Montevideo, Uruguay | September 23, 2026



Steffano Britos
MySQL DBA at Percona

PERCONA UNIVERSITY

MySQL Replication Demystified



Steffano Britos

DBA I, ExpertOps – Percona

Professional Background

10+ years as MySQL DBA

Percona, Ministerio de Ambiente, Presidencia de la República, Intendencia