NSD DBA1 DAY05

1 数据备份与恢复

1.1 问题

本案例要求熟悉MySQL的备份与恢复，完成以下任务操作：

- 逻辑备份工具 mysqldump
- 使用mysql 恢复数据库

1.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：使用mysqldump进行逻辑备份

1）备份MySQL服务器上的所有库

将所有的库备份为mysql-all.sql文件：

```
[root@dbsvr1 ~]# mysqldump -u root -p --all-databases > /root/alldb.sql
Enter password:                                    //验证口令
[root@dbsvr1 mysql]# file /root/alldb.sql          //确认备份文件类型
/root/alldb.sql: UTF-8 Unicode English text, with very long lines
```

查看备份文件alldb.sql的部分内容：

```
[root@dbsvr1 ~]# grep -vE '^/|^-|^$' /root/alldb.sql | head -15
CREATE DATABASE /*!32312 IF NOT EXISTS*/ `home` /*!40100 DEFA
USE `home`;
DROP TABLE IF EXISTS `biao01`;
CREATE TABLE `biao01` (
  `id` int(2) NOT NULL,
  `name` varchar(8) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
LOCK TABLES `biao01` WRITE;
UNLOCK TABLES;
DROP TABLE IF EXISTS `biao02`;
```

Top

```
12.  CREATE TABLE `biao02` (
13.    `id` int(4) NOT NULL,
14.    `name` varchar(8) DEFAULT NULL,
15.    PRIMARY KEY (`id`)
16.  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
17.  .. ..
```

注意：若数据库都使用MyISAM存储引擎，可以采用冷备份的方式，直接复制对应的数据库目录即可；恢复时重新复制回来就行。

2）只备份指定的某一个库

将userdb库备份为userdb.sql文件：

```
01.  [root@dbsvr1 ~]# mysqldump -u root -p userdb > userdb.sql
02.  Enter password:                          //验证口令
```

查看备份文件userdb.sql的部分内容：

```
01.  [root@dbsvr1 ~]# grep -vE '^/|^-|^$' /root/userdb.sql
02.  DROP TABLE IF EXISTS `stu_info`;
03.  CREATE TABLE `stu_info` (
04.    `name` varchar(12) NOT NULL,
05.    `gender` enum('boy','girl') DEFAULT 'boy',
06.    `age` int(3) NOT NULL
07.  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
08.  LOCK TABLES `stu_info` WRITE;
09.  .. ..
```

3）同时备份指定的多个库

同时备份mysql、userdb库，保存为mysql+userdb.sql文件：

```
01.  [root@dbsvr1 ~]# mysqldump -u root -p -B mysql  userdb > mysql+test+
02.  Enter password:                          //验证口令
```

Top

查看备份文件userdb.sql的部分内容：

```
[root@dbsvr1 ~]# grep '^CREATE DATA' /root/mysql+userdb.sql
CREATE DATABASE /*!32312 IF NOT EXISTS*/ `mysql` /*!40100 DEFA
CREATE DATABASE /*!32312 IF NOT EXISTS*/ `userdb` /*!40100 DEF
```

步骤二：使用mysql命令从备份中恢复数据库、表

以恢复userdb库为例，可参考下列操作。通常不建议直接覆盖旧库，而是采用建立新库并导入逻辑备份的方式执行恢复，待新库正常后即可废弃或删除旧库。

1）创建名为userdb2的新库

```
mysql> CREATE DATABASE userdb2;
Query OK, 1 row affected (0.00 sec)
```

2）导入备份文件，在新库中重建表及数据

```
[root@dbsvr1 ~]# mysql -u root -p userdb2 < /root/userdb.sql
Enter password:                          //验证口令
```

3）确认新库正常，启用新库

```
mysql> USE userdb2;                          //切换到新库
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT sn,username,uid,gid,homedir        //查询数据，确认可
    -> FROM userlist LIMIT 10;
+----+----------+-----+-----+----------------+
| sn | username | uid | gid | homedir        |
+----+----------+-----+-----+----------------+
|  1 | root     |   0 |   0 | /root          |
```

