Highly Efficient Backups with Percona Xtrabackup

PLMCE-2015 (MySQL 101)

Valerii Kravchuk
Principal Support Engineer

Nilnandan Joshi
Support Engineer

Introduction

- **Percona Xtrabackup** provides a fully open-source, free, high-performance, non-blocking backup system for **InnoDB or XtraDB** tables
- It can also backup **MyISAM, Archive, Merge** and other SQL-level objects
- It is a reliable, widely-used alternative to Oracle's **MySQL Enterprise Backup**

Features

Open Source (GPL)	Encrypted backups	Point-in-time recovery support	Rsync support to minimize lock time
Cost: Free	Streaming backups	Safe slave backups	Improved FTWRL handling
Supported all MySQL flavors	Parallel local backups and compression	Compact backups	Backup history table
Non-blocking InnoDB backups*	Parallel encryption	Buffer pool state backups	Data & index file statistics
Incremental backups	Partial backups(with individual partition)	MySQL replication & Galera Support	Individual tables or partitions export
Full/Incremental compress backup	Full Binary Backups	Community Help/Support	Throttling

For more info: http://www.percona.com/doc/percona-xtrabackup/2.2/intro.html

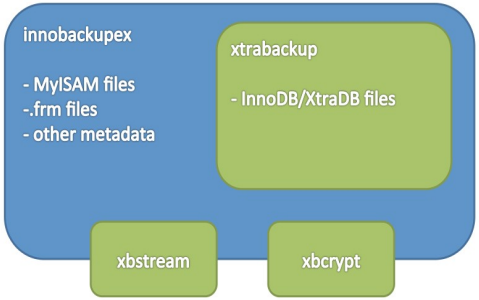
Compatibility

- **Version 2.1 and 2.2 are compatible with:**
  Oracle MySQL 5.1 (plugin version), 5.5, 5.6, corresponding versions of Percona Server, Percona XtraDB Cluster and MariaDB, including MariaDB 10.0.x.
- **Versions <= 2.0 are compatible with :**
  Oracle MySQL, 5.1, 5.5 and 5.6, equivalent versions of Percona Server and MariaDB.
- **MySQL 5.7 support is planned in upcoming 2.3, see**

https://bugs.launchpad.net/percona-xtrabackup/+bug/1437415

xtrabackup vs. innobackupex

Percona XtraBackup includes two main executables:



- **xtrabackup** is the low level binary that handles the creation and consistency checking of InnoDB backups.
- **innobackupex** is a higher-level Perl wrapper for xtrabackup and is also responsible for the rest of the operations, including the backup of other engines. You usually should use innobackupex.
- Starting with **Xtrabackup 2.3** the only official and recommended tool will be **xtrabackup**, functionality of **innobackupex** has already been rewritten in C and added to xtrabackup. 2.3 will be available "soon".

Installation

- The most straightforward way is to install from the Percona **apt** or **yum** repositories.

