Percona Xtrabackup: Hot Backup Solution for MySQL

Martin Arrieta
February 2013
Agenda

1. Why backups?
2. Different options
3. Why Percona Xtrabackup
4. Installation
5. How it works?
6. Backup examples
7. Backup files
8. Restore process
9. Q & A
1. Why backups?

Really?
Do I have to talk about this?
2 - Different options
2.1 - Binary backups
2.2 - Logical backups
2.3 - MEB
2.1 - Binary backups

- **Cold copy (cp, rsync)**
  - Easy
  - Secure
  - Cold backup

- **LVM snapshots**
  - Hot Backup
  - Special setup required.
  - Performance penalty because of COW (Copy On Write)
2.2 - Logical backups

- **mysqldump**
  - Easy
  - Per-table or database backup
  - Hot Backup
  - Single thread
  - Restore process takes too long

- **mydumper**
  - Per-table or database backup
  - Hot Backup
  - Parallel
  - Restore process takes too long (less than mysqldump but it is slow any way)
2.3 - MEB

- MySQL Enterprise Backup
  - Similar to Percona Xtrabackup
  - Closed source
  - Annual subscription
3 - Why Percona Xtrabackup?
3 - Why Percona Xtrabackup?

• OpenSource
• Hot backup
• Non blocking for InnoDB and XtraDB tables.
• Incremental backups
• Fast restore
• Easy to use (we will see this later)
• Remote backups using streaming
• Many more...
4 - Installation
4.1 - RHEL and CentOS
4.2 - Debian based distributions
4.3 - Binary
4.1 - RHEL and CentOS

• Install the Percona repository:

$ rpm -Uvh http://www.percona.com/downloads/percona-release/percona-release-0.0-1.x86_64.rpm

• Install the package:

$ yum update && yum install percona-xtrabackup
4.2 - Debian based distributions

• Debian and Ubuntu packages from Percona are signed with a key. Before using the repository, you should add the key to apt. To do that, run the following commands:

```
$ gpg --keyserver hkp://keys.gnupg.net --recv-keys 1C4CBDCDCD2EFD2A
... [some output removed] ...
gpg:               imported: 1

$ gpg -a --export CD2EFD2A | sudo apt-key add -
```

• Add this to `/etc/apt/sources.list`, replacing VERSION with the name of your distribution:

```
deb http://repo.percona.com/apt VERSION main
deb-src http://repo.percona.com/apt VERSION main
```

• Install the package

```
$ apt-get update && apt-get install percona-xtrabackup
```
4.3 - Binary tar.gz

Download the binary package from the web:

```bash
$ tar cvzf percona-xtrabackup-2.0.5-499.tar.gz
```
5 - How it works?
5.1 - Package content
5.2 - The backup process
5.1 - Package content

• **xtrabackup** - a compiled C binary, which copies only InnoDB and XtraDB data

• **innobackupex** - a wrapper script that provides functionality to backup MyISAM tables and the rest of the database files.

• **xbstream** - utility that allows streaming and extracting files to/from the xbstream format.
## 5.2 - The backup process

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Start <code>xtrabackup_log</code></td>
</tr>
<tr>
<td>2</td>
<td>Copy <code>.ibd</code> files and <code>ibdata1</code></td>
</tr>
<tr>
<td>3</td>
<td>FLUSH TABLES WITH READ LOCK</td>
</tr>
<tr>
<td>4</td>
<td>Copy <code>.FRM</code>, <code>.MYD</code>, <code>MYI</code> files.</td>
</tr>
<tr>
<td>5</td>
<td>Get binary log position</td>
</tr>
<tr>
<td>6</td>
<td>UNLOCK TABLES</td>
</tr>
<tr>
<td>7</td>
<td>stop and copy <code>xtrabackup_log</code></td>
</tr>
</tbody>
</table>