Top

```
| 2 | bin      |  1 |  1 | /bin           |
| 3 | daemon   |  2 |  2 | /sbin          |
| 4 | adm      |  3 |  4 | /var/adm       |
| 5 | lp       |  4 |  7 | /var/spool/lpd |
| 6 | sync     |  5 |  0 | /sbin          |
| 7 | shutdown |  6 |  0 | /sbin          |
| 8 | halt     |  7 |  0 | /sbin          |
| 9 | mail     |  8 | 12 | /var/spool/mail |
| 10 | operator | 11 |  0 | /root          |
+----+----------+-----+-----+----------------+
10 rows in set (0.00 sec)
```

4）废弃或删除旧库

```
mysql> DROP DATABASE userdb;
Query OK, 2 rows affected (0.09 sec)
```

2 使用binlog日志

2.1 问题

利用binlog恢复库表，要求如下：

- 启用binlog日志
- 创建db1库tb1表，插入3条记录
- 删除tb1表中刚插入的3条记录
- 使用mysqlbinlog恢复删除的3条记录

2.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：启用binlog日志

1）调整/etc/my.cnf配置，并重启服务

```
[root@dbsvr1 ~]# vim  /etc/my.cnf
[mysqld]
.. ..
```

Top

```
04.  log-bin-index=mysql-bin                    //启用二进制日志，并指定
05.  server_id=1
06.  binlog_format=STATEMENT
07.  //在Mysql5.7中，binlog日志格式默认为ROW，但它不记录sql语句上下
08.  .. ..
09.  [root@dbsvr1 ~]# systemctl  restart mysqld.service
```

2）确认binlog日志文件

新启用binlog后，每次启动MySQl服务都会新生成一份日志文件：

```
01.  [root@dbsvr1 ~]# ls /var/lib/mysql/mysql-bin.*
02.  /var/lib/mysql/mysql-bin.000001  /var/lib/mysql/mysql-bin.index
```

其中mysql-bin.index文件记录了当前保持的二进制文件列表：

```
01.  [root@dbsvr1 ~]# cat /var/lib/mysql/mysql-bin.index
02.  ./mysql-bin.000001
```

重启MySQL服务程序，或者执行SQL操作“FLUSH LOGS;”，会生成一份新的日志：

```
01.  [root@dbsvr1 ~]# ls /var/lib/mysql/mysql-bin.*
02.  /var/lib/mysql/mysql-bin.000001  /var/lib/mysql/mysql-bin.index
03.  /var/lib/mysql/mysql-bin.000002
04.
05.  [root@dbsvr1 ~]# cat /var/lib/mysql/mysql-bin.index
06.  ./mysql-bin.000001
07.  ./mysql-bin.000002
```

步骤二：利用binlog日志重做数据库操作

1）执行数据库表添加操作

Top

创建db1·库tb1表，表结构自定义：

```
mysql> CREATE DATABASE db1;
Query OK, 1 row affected (0.05 sec)

mysql> USE db1;
Database changed
mysql> CREATE TABLE tb1(
    -> id int(4) NOT NULL,name varchar(24)
    -> );
Query OK, 0 rows affected (0.28 sec)
```

插入3条表记录：

```
mysql> INSERT INTO tb1 VALUES
    -> (1,'Jack'),
    -> (2,'Kenthy'),
    -> (3,'Bob');
Query OK, 3 rows affected (0.12 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

确认插入的表记录数据：

```
mysql> SELECT * FROM tb1;
+----+--------+
| id | name   |
+----+--------+
|  1 | Jack   |
|  2 | Kenthy |
|  3 | Bob    |
+----+--------+
3 rows in set (0.00 sec)
```

Top

2）删除前一步添加的3条表记录

执行删除所有表记录操作：