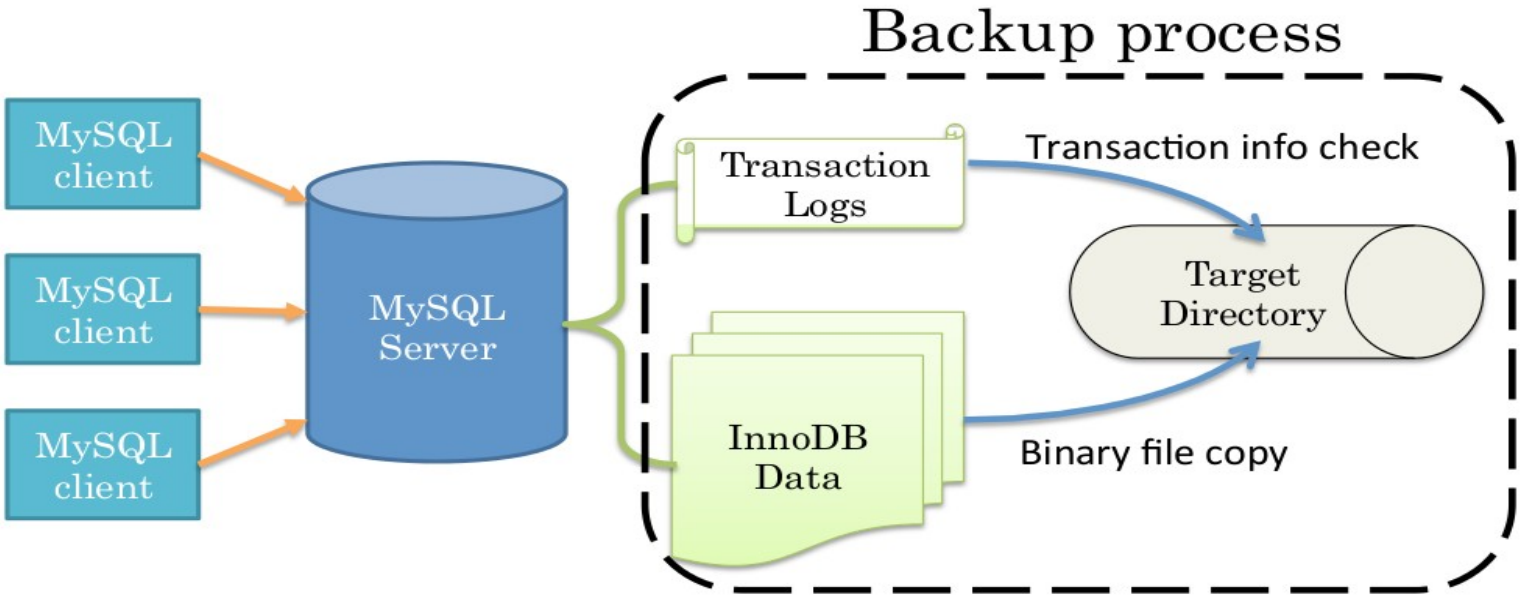
- Then install the package with:
  - **yum install percona-xtrabackup**
  - aptitude update && aptitude install percona-xtrabackup
  - **apt-get install percona-xtrabackup**

- Other installation options can be found at: **http://www.percona.com/downloads/XtraBackup/**