**innobakcupex process**
# 5.2 - The backup process

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Start xtrabackup_log</td>
</tr>
<tr>
<td>2</td>
<td>Copy .ibd files and ibdata1</td>
</tr>
<tr>
<td>3</td>
<td>FLUSH TABLES WITH READ LOCK</td>
</tr>
<tr>
<td>4</td>
<td>Copy .FRM, .MYD, MYI files.</td>
</tr>
<tr>
<td>5</td>
<td>Get binary log position</td>
</tr>
<tr>
<td>6</td>
<td>UNLOCK TABLES</td>
</tr>
<tr>
<td>7</td>
<td>stop and copy xtrabackup_log</td>
</tr>
</tbody>
</table>

**innobakcupex process**
## 5.2 - The backup process

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Start xtrabackup_log</td>
</tr>
<tr>
<td>2</td>
<td>Copy .ibd files and ibdata1</td>
</tr>
<tr>
<td>3</td>
<td>FLUSH TABLES WITH READ LOCK</td>
</tr>
<tr>
<td>4</td>
<td>Copy .FRM, .MYD, MYI files.</td>
</tr>
<tr>
<td>5</td>
<td>Get binary log position</td>
</tr>
<tr>
<td>6</td>
<td>UNLOCK TABLES</td>
</tr>
<tr>
<td>7</td>
<td>stop and copy xtrabackup_log</td>
</tr>
</tbody>
</table>

**innobakcupex process**
5.2 - The backup process

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Start xtrabackup_log</td>
</tr>
<tr>
<td>2</td>
<td>Copy .ibd files and ibdata1</td>
</tr>
<tr>
<td>3</td>
<td><strong>FLUSH TABLES WITH READ LOCK</strong></td>
</tr>
<tr>
<td>4</td>
<td>Copy .FRM, .MYD, MYI files.</td>
</tr>
<tr>
<td>5</td>
<td>Get binary log position</td>
</tr>
<tr>
<td>6</td>
<td><strong>UNLOCK TABLES</strong></td>
</tr>
<tr>
<td>7</td>
<td>stop and copy xtrabackup_log</td>
</tr>
</tbody>
</table>

innobakcupex process
### 5.2 - The backup process

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Start <code>xtrabackup_log</code></td>
</tr>
<tr>
<td>2</td>
<td>Copy <code>.ibd</code> files and <code>ibdata1</code></td>
</tr>
<tr>
<td>3</td>
<td><code>FLUSH TABLES WITH READ LOCK</code></td>
</tr>
<tr>
<td>4</td>
<td>Copy <code>.FRM</code>, <code>.MYD</code>, <code>.MYI</code> files.</td>
</tr>
<tr>
<td>5</td>
<td>Get binary log position</td>
</tr>
<tr>
<td>6</td>
<td><code>UNLOCK TABLES</code></td>
</tr>
<tr>
<td>7</td>
<td>stop and copy <code>xtrabackup_log</code></td>
</tr>
</tbody>
</table>

The innobakcupex process is responsible for the backup operations.
5.2 - The backup process

1. Start xtrabackup_log
2. Copy .ibd files and ibdata1
3. FLUSH TABLES WITH READ LOCK
4. Copy .FRM, .MYD, MYI files.
5. Get binary log position
6. UNLOCK TABLES
7. stop and copy xtrabackup_log
5.2 - The backup process

1. Start xtrabackup_log
2. Copy .ibd files and ibdata1
3. FLUSH TABLES WITH READ LOCK
4. Copy .FRM, .MYD, MYI files.
5. Get binary log position
6. UNLOCK TABLES
7. stop and copy xtrabackup_log
## 5.2 - The backup process

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Start <code>xtrabackup_log</code></td>
</tr>
<tr>
<td>2</td>
<td>Copy <code>.ibd</code> files and <code>ibdata1</code></td>
</tr>
<tr>
<td>3</td>
<td><code>FLUSH TABLES WITH READ LOCK</code></td>
</tr>
<tr>
<td>4</td>
<td>Copy <code>.FRM</code>, <code>.MYD</code>, <code>.MYI</code> files.</td>
</tr>
<tr>
<td>5</td>
<td>Get binary log position</td>
</tr>
<tr>
<td>6</td>
<td><code>UNLOCK TABLES</code></td>
</tr>
<tr>
<td>7</td>
<td><code>stop and copy xtrabackup_log</code></td>
</tr>
</tbody>
</table>