```
mysql> DELETE FROM tb1;
Query OK, 3 rows affected (0.09  sec)
```

确认删除结果：

```
mysql> SELECT * FROM tb1;
Empty set (0.00 sec)
```

步骤三：通过binlog日志恢复表记录

binlog会记录所有的数据库、表更改操作，所以可在必要的时候重新执行以前做过的一部分数据操作，但对于启用binlog之前已经存在的库、表数据将不适用。

根据上述“恢复被删除的3条表记录”的需求，应通过mysqlbinlog工具查看相关日志文件，找到删除这些表记录的时间点，只要恢复此前的SQL操作（主要是插入那3条记录的操作）即可。

1）查看mysql-bin.000002日志内容

```
[root@dbsvr1 ~]# mysqlbinlog /var/lib/mysql/mysql-bin.000002
/*!50530 SET @@SESSION.PSEUDO_SLAVE_MODE=1*/;
/*!50003 SET @OLD_COMPLETION_TYPE=@@COMPLETION_TYPE,COMP
DELIMITER /*!*/;
# at 4
#170412 12:05:32 server id 1  end_log_pos 123 CRC32 0x6d8c069c
# Warning: this binlog is either in use or was not closed properly.
ROLLBACK/*!*/;
BINLOG '
jKftWA8BAAAAdwAAAHsAAAABAAQANS43LjE3LWxvZwAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAACMp+1YEzgNAAgAEgAEBAQEEgAAXwAEGggAAAAI
AZwGjG0=
'/*!*/;
# at 123
#170412 12:05:32 server id 1  end_log_pos 154 CRC32 0x17f50164
# [empty]
# at 154
#170412 12:05:59 server id 1  end_log_pos 219 CRC32 0x4ba5a976
```

Top

```
SET @@SESSION.GTID_NEXT= 'ANONYMOUS'/*!*/;
# at 219
#170412 12:05:59 server id 1  end_log_pos 310 CRC32 0x5b66ae13
SET TIMESTAMP=1491969959/*!*/;
SET @@session.pseudo_thread_id=3/*!*/;
SET @@session.foreign_key_checks=1, @@session.sql_auto_is_null=0,
SET @@session.sql_mode=1436549152/*!*/;
SET @@session.auto_increment_increment=1, @@session.auto_increme
/*!\C utf8 *//*!*/;
SET @@session.character_set_client=33,@@session.collation_connecti
SET @@session.lc_time_names=0/*!*/;
SET @@session.collation_database=DEFAULT/*!*/;
CREATE DATABASE db1
/*!*/;
# at 310
#170412 12:06:23 server id 1  end_log_pos 375 CRC32 0x2967cc28
SET @@SESSION.GTID_NEXT= 'ANONYMOUS'/*!*/;
# at 375
#170412 12:06:23 server id 1  end_log_pos 502 CRC32 0x5de09aae
use `db1`/*!*/;
SET TIMESTAMP=1491969983/*!*/;
CREATE TABLE tb1(
id int(4) NOT NULL,name varchar(24)
)
/*!*/;
# at 502
#170412 12:06:55 server id 1  end_log_pos 567 CRC32 0x0b8cd418
SET @@SESSION.GTID_NEXT= 'ANONYMOUS'/*!*/;
# at 567
#170412 12:06:55 server id 1  end_log_pos 644 CRC32 0x7e8f2fa0  C
SET TIMESTAMP=1491970015/*!*/;
BEGIN
/*!*/;
# at 644
#170412 12:06:55 server id 1  end_log_pos 772 CRC32 0x4e3f728e
SET TIMESTAMP=1491970015/*!*/;
INSERT INTO tb1 VALUES(1,'Jack'),(2,'Kenthy'), (3,'Bob')
```

Top

