MySQL®
Backup & Recovery
Best Practices

A Percona Whitepaper

Ryan Lowe, Principal Consultant

June 17, 2014

Contents

Backup and recovery processes are a foundational piece of any application infrastructure. A well-tested backup and recovery system can be the difference between a minor outage and the end of a business. In this paper, we enumerate best practices for MySQL® backup and recovery systems.

Recovery Requirements Determine Backup Requirements

There are three important questions that define what kind of backup and recovery system an organization needs. These questions must be asked for each discrete set of data:[1]

- What is the Recovery Time Objective?[2]
- What is the Recovery Point Objective?[3]
- What Risks should be mitigated?

It is important to note at the outset that certain protections can be achieved using delayed replication slaves, RAID, and/or solutions like DRBD. However, these are not a substitute for proper backups.

Recovery Time Objective

The Recovery Time Objective (RTO) refers to the amount of time that may pass during a disruption before it exceeds the maximum allowable threshold specified in the Business Continuity Plan.[4]

The key question related to RTO is, “How quickly must the data on this system be restored?”

Recovery Point Objective

The Recovery Point Objective (RPO) is the duration of time and service level within which a business process must be stored after a disaster in order to avoid unacceptable consequences associated with a break in continuity.

The key question related to RPO is, “How much data can I lose?”

---

[1] http://www.druva.com/blog/2008/03/22/understanding-rpo-and-rto/

[2] http://en.wikipedia.org/wiki/Recovery_time_objective

[3] http://en.wikipedia.org/wiki/Recovery_point_objective

[4] http://en.wikipedia.org/wiki/Business_continuity_planning

Risk Management

Backups can help mitigate failure scenarios as well as satisfy legal or industry regulations such as:

- (Multiple) Host Failure
- (Multiple) DC Failure
- Data Corruption/Loss
- Legislation/Regulation

The question is, "Against what failure scenarios must the data be protected?"

Designing a Backup and Recovery System

Based on the answers to the above three questions, a plan can be devised to satisfy the RTO, RPO, and address whatever risks are applicable to the target environment.

It is important to stress that not all data should have the same requirements. For example, an HR database should generally have a much longer RTO than the database used for transaction processing. The cost of a 1 day HR database outage may be small compared to the cost of being unable to accept payments for 1 hour.

The following sections will discuss different types of backups and how they can be used to satisfy the identified requirements.

Binary Backups

A binary backup of MySQL typically refers to a copy made of the entire on-disk database. These are used to mitigate a single (or sometimes multiple) host failure and to build replication slaves. The recovery times for binary backups are generally much lower than for logical backups but also require a restoration of the entire MySQL instance.

Some binary backups used for MySQL include:

- mylvmbackup[5] - This uses Logical Volume Manager (LVM)[6] tools and MySQL commands to create a snapshot of the volume containing the MySQL data directory. It requires that LVM be set up on the appropriate volumes – which can be difficult on very large and/or busy systems – and there are

[5] http://www.lenzg.net/mylvmbackup/

[6] http://en.wikipedia.org/wiki/Logical_Volume_Manager_(Linux)

some performance degradations[7] that can occur on very busy systems during the snapshot process.

- Percona XtraBackup[8] - Percona XtraBackup is a non-blocking utility designed to back up InnoDB tables on busy systems. Although MyISAM and other tables can be backed up, those operations are blocking (which cannot be avoided).
- MySQL Enterprise Backup[9] - Although very similar in scope to Percona XtraBackup, MySQL Enterprise Backup is only available in select commercial MySQL editions.

Logical Backups

A logical backup generates SQL files that can be used to regenerate a database. The advantages of logical backups are that they can be:

- Easily used to restore a single table (or a subset of data)
- Generally restored across major and minor versions
- Parsed using standard UNIX tools to find targeted data
- Automatically replicated from a master to a slave (e.g., restoring a table)

On the other hand, restore times with logical backups can take orders of magnitude longer than a restore of a full binary backup. However, when performing partial restores, a logical backup may end up saving significant time.

Some representative logical backup tools for MySQL include:

- mysqldump[10] - A client utility installed with MySQL, this tool writes to STDOUT (typically redirected to a single file) on a table-at-a-time basis. It can take a significant amount of time to both backup and restore. Many organizations begin with this as their first backup implementation because it is the most straightforward and well documented. However, it does not scale to larger data sets or low RTOs.

---

[7] http://www.mysqlperformanceblog.com/2009/02/05/disaster-lvm-performance-in-snapshot-mode/

[8] http://www.percona.com/doc/percona-xtrabackup/

[9] http://www.mysql.com/products/enterprise/backup.html

[10] http://dev.mysql.com/doc/refman/5.6/en/mysqldump.html

- mydumper [11] - A multi-threaded implementation of mysqldump functionality, mydumper dumps table data in parallel for faster backup (and restore) times. Given that the best use cases for logical backups is restoring a subset of tables, this is a much more appropriate (although less widely deployed) tool for the job.