`innobakcuplex` process
6 - Backup examples
6.1 - Local full backup
6.2 - Local incremental backup
6.3 - Remote backup
6.1 - Local full backup

$ innobackupex /backup/path

Other options are:
--defaults-file
--datadir (default assumption /var/lib/mysql)!
--user
--password
--host If you need to connected via specific IP
--socket
6.2 - Local incremental backup

**WARNING:** Only works for InnoDB and XtraDB engines!

To create an incremental backup, first we need a full backup.

```$ innobackupex backups/```

Then, we create an incremental backup from that backup, in this case was "backups/2013-02-06_21-17-36/"

```$ innobackupex --incremental=backups/ \ 
  --incremental-basedir=backups/2013-02-06_21-17-36/```

Options:

--incremental-basedir (Last backup)

--incremental Backup directory
6.2 - Local incremental backup

Takes the information between the previews LSN to the current one.

LSN: Acronym for **log sequence number**. This arbitrary, ever-increasing value represents a point in time corresponding to operations recorded in the redo log.

**How to check?**

```bash
$ cat backups/2013-02-06_21-17-36/xtrabackup_checkpoints
backup_type = full-backuped
from_lsn = 0
to_lsn = 370075856
last_lsn = 370075856

$ cat backups/2013-02-06_21-35-02/xtrabackup_checkpoints
backup_type = incremental
from_lsn = 370075856
to_lsn = 620749616
last_lsn = 620749616
```
6.3 - Remote backup

Options:

--compress (optional) compress the stream
--stream (type of stream to use, we are using xbstream)

$ innobackupex --compress --stream=xbstream ./ | ssh user@otherhost "xbstream -x -C /remote_backup"
7 - Backup files
7 - Backup files

• `xtrabackup_checkpoints`
  containing the LSN and the type of backup.

• `xtrabackup_binlog_info`
  containing the position of the binary log at the moment of backing up.

• `xtrabackup_binlog_pos_innodb`
  containing the position of the binary log at the moment of backing up relative to InnoDB transactions.

• `xtrabackup_slave_info`
  containing the MySQL binlog position of the master server, you have to run the backup with the option `--slave-info`.

• `backup-my.cnf`
  containing only the my.cnf options required for the backup, this is NOT a backup of your mysql configuration.

• `xtrabackup_binary`
  containing the binary version used for the backup.

• `mysql-stderr`
  containing the STDERR of mysqld during the process.

• `mysql-stdout`
  containing the STDOUT of the server.
8 - Restore process
8.1 - Restore a full backup
8.2 - Restore an incremental backup
8 - Restore process

- Similar to the InnoDB crash recovery.
- Will use the `backup-my.cnf` to start a mysql instance.
8.1 - Restore full backup

• To restore a full backup, you just have to apply the `xtrabackup_logfile` to the ibdata and *.ibd files.

```bash
$ innobackupex --apply-log backup/2013-02-06_21-17-36/
```

• To speed up the process, you can use the `--use-memory`, xtrabackup will use this amount of memory as a buffer pool to apply the logs (default is 100mb).

```bash
$ innobackupex --apply-log --use-memory=4G \ backup/2013-02-06_21-17-36/
```
8.1 - Restore an incremental backup

• If you replay the committed transactions and rollback the uncommitted ones on the base backup, you will not be able to add the incremental ones.

--redo-only  With this option, xtrabackup will apply only the redo-log.

```bash
$ innobackupex --apply-log --redo-only \
backup/2013-02-06_21-17-36/
```
8.2 - Restore an incremental backup

Then we have to apply the redo log of each incremental backup one by one to the base (first full) backup.

```bash
$ innobackupex --apply-log --redo-only backup/2013-02-06_21-17-36/ \\  --incremental-dir=backup/2013-02-07_21-17-36/
$ innobackupex --apply-log --redo-only backup/2013-02-06_21-17-36/ \\  --incremental-dir=backup/2013-02-08_21-17-36/
```

Once you have all together, you can prepare again the full backup (base + incrementals) once again to rollback the uncommitted transactions.

```bash
$ innobackupex --apply-log backup/2013-02-06_21-17-36/
```
Questions?