```
/*!*/;
# at 772
#170412 12:06:55 server id 1  end_log_pos 803 CRC32 0x6138b21f
                                        //确认事务的时间点
COMMIT/*!*/;
# at 803
#170412 12:07:24 server id 1  end_log_pos 868 CRC32 0xbef3f472  A
SET @@SESSION.GTID_NEXT= 'ANONYMOUS'/*!*/;
# at 868
#170412 12:07:24 server id 1  end_log_pos 945 CRC32 0x5684e92c
SET TIMESTAMP=1491970044/*!*/;
BEGIN
/*!*/;
# at 945
#170412 12:07:24 server id 1  end_log_pos 1032 CRC32 0x4c1c75fc
SET TIMESTAMP=1491970044/*!*/;
DELETE FROM tb1
/*!*/;
# at 1032
#170412 12:07:24 server id 1  end_log_pos 1063 CRC32 0xccf549b2
COMMIT/*!*/;
SET @@SESSION.GTID_NEXT= 'AUTOMATIC' /* added by mysqlbinlog */
DELIMITER ;
# End of log file
/*!50003 SET COMPLETION_TYPE=@OLD_COMPLETION_TYPE*/;
/*!50530 SET @@SESSION.PSEUDO_SLAVE_MODE=0*/;
```

2）执行指定Pos节点范围内的sql命令恢复数据

根据上述日志分析，只要恢复从2014.01.12 20:12:14到2014.01.12 20:13:50之间的操作即可。可通过mysqlbinlog指定时间范围输出，结合管道交给msyql命令执行导入重做：

```
[root@dbsvr1 ~]# mysqlbinlog \
   --start-datetime="2017-04-12 12:06:55" \
   --stop-datetime="2017-04-12 12:07:23" \
   /var/lib/mysql/mysql-bin.000002 | mysql -u root -p
```

Top

```
05.  Enter password:                          //验证口令
```

3）确认恢复结果

```
01.  mysql> SELECT * FROM db1.tb1;
02.  +----+--------+
03.  | id | name   |
04.  +----+--------+
05.  |  1 | Jack   |
06.  |  2 | Kenthy |
07.  |  3 | Bob    |
08.  +----+--------+
09.  3 rows in set (0.00 sec)
```

3 innobackupex备份工具

3.1 问题

- 安装percona软件包
- innobackupex完整备份、增量备份操作。
- 恢复数据

1.

3.2 步骤

实现此案例需要按照如下步骤进行。

步骤一：安装XtraBackup软件包

1）了解软件包描述信息

```
01.  [root@dbsvr1 pub]# rpm -qpi percona-xtrabackup-24-2.4.6-2.el7.x86_64
02.  Name        : percona-xtrabackup-24
03.  Version     : 2.4.6
04.  Release     : 2.el7
05.  Architecture: x86_64
06.  Install Date: (not installed)
07.  Group       : Applications/Databases
```

Top

```
08.  Size        : 32416340
09.  License     : GPLv2
10.  Signature   : DSA/SHA1, 2017年02月27日 星期一 20时28分17秒, Key
11.  Source RPM  : percona-xtrabackup-24-2.4.6-2.el7.src.rpm
12.  Build Date  : 2017年02月27日 星期一 20时27分21秒
13.  Build Host  : vps-centos7-x64-01.ci.percona.com
14.  Relocations : (not relocatable)
15.  URL         : http://www.percona.com/software/percona-xtrabackup
16.  Summary     : XtraBackup online backup for MySQL / InnoDB
17.  Description :
18.  Percona XtraBackup is OpenSource online (non-blockable) backup soluti
```

2）安装依赖包perl-DBD-MySQL perl-Digest-MD5 libev

使用RHEL 7自带的即可，yum方式安装：

```
01.  [root@dbsvr1 pub]# yum -y install perl-DBD-MySQL perl-Digest-MD5
02.  libev使用网上找的rpm包 libev-4.15-1.el6.rf.x86_64.rpm //该包由讲师提
03.  [root@dbsvr1 pub]#rpm –ivh libev-4.15-1.el6.rf.x86_64.rpm
```

如果未安装这些依赖包，则直接安装percona-xtrabackup时会报错：

```
                                                                    代码
```

3）安装percona-xtrabackup

```
01.  [root@dbsvr1 pub]#rpm -ivh percona-xtrabackup-*.rpm
02.  警告：percona-xtrabackup-24-2.4.6-2.el7.x86_64.rpm: 头V4 DSA/SHA1
03.  准备中...                          ##################################
04.  正在升级/安装...
05.     1:percona-xtrabackup-24-2.4.6-2.el7##########################
06.     2:percona-xtrabackup-test-24-2.4.6-##########################
07.     3:percona-xtrabackup-24-debuginfo-2##########################
                                                                    Top
```

4）确认安装的主要程序/脚本