Logical backups are typically used in conjunction with binary backups. Binary backups are used to address host failure. Logical backups are used to address data corruption or the need to restore a subset of tables.

Binary Log Backups

So far, RTO has been addressed but RPO has not. Binlog backups specifically address RPO. A common starting point is for organizations to rotate binary logs once an hour and backup the not-currently-used binlogs to a remote location. Combining binlog backups with a binary backup that saves the current binlog position allows restoration up to the end of the most-recently-backed-up binary log which, in this example, provides a worst case RPO of one hour (i.e., one hour of data loss). Although this can be lowered to a cycle of 30-minutes or less, the more frequently this action is performed, the more it can potentially interfere with the normal operations of the database.

If your RPO is closer to real-time, there is an alternative approach. MySQL 5.6 provides a mysqlbinlog [12] tool with enhanced functionality. It is possible to "stream" [13] binary logs from a remote server, thus achieving a near-real-time recovery point. Percona Server 5.6 also supports binary log steaming as well as compression and encryption to a remote datacenter. Using binary log streaming with logical backups uploaded to the cloud ensures you have all the data you need in case of a datacenter failure.

Note that when complexity is added to a backup system, it often makes the recovery procedures more involved as well.

Offsite Storage

In order to protect against full data center failure, it is important to have backup storage offsite (if the data is important enough to warrant it). Many organizations

---

[11] https://launchpad.net/mydumper

[12] http://dev.mysql.com/doc/refman/5.6/en/mysqlbinlog.html

[13] http://www.mysqlperformanceblog.com/2012/01/18/backing-up-binary-log-files-with-mysqlbinlog/

choose to use cloud storage for offsite backup storage because it is wholly separate from their infrastructure and typically comes with a high service level agreement with respect to data durability.

Not all backups should be stored remotely. Recovery time includes the time it takes to ship the files from their remote locations. With cloud providers, this process may end up being the biggest time consumer in the recovery process.

A standard approach is to keep 1-3 days locally on the database server, 7 days locally within the datacenter, and 14 days at a remote facility.

Encryption

Whenever sensitive data leaves a secure environment, it should be encrypted. This is especially relevant for offsite storage that is not within your organization's control.

Encrypting data (usually done via gpg[14] or a similar alternative) negatively affects both backup and recovery times. Additionally, it adds complexity and potential failure conditions, such as a revoked key used to encrypt the data.

Recovery Testing

Backups are not a goal in and of themselves. Rather, the purpose of backups is to enable recovery in the event of a disaster. Too often, backups are tested at 3am during an emergency or outage. Instead, recovery procedures should be tested on a regular basis with best practices dictating testing at least once per quarter.

Recovery testing validates the correctness of the backup infrastructure and it also provides critical metrics on recovery times. At the very least, this process should be automated such that the backup gets fully restored and the tables checked for errors. Further automation can be done that sanity checks the data size, performs queries against the data, validates result sets, and checksums the data.[15]

In addition to the above, recovery testing can be achieved by periodically refreshing data in development/test environments with production backups.

Backup Types

In addition to the traditional full backup where the entire dataset is backed up, a number of alternatives exist. The most relevant to MySQL backups is the incremental backup. An incremental backup is a backup of everything that has

[14] http://www.gnupg.org/

[15] http://www.percona.com/doc/percona-toolkit/2.2/pt-table-checksum.html

changed since the last backup of any type (a binary log backup is a special case of an incremental backup).

For example, you may have a full backup on Sunday and then incremental backups throughout the week. This will make the backup each day (other than Sunday) smaller and shorter. However, the recovery time must include restoration of the full backup and application of the incremental backup for each day. This strategy is commonly used when the dataset is too large to back up, meaning it can take more than 24 hours to backup or storage space is not available to house multiple backups.

The main risks associated with incremental backups are:

- A single corrupt incremental backup may invalidate all the others
- Incremental backups typically negatively affect the RTO

A differential backup simply records the differences since your last full backup. The advantage of taking a differential backup is usually the space savings. Most databases have a lot of data that does not change from one backup to the next. Not copying this data into your backups can result in significantly smaller backups. In addition, depending on the backup tool used, a differential backup can be less labor intensive for the server. If a differential backup does not have to scan all of the data to determine what has changed, the differential backup process can be significantly more efficient.

Percona XtraBackup supports[16] both incremental and differential backups, as does MySQL Enterprise Backup.

