackup_logfile
```

```
[root@drfdai-17 2013-10-29_16-33-09]# ls
backup-my.cnf  ibdata1  xtrabackup_binary                       xtrabackup_checkpoints
drfdai        mysql    xtrabackup_binlog_info  xtrabackup_logfile
```



### 备份多个表:

```
innobackupex --defaults-file=/etc/my.cnf --user=root --password=111111  
--port=3306 --database='drfdai.aal drfdai.aa2' /tmp/
```

#### 解释:

`--database='drfdai.aal drfdai.aa2'` #指定备份 drfdai 库下面的 aal 表和 aa2 表

### 还原操作:

停止 mysql 数据库并删除欲恢复的数据库文件夹

如我的 mysql 数据库文件夹是在 /data/mysql/data 中, 所以我删掉这个文件夹

然后重建一个 /data/mysql/data/

```
rm -rf /data/mysql/data  
mkdir /data/mysql/data
```

### 还原完整备份:

```
innobackupex --defaults-file=/etc/my.cnf --user=root --password=111111  
--port=3306 --apply-log /tmp/2013-10-29_17-17-47/
```

这里的 `--apply-log` 指明是将日志应用到数据文件上, 完成之后将备份文件中的数据恢复到数据库中

```
innobackupex --defaults-file=/etc/my.cnf --user=root --password=111111  
--port=3306 --copy-back /tmp/2013-10-29_17-17-47/
```

这里的一 `copy-back` 指明是进行数据恢复。数据恢复完成之后, 需要修改相关文件的权限 mysql 数据库才能正常启动。

```
chown -R mysql:mysql /data/mysql/data
```

### 增量备份:

#### 次先进行完备:

```
innobackupex --defaults-file=/etc/my.cnf --user=root --password=111111  
--port=3306 /tmp/
```

备份完后, 会生成最新的目录, 名为: 2013-10-29\_17-17-47

#### 进行第一次增量备份:

```
innobackupex --defaults-file=/etc/my.cnf --user=root --password=111111  
--port=3306 --incremental --incremental-basedir=/tmp/2013-10-29_17-17-47 /tmp
```

备份完成后, 会生成最新增量备份, 名为: 2013-10-29\_17-57-21

#### 解释:

`--incremental` #指定为此次为增量备份

<http://www.linuxyw.com> Email:63780668@qq.com

---

`--incremental-basedir=/tmp/2013-10-29_17-17-47` #指定是从哪个备份目录为参考点进行增量备份

#### 进行第二次增量备份:

```
innobackupex --defaults-file=/etc/my.cnf --user=root --password=111111
--port=3306 --incremental --incremental-basedir=/tmp/2013-10-29_17-57-21/ /tmp
```

以此类推，第三，第四 ……

#### 增量备份恢复:

增量备份恢复的步骤和完整备份恢复的步骤基本一致，只是应用日志的过程稍有不同。增量备份恢复时，是先将所有的增量备份挨个应用到完整备份的数据文件中，然后再将完整备份中的数据恢复到数据库中。命令如下：

#### 应用第一个增量备份

```
innobackupex --user=root --password=MYSQLPASSWORD --defaults-file=/etc/my.cnf
--apply-log /mysqlbackup/full/2011-08-09_14-50-20/ --incremental-dir=/mysqlbackup/trn/2011-08-09_15-12-43/
```

#### 应用第二个增量备份

```
innobackupex --user=root --password=MYSQLPASSWORD --defaults-file=/etc/my.cnf
--apply-log /mysqlbackup/full/2011-08-09_14-50-20/ --incremental-dir=/mysqlbackup/trn/2011-08-05_15-15-47/
```

#### 将完整备份中的数据恢复到数据库中。

```
innobackupex --user=root --password=MYSQLPASSWORD --defaults-file=/etc/my.cnf
--copy-back /mysqlbackup/full/2011-08-05_14-50-20/
```

其中，`--incremental-dir` 指定要恢复的增量备份的位置。

更多资料:

<http://blog.chinaunix.net/uid-20682026-id-3319204.html>

<http://database.5lcto.com/art/201108/283254.htm>