```
[root@dbsvr1 pub]# rpm -ql percona-xtrabackup-24-2.4.6-2.el7.x86_64
/usr/bin/innobackupex
/usr/bin/xbcloud
/usr/bin/xbcloud_osenv
/usr/bin/xbcrypt
/usr/bin/xbstream
/usr/bin/xtrabackup
/usr/share/doc/percona-xtrabackup-24-2.4.6
/usr/share/doc/percona-xtrabackup-24-2.4.6/COPYING
/usr/share/man/man1/innobackupex.1.gz
/usr/share/man/man1/xbcrypt.1.gz
/usr/share/man/man1/xbstream.1.gz
/usr/share/man/man1/xtrabackup.1.gz
```

步骤二：innobackupex完整备份、增量备份操作

--host 主机名

--port 3306

--user 用户名

--password 密码

--databases="库名"

--databases="库1 库2"

--databases="库.表"

--no-timestamp 不用日期命名备份文件存储的子目录，使用备份的数据库名做备份目录名

--no-timestmap 不使用日期命名备份目录名

1）做一个完整备份

默认情况下，备份文件存储的子目录会用日期命名，

innobackupex作为客户端工具，以mysql协议连入mysqld，将数据备份到/backup文件夹：

Top

```
[root@dbsvr1 ~]# innobackupex --user=root --password=1234567 /back
170425 11:05:44 innobackupex: Starting the backup operation
```

```
03.
04.  IMPORTANT: Please check that the backup run completes successfully.
05.            At the end of a successful backup run innobackupex
06.            prints "completed OK!".
07.
08.  Unrecognized character \x01; marked by <-- HERE after <-- HERE near c
09.  170425 11:05:45 Connecting to MySQL server host: localhost, user: ro
10.  Using server version 5.7.17
11.  innobackupex version 2.4.6 based on MySQL server 5.7.13 Linux (x86_
12.  xtrabackup: uses posix_fadvise().
13.  xtrabackup: cd to /var/lib/mysql
14.  xtrabackup: open files limit requested 0, set to 1024
15.  xtrabackup: using the following InnoDB configuration:
16.  xtrabackup:   innodb_data_home_dir = .
17.  xtrabackup:   innodb_data_file_path = ibdata1:12M:autoextend
18.  xtrabackup:   innodb_log_group_home_dir = ./
19.  xtrabackup:   innodb_log_files_in_group = 2
20.  xtrabackup:   innodb_log_file_size = 50331648
21.  InnoDB: Number of pools: 1
22.  170425 11:05:45 >> log scanned up to (2543893)
23.  xtrabackup: Generating a list of tablespaces
24.  InnoDB: Allocated tablespace ID 2 for mysql/plugin, old maximum was 0
25.  170425 11:05:45 [01] Copying ./ibdata1 to /backup/ibdata1
26.  170425 11:05:45 [01]        ...done
27.  170425 11:05:46 [01] Copying ./mysql/plugin.ibd to /backup/mysql/p
28.  170425 11:05:46 [01]        ...done
29.  170425 11:05:46 [01] Copying ./mysql/servers.ibd to /backup/mysql/
30.  170425 11:05:46 [01]        ...done
31.  170425 11:05:46 [01] Copying ./mysql/help_topic.ibd to /backup/mys
32.  170425 11:05:46 [01]        ...done
33.  170425 11:05:46 >> log scanned up to (2543893)
34.  .. ..
35.  170425 11:06:00 [01] Copying ./sys/x@0024waits_global_by_latency.f
36.  170425 11:06:00 [01]        ...done
37.  170425 11:06:00 [01] Copying ./sys/session_ssl_status.frm to /backu
38.  170425 11:06:00 [01]        ...done
39.  170425 11:06:00 [01] Copying ./db1/db.opt to /backup/db1/db.opt
```

Top