Backup Management

A number of solutions are available that manage the scheduling and archiving of backups. Without commentary, some of the more popular solutions include:

- XtraBackup Manager[17]
- Zmanda[18]
- Symantec NetBackup[19]
- Amanda[20]

---

[16] http://www.percona.com/doc/percona-xtrabackup/xtrabackup_bin/incremental_backups.html
[17] https://code.google.com/p/xtrabackup-manager/
[18] http://www.zmanda.com/
[19] http://www.symantec.com/netbackup
[20] http://www.amanda.org/

Many backup management tools can be used to manage backups throughout the organization and are not limited to managing MySQL-specific backups.

Monitoring Backups

Backup software should alert when a backup fails and should not be ignored. Most MySQL backup software will return a well-defined exit code that can be trapped and should be sent to an alerting system. Backup freshness and success/failure should alert as well as recoveries.

An automated recovery testing system should warn when the recovery time does not meet the RTO (this should include the full time of retrieving the backup from storage until the mysqld instance is fully operational). Emergency recovery is rarely performed so it is common for a recovery process to take longer than expected.

Graphing

Graphing backup events and their duration may seem unnecessary, but it is a great way to provide regular feedback and visualization on the process. Perhaps more importantly, if your graphing system allows you to insert arbitrary data points, it is possible to overlay backup start and stop times on other graphs so the cascading effects can be observed. This can provide indicators that your backup is interfering with normal database operations, perhaps due to the additional disk input/output.

Another example is when backups are moved off of the local database server and there is a corresponding spike in network bandwidth indicated in your other graphs. In one particular instance, this author was able to correlate offsite backup transfer to firewall capacity being reached which resulted in application instability.

Without graphs, these problems can be very difficult to detect.

Retention Requirements

Finally, it is important to keep multiple copies of a system's backup. A single backup can never be counted on as uncorrupted and it may contain data that was maliciously altered (and that nobody noticed for an entire backup cycle).

A solid starting point would be to keep:

- 1-2 daily backups locally on the database server (space permitting)
- 7 daily and 4 weekly backups in the local datacenter
- Monthly backups in a remote location for 1 year
- Annual backups encrypted and compressed in a remote location forever

Legal or regulatory requirements may also dictate how long data must be archived.

Conclusion

There are many considerations when building backup and recovery systems. This is a project that can take anywhere from 1 day to 6 months, depending on the value of the data, cost of downtime, and recovery-time and recovery-point objectives. Although often overlooked, solid and well-practiced backup and recovery procedures can mean the difference between having a bad day and having to look for a new job.

Additional Resources

This paper represents a subset of backup and recovery strategies. It borrows heavily from existing works. Following are a number of books that can provide additional insight into both general and MySQL-specific backup and recovery strategies.

- Preston, Curtis. **Backup & Recovery.** O'Reilly Media, 2007.
- **Effective MySQL Backup and Recovery**. Oracle Press, 2012.
- Schwartz, Baron and Peter Zaitsev and Vadim Tkachenko. **High Performance MySQL: Optimization, Backups, and Replication.** O'Reilly Media, 2012.

Percona Can Help

Percona can help you choose, implement, and optimize the most appropriate MySQL backup and recovery solution for your MySQL ecosystem. If your current solution unexpectedly fails, we can facilitate your recovery with onsite, remote, or emergency consulting services. We can also help you take steps to prevent another occurrence. Every situation is unique and we will work with you to create the most effective solution for your business.

We also offer Percona Managed Services which includes two options for outsourcing your MySQL backup and recovery. The Percona Backup Service ensures that you backups run smoothly and reliably by outsourcing configuration, monitoring, management, and recovery to the experts at Percona. The Percona Remote DBA service includes the Percona Backup Service along with other critical daily database administration tasks. Both services are available to organizations worldwide with support for most versions of MySQL including Oracle® MySQL, Percona Server, and MariaDB®. We also support Percona XtraDB Cluster and other High Availability architectures and tools. Your MySQL servers may be installed in your data center, at a hosting provider, or a cloud-based MySQL service.

For more information about Percona services, call our sales department at (888) 316-9775 in the United States or +44-208-133-0309 in Europe.

About Percona

Percona has made MySQL faster and more reliable for over 2,000 customers worldwide since 2006. Percona provides enterprise-grade **MySQL Support**, **Consulting**, **Managed Services**, and **Server Development** services. Percona's founders authored the definitive book **High Performance MySQL** from O'Reilly Media and the widely read **MySQL Performance Blog**. Percona also develops software for MySQL users, including **Percona Server**, **Percona XtraBackup**, **Percona XtraDB Cluster**, and **Percona Toolkit**. The popular **Percona Live conferences** draw attendees and acclaimed speakers from around the world. For more information, visit **www.percona.com**.

For more information:

1-888-316-9775

or +1-208-473-2904

Percona

4125 Mohr Avenue, Suite H

Pleasanton, California 94566

United States of America