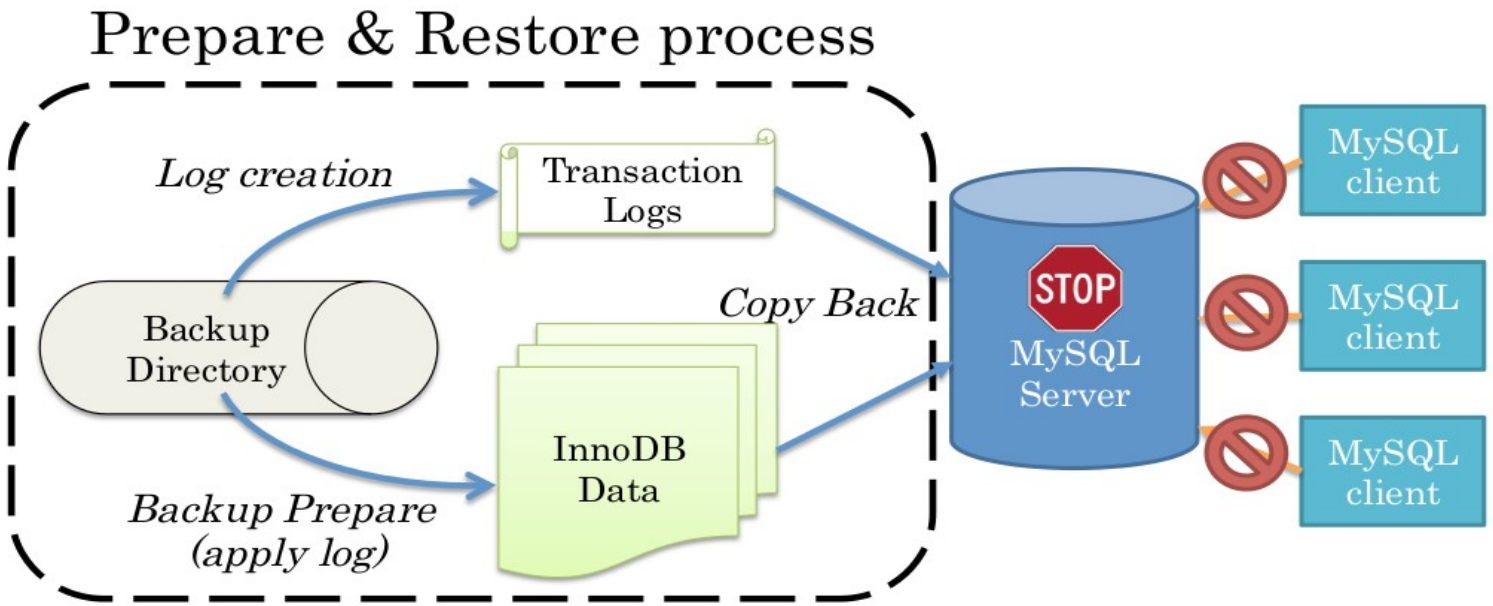
Requirements and Limitations

- Only fully supported on Linux machines
- **innobackupex** is a Perl script , Package should take care of all dependencies (Perl interpreter, mysql driver, other libraries)
- Local access to the MySQL datadir is required.
- **READ, WRITE (?) and EXECUTE** (?) filesystem privileges are needed.
- This makes xtrabackup incompatible with RDS or some web hosting providers.

How Percona XtraBackup Works: Full Backup



How Percona XtraBackup Works: Full Restore



Command Line Execution

- Minimal options for a full backup:
  # innobackupex /path/to/backup/

- Actual backup is stored in a timestamped subdirectory. It reads the datadir and innodb paths from my.cnf

- A more complete set of options:
  # innobackupex --host=localhost --user=root --password=secret /tmp

Files in Target Directory

- Hierarchy of files and directories mirroring the original database structure.
- **backup-my.cnf** : Not a backup of the server configuration. It only contains the minimal InnoDB settings at the time the backup to execute the --apply-log phase
- **xtrabackup_info:** Contains all informations about xtrabackup binary, server version, start and end time of backup, binlog position etc.
- **xtrabackup_checkpoints:** Metadata about the backup (type of backup, lsn, etc.)
- **xtrabackup_logfile:** Data needed for the –apply-log phase.

Preparation Phase

- Before the backup directory can be used, we must:
  - Make sure that the InnoDB tablespaces have consistent, non-corrupt data
  - Create new transaction logs

- **innobackupex** executes **xtrabackup** twice:
  - **First**, to do a controlled crash recovery, applying the log entries with xtrabackup's embedded InnoDB. This redoes all written committed transactions and undoes uncommitted ones.
  - **Second**, to create empty logs

Preparation Phase (cont)

- Only **xtrabackup/innobackupex** and the backup directory are needed
- It can be done on different machine than the original server (we have the xtrabackup binary and the innodb configuration parameters)
- Generally, you do it just before restore. If you only do full backups and want to minimize the **MTTR (Mean Time To Recover)**, you can prepare immediately after backup

Command Line Execution

- **The simplest form:**
  $ innobackupex --apply-log
  /home/nilnandan/backup/2015-03-03_12-14-55

- **You may specify a memory parameter:**
  $ innobackupex --apply-log --use-memory=4G
  /home/nilnandan/backup/2015-03-03_12-14-55

- **This increases the available buffer pool for the recovery phase (100M by default), to improve performance.**

Restore

- After **--apply-log**, we have a consistent full raw backup and we can just move it back to the datadir with standard OS commands once the server is shutdown.
- As a physical backup, restoration is as fast as you can copy the directory contents to the local filesystem
- There are specific commands for restoration in XtraBackup

Copy Back/Move Back

- The integrated command to restore a backup is:
  # innobackupex –copy-back /home/nilnandan/backup/2015-03-03_12-14-55

- It complains if the datadir, as read from the config file, is not empty [1] (it will not overwrite existing Data) Only partial restores can be done to a running server xtrabackup_* files are not needed.