```
170425 11:06:00 [01]        ...done
170425 11:06:00 [01] Copying ./db1/tb1.frm to /backup/db1/tb1.frm
170425 11:06:00 [01]        ...done
170425 11:06:00 Finished backing up non-InnoDB tables and files
170425 11:06:00 Executing FLUSH NO_WRITE_TO_BINLOG ENGINE LOG
xtrabackup: The latest check point (for incremental): '2543884'
xtrabackup: Stopping log copying thread.
.170425 11:06:00 >> log scanned up to (2543893)

170425 11:06:00 Executing UNLOCK TABLES
170425 11:06:00 All tables unlocked
170425 11:06:00 [00] Copying ib_buffer_pool to /backup/ib_buffer_pool
170425 11:06:00 [00]        ...done
170425 11:06:00 Backup created in directory '/backup/'
170425 11:06:00 [00] Writing backup-my.cnf
170425 11:06:00 [00]        ...done
170425 11:06:00 [00] Writing xtrabackup_info
170425 11:06:00 [00]        ...done
xtrabackup: Transaction log of lsn (2543884) to (2543893) was copied.
170425 11:06:01 completed OK
```

确认备份好的文件数据：

```
[root@dbsvr1 ~]#ls /backup/
backup-my.cnf  ib_buffer_pool  mysql     sys                xtrabackup_info
db1  ibdata1     performance_schema  xtrabackup_checkpoints  xtrabackup
```

2）做一个增量备份（基于前一步的完整备份）

随意做一些新增或更改库表的操作，比如在db1库中新建一个mytb的表：

```
mysql> USE db1;
Database changed
mysql> CREATE TABLE mytb(id int(4), name varchar(24));
Query OK, 0 rows affected (0.38 sec)
```

Top

```
mysql> INSERT INTO tb1 VALUES
    -> (1,'bon'),
    -> (2,'bo'),
Query OK, 2 rows affected (0.12 sec)
Records: 2  Duplicates: 0  Warnings: 0
mysql> SELECT * FROM tb1;
+------+------+
| id   | name |
+------+------+
|    1 | bob  |
|    2 | bo   |
+------+------+
2 rows in set (0.00 sec)
```

以前一次保存到/backup的完整备份为基础，做一个增量备份，保存到/incr01/，指定增量备份参照的基本目录（完整备份目录）需要用到选项--incremental-basedir。相关操作如下：

```
[root@dbsvr1 ~]# innobackupex  --user=root --password=12345678 --in
170425 11:30:14 innobackupex: Starting the backup operation

IMPORTANT: Please check that the backup run completes successfully.
           At the end of a successful backup run innobackupex
           prints "completed OK!".

Unrecognized character \x01; marked by <-- HERE after <-- HERE near c
170425 11:30:14 Connecting to MySQL server host: localhost, user: ro
Using server version 5.7.17
innobackupex version 2.4.6 based on MySQL server 5.7.13 Linux (x86_6
incremental backup from 2543884 is enabled.
xtrabackup: uses posix_fadvise().
xtrabackup: cd to /var/lib/mysql
xtrabackup: open files limit requested 0, set to 1024
xtrabackup: using the following InnoDB configuration:
xtrabackup:   innodb_data_home_dir = .
xtrabackup:   innodb_data_file_path = ibdata1:12M:autoextend
```

Top

```
19.  xtrabackup:  innodb_log_group_home_dir = ./
20.  xtrabackup:  innodb_log_files_in_group = 2
21.  xtrabackup:  innodb_log_file_size = 50331648
22.  InnoDB: Number of pools: 1
23.  170425 11:30:14 >> log scanned up to (2549933)
24.  xtrabackup: Generating a list of tablespaces
25.  InnoDB: Allocated tablespace ID 2 for mysql/plugin, old maximum was 0
26.  xtrabackup: using the full scan for incremental backup
27.  170425 11:30:15 [01] Copying ./ibdata1 to /incr01/ibdata1.delta
28.  170425 11:30:15 [01]        ...done
29.  170425 11:30:15 >> log scanned up to (2549933)
30.  170425 11:30:15 [01] Copying ./mysql/plugin.ibd to /incr01/mysql/plu
31.  170425 11:30:15 [01]        ...done
32.  ... ...
33.  170425 11:30:35 Executing UNLOCK TABLES
34.  170425 11:30:35 All tables unlocked
35.  170425 11:30:35 [00] Copying ib_buffer_pool to /incr01/ib_buffer_poo
36.  170425 11:30:35 [00]        ...done
37.  170425 11:30:35 Backup created in directory '/incr01/'
38.  170425 11:30:35 [00] Writing backup-my.cnf
39.  170425 11:30:35 [00]        ...done
40.  170425 11:30:35 [00] Writing xtrabackup_info
41.  170425 11:30:35 [00]        ...done
42.  xtrabackup: Transaction log of lsn (2549924) to (2549933) was copied.
43.  170425 11:30:35 completed OK!
```

确认备份好的文件数据：

```
01.  [root@dbsvr1 ~]# ls /incr01/
02.  backup-my.cnf  ib_buffer_pool  ibdata1.meta  performance_schema  xtra
03.  db1            ibdata1.delta  mysql         sys
```

对比完整备份、增量备份的大小：

Top

```
01.  [root@dbsvr1 ~]# du -sh /backup/ /incr01/
```

```
02.    142M    /backup/                    //完整备份的大小
03.    3.5M    /incr01/                    //增量备份的大小
```

步骤三：恢复数据

通过XtraBackup工具备份的数据库目录，若要恢复到另一个MySQL服务器，需要先做一个“--apply-log --redo-only”的准备操作。

1）准备恢复“完整备份”

完成准备以后，最终/backup可用来重建MySQL服务器。这种情况下，需要先做一个“--apply-log --redo-only”的准备操作，以确保数据一致性：

```
01.  [root@dbsvr1 ~]#innobackupex  --user=root --password=12345678 --ap
02.  170425 11:42:19 innobackupex: Starting the apply-log operation
03.
04.  IMPORTANT: Please check that the apply-log run completes successfully
05.            At the end of a successful apply-log run innobackupex
06.            prints "completed OK!".
07.
08.  innobackupex version 2.4.6 based on MySQL server 5.7.13 Linux (x86_
09.  xtrabackup: cd to /backup/
10.  xtrabackup: This target seems to be already prepared.
11.  InnoDB: Number of pools: 1
12.  xtrabackup: notice: xtrabackup_logfile was already used to '--prepare'.
13.  xtrabackup: using the following InnoDB configuration for recovery:
14.  xtrabackup:   innodb_data_home_dir = .
15.  xtrabackup:   innodb_data_file_path = ibdata1:12M:autoextend
16.  xtrabackup:   innodb_log_group_home_dir = .
17.  xtrabackup:   innodb_log_files_in_group = 2
18.  xtrabackup:   innodb_log_file_size = 50331648
19.  xtrabackup: using the following InnoDB configuration for recovery:
20.  xtrabackup:   innodb_data_home_dir = .
21.  xtrabackup:   innodb_data_file_path = ibdata1:12M:autoextend
22.  xtrabackup:   innodb_log_group_home_dir = .
23.  xtrabackup:   innodb_log_files_in_group = 2
24.  xtrabackup:   innodb_log_file_size = 50331648                    Top
25.  xtrabackup: Starting InnoDB instance for recovery.
26.  xtrabackup: Using 104857600 bytes for buffer pool (set by --use-memor
```

```
InnoDB: PUNCH HOLE support available
InnoDB: Mutexes and rw_locks use GCC atomic builtins
InnoDB: Uses event mutexes
InnoDB: GCC builtin __atomic_thread_fence() is used for memory barrier
InnoDB: Compressed tables use zlib 1.2.7
InnoDB: Number of pools: 1
InnoDB: Not using CPU crc32 instructions
InnoDB: Initializing buffer pool, total size = 100M, instances = 1, chunk
InnoDB: Completed initialization of buffer pool
InnoDB: page_cleaner coordinator priority: -20
InnoDB: Highest supported file format is Barracuda.