  [1] We can override this behavior with the option --force-non-empty-directories, but it will never overwrite existing files

- As an alternative to **--copy-back**, we can also do:
  # innobackupex –move-back /home/nilnandan/backup/2015-03-03_12-14-55

- This may be faster or more convenient if the backup is already on the same partition or don't have enough free space.

Restoring Permissions

- After restore, as file permissions are preserved, you will probably have to fix file ownership:

  # chown -R mysql:mysql /var/lib/mysql

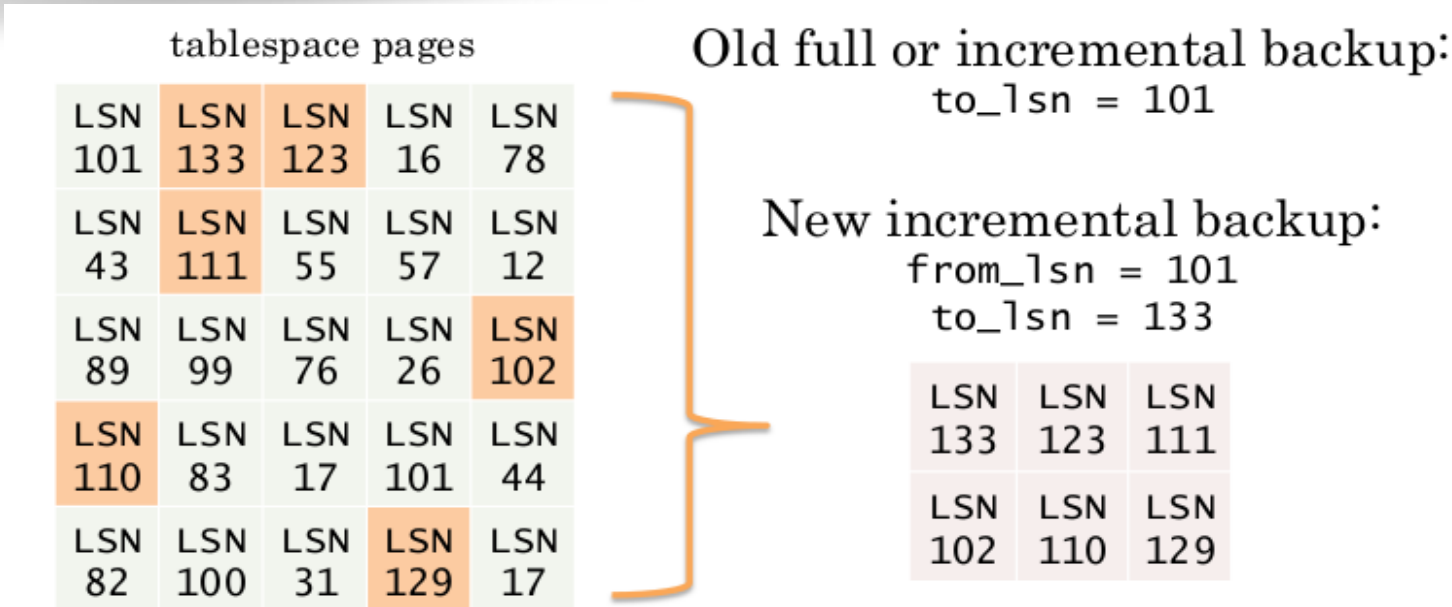
- If MySQL cannot write to the data directory or some of it files, it will refuse to start

Incremental Backup

- **Differential backup:** saves only the difference in the data since the last full backup
- **Incremental backup:** saves only the difference in the data since the last backup (full or another incremental)
- **Xtrabackup** can perform both, which can lead to an important save in **space** (and potentially, **time**), specially if the backups are very frequent or the database is very **static**.
- Differential and incremental backups work the same way with xtrabackup:
  - Every page stores the **Log Sequence Number (LSN)** when it was last changed
  - Only pages that have a higher LSN than the base backup (full or incremental) are copied
- With PS 5.6, one can set **innodb_track_changed_pages** variable which keep track of changed pages. It can be used to speed up incremental backups by removing the need to scan whole data files to find the changed pages. For more info

  http://www.percona.com/doc/percona-server/5.6/management/changed_page_tracking.html

Incremental Backup(cont..)



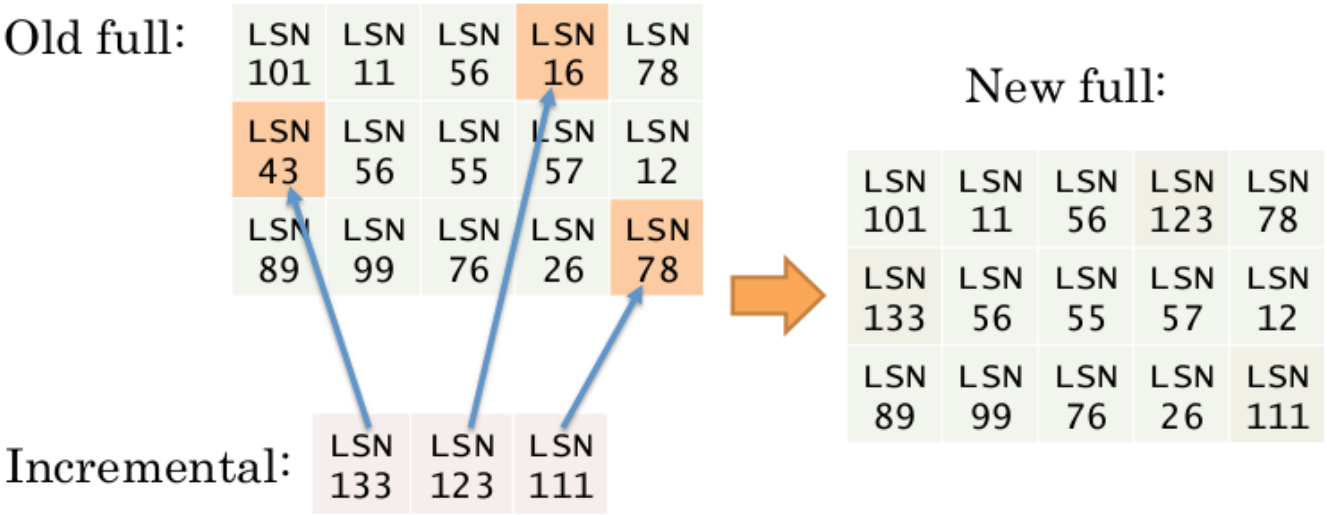

```
# innobackupex --incremental --incremental-basedir=/backups/last
/backups/incremental
```

Incremental Backup(cont..)

- **Restoration of Incremental Backups:**
  - On prepare time, the new pages have to be applied in order to the full backup before restore.
  - This is the reason why you want to wait to fully apply the log just before restore time
  - Uncommitted transactions must not be undone, as they may have finished on a subsequent incremental backup
  - Intermediate backups have to apply changes with the option **--redo-only**

Note : Only works for InnoDB tables, MyISAM tables still copied fully.

Incremental Backup(cont..)




```
# innobackupex --apply-log --redo-only
  /backups/full --incremental-dir=/backups/incremental
```

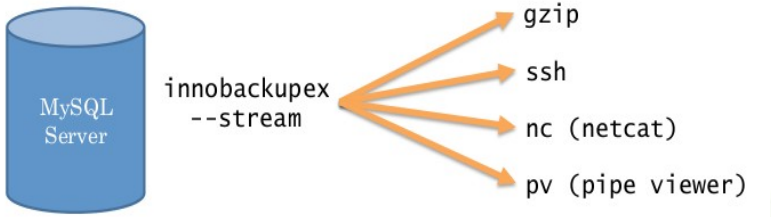
Partial Backups

- When using **xtrabackup** (or **innobackupex**), we can filter out some of the tables, effectively copying only some databases or tables.
- Two main restrictions apply for InnoDB tables:
  - Original database must use **innodb_file_per_table**, as xtrabackup works at filesystem level
  - Recovering individual tables is only possible for **MySQL 5.6.** previous versions require **Percona XtraDB** extensions with tablespace import enabled