xtrabackup: starting shutdown with innodb_fast_shutdown = 1
InnoDB: Starting shutdown...
InnoDB: Shutdown completed; log sequence number 2544177
InnoDB: Number of pools: 1
170425 11:42:20 completed OK!
```

准备恢复“增量备份”

```
[root@dbsvr1 ~]#innobackupex  --user=root --password=12345678 --ap
170425 11:42:55 innobackupex: Starting the apply-log operation

IMPORTANT: Please check that the apply-log run completes successfully
           At the end of a successful apply-log run innobackupex
           prints "completed OK!".

innobackupex version 2.4.6 based on MySQL server 5.7.13 Linux (x86_
incremental backup from 2543884 is enabled.
xtrabackup: cd to /backup/
xtrabackup: This target seems to be already prepared with --apply-log-on
InnoDB: Number of pools: 1
xtrabackup: xtrabackup_logfile detected: size=8388608, start_lsn=(254
xtrabackup: using the following InnoDB configuration for recovery:
xtrabackup:   innodb_data_home_dir = .
xtrabackup:   innodb_data_file_path = ibdata1:12M:autoextend
```

Top

```
xtrabackup:   innodb_log_group_home_dir = /incr01/
xtrabackup:   innodb_log_files_in_group = 1
xtrabackup:   innodb_log_file_size = 8388608
xtrabackup: Generating a list of tablespaces
InnoDB: Allocated tablespace ID 2 for mysql/plugin, old maximum was 0
xtrabackup: page size for /incr01//ibdata1.delta is 16384 bytes
Applying /incr01//ibdata1.delta to ./ibdata1...
... ...
170425 11:43:09 [01] Copying /incr01/performance_schema/global_s
170425 11:43:09 [01]        ...done
170425 11:43:09 [01] Copying /incr01/performance_schema/session_
170425 11:43:09 [01]        ...done
170425 11:43:09 [00] Copying /incr01//xtrabackup_info to ./xtrabacku
170425 11:43:09 [00]        ...done
170425 11:43:10 completed OK!
```

2）关闭mysql服务，并将/var/lib/mysql/下的文件删除，假设数据被删除。

```
[root@dbsvr1 ~]#systemctl  stop  mysqld
[root@dbsvr1 ~]#rm -rf /var/lib/mysql
```

3）恢复“完整备份+增量备份”

完成准备以后，最终仍然是/backup用来重建MySQL服务器，但这种情况下需提前合并相关增量备份的数据

```
[root@dbsvr1 ~]# innobackupex --user=root --password=12345678 --co
... ...
170425 11:51:39 [01] Copying ./performance_schema/global_status.f
170425 11:51:39 [01]        ...done
170425 11:51:39 [01] Copying ./performance_schema/session_status
170425 11:51:39 [01]        ...done
170425 11:51:39 [01] Copying ./ib_buffer_pool to /var/lib/mysql/ib_b
170425 11:51:39 [01]        ...done
170425 11:51:39 [01] Copying ./ibtmp1 to /var/lib/mysql/ibtmp1
170425 11:51:39 [01]        ...done
```

Top

```
170425 11:51:39 [01] Copying ./xtrabackup_info to /var/lib/mysql/xtra
170425 11:51:39 [01]        ...done
170425 11:51:39 completed OK!
```

4）修改/var/lib/mysql/下文件属主与属组,查看数据：

恢复后，/var/lib/mysql下文件属组与属主皆为root，需要更改为mysql

```
[root@dbsvr1 ~]#chown -R mysql:mysql /var/lib/mysql
[root@dbsvr1 ~]#systemctl  start mysqld.service
[root@dbsvr1 ~]#mysql -uroot -p12345678 -e "select * from db1.tb1"
mysql: [Warning] Using a password on the command line interface can b
+------+------+
| id   | name |
+------+------+
|    1 | bob  |
|    2 | bo   |
+------+------+
```

Top