- **Command Line Options for Partial Backups:**
  - **--include :** Matches the fully qualified name (database.table) against a regular expression
  - **--databases :** Matches a list of databases (or tables) separated by spaces.
  - **--tables-file :** Matches a list of tables, one per line,stored on the specified file

Streaming

- **Xtrabackup** allows the streaming of the backup files to the UNIX standard output. This allows to post-process them or send them to a remote location while the backup is ongoing.
- The stream option accepts two parameters:
  - **--stream=tar :** It sends all files to the standard output in the tar archive format (single file without compression
  - **--stream=xbstream :** The standard output will using the new included xbstream format, which allows simultaneous compression, encryption and parallelization. A temporary file path is still required



Streaming(Cont.)

- **Using tar:**

  # innobackupex --stream=tar /tmp > backup.tar

  # innobackupex --stream=tar /tmp | \ bzip2 - > backup.tar.bz2

  To extract, the tar -i option (ignore zeros) must be used:

  # tar -xivf backup.tar

  # tar -jxif backup.tar.bz2

Note: only GNU tar can extract archives created by xtrabackup.

- **Using xbstream:**

  # innobackupex --stream=xbstream /tmp > \ backup.xbstream

  To extract, use the xbstream binary provided with Percona Xtrabackup:

  # xbstream -x < backup.xbstream -C /root

Compression and xbstream

- We can use **--compress** to produce quick compressed files:

```
# innobackupex --compress /media/backups
```

- To decompress, we can use **--decompress** or **qpress*** to decompress the individual .qp files manually:

```
# innobackupex --decompress \ /media/backups/2017-07-08_13-22-12
# find . -name "*.qp" -execdir qpress -d \{\} . \;
```

- The **--compress** flag can be combined with xbstream (not with tar):

```
# innobackupex --compress --stream=xbstream /tmp \
> /media/backups/backup.xbstream
```

- Before preparation, we must use **first xbstream**, then **decompress**:

```
# xbstream -x -C /media/backups < backup.xbstream
# innobackupex --decompress /media/backups
```

Parallel Copy and Compression

- If using **xbstream**, we can use --parallel to archive several files at the same time:

  # innobackupex --parallel=4 --stream=xbstream /
  tmp > /media/backups/backup.tar

  Note: parallel option works with streaming as well as local backups but it doesn't work for TAR streaming.

- If we are using compression, we can parallelize the process for each file with –**compress-threads**

  **# innobackupex --compress --compress-threads=8 **
  --stream=xbstream /tmp > /media/backups/backup.tar

Remote Backups

- **Xtrabackup** can send automatically the backup files to a remote host by redirecting the output to applications like **ssh** or **netcat**:

```
# innobackupex --stream=tar /tmp \| ssh user@host "cat - >  /media/backups/backup.tar"
# ssh user@host "( nc -l 9999 > \ /media/backups/backup.tar & )" \
&& innobackupex --stream=tar /tmp \| nc host 9999
```

- **Throttling Remote Backups : pv (pipe viewer)** utility can be useful for both monitoring and limiting the bandwidth of the copy process. The following limits the transmission to 10MB/s:

```
# innobackupex --stream=tar ./ \| pv -q -L10m \| ssh user@desthost \
"cat - > /media/backups/backup.tar"
```

Compact Backups

- In order to reduce backup size, we can skip the copy of secondary index pages:

  # innobackupex --compact /media/backups

- The space saved will depend on the size of the secondary indexes. For example:

  #backup size without --compact : 2.0G 2015-02-01_10-18-38
  #backup size with --compact : 1.4G 2015-02-01_10-29-48

Compact Backups(cont.)

- The downside of compact backups is that the prepare phase will take longer, as the secondary keys have to be regenerated (**--rebuild-indexes**):

```
# cat xtrabackup_checkpoints
backup_type = full-backuped
from_lsn = 0
to_lsn = 9541454813
last_lsn = 9541454813
compact = 1
```

```
# innobackupex --apply-log --rebuild-indexes /media/backups/2017-07-08_13-22-12
```

- **Rebuilding Indexes:** The --rebuild-threads option can be used to speed up the recreation of the keys, by doing it in parallel:

```
#innobackupex --apply-log --rebuild-indexes --rebuild-threads=8 \
/media/backups/2017-07-08_13-22-12
```

- Rebuilding indexes has the side effect of defragmenting them, as they are done in order. We can rebuild regular full-backups, too.

Troubleshooting and Performance

- **Locking Issues:** By default, innobackupex will lock the tables (FTWRL) during the last phase of the backup. You can use –no-lock
  - if you are only backing up InnoDB tables and no DDL are executing on that last Phase*
  - if you backup other engines if no DDLs are executed, and non- innodb tables are not modified (e.g. users are created on the mysql database)
  - You can use --rsync to minimize the copy time when locked

  *** Please note that --no-lock also makes impossible to obtain a reliable binary log coordinates.**

- **File Not Found :** Set a correct datadir option on /etc/my.cnf

```
2017-07-08 12:56:56 7f0dfe4737e0 InnoDB: Operating system error number 2 in a file operation.
InnoDB: The error means the system cannot find the path specified.
2017-07-08 12:56:56 7f0dfe4737e0 InnoDB: File name ./ibdata1
2017-07-08 12:56:56 7f0dfe4737e0 InnoDB: File operation call: 'open' returned OS error 71.
2017-07-08 12:56:56 7f0dfe4737e0 InnoDB: Cannot continue operation.
innobackupex: Error: ibbackup child process has died at /usr/bin/ innobackupex line 389.
```

Troubleshooting and Performance

- **Permission Denied :** Use the Linux root user/sudo or any other account with read permissions on datadir and on target-dir.

```
InnoDB: read-write can't be opened in ./ibdata1 mode
xtrabackup: Could not open or create data files. [...]
xtrabackup: error: xb_data_files_init() failed with error code 1000
innobackupex: Error: ibbackup child process has died at /usr/bin/ innobackupex line 389.
innobackupex: Error: Failed to create backup directory /backups/
2017-07-08_12-19-01: Permission denied at /usr/bin/innobackupex line 389.
```

- **MySQL Does Not Start After Recovery :**

```
Starting MySQL server... FAILED
The server quit without updating PID file
```

- Did you change file permissions of the files to the mysql user?
- Did you remember to apply the log before restart? Failing to do so can also corrupt the backup.
- Did you restore an appropriate my.cnf file? Previous versions of MySQL (< 5.6) do not start if transaction log files have a different size than specified/default

Troubleshooting and Performance

- **The Backup Takes Too Long :** As MySQL performs physical backups, the speed should be close to a filesystem copy. If IO impact is not a concern, we can speed it up by:

  - **Using parallelization :** (--parallel , --compression-threads , --encrypt-threads , --rebuild-threads)
  - **Using compression :** (--compress) if the extra CPU compensates the lower bandwidth required
  - **Using incremental backups :** (specially with Percona extensions) to perform rolling full backups

Known Bugs

https://bugs.launchpad.net/percona-xtrabackup/+bug/1413044
https://bugs.launchpad.net/percona-xtrabackup/+bug/1192834
https://bugs.launchpad.net/percona-xtrabackup/+bug/1388122

- **To check of Report bug, you can visit:**

https://launchpad.net/percona-xtrabackup/+bugs

Valerii Kravchuk

Principal Support Engineer

http://mysqlentomologist.blogspot.com/

https://www.facebook.com/valerii.kravchuk

valerii.kravchuk@percona.com

Nilnandan Joshi

Support Engineer

http://nilinfobin.com/

nilnandan.joshi@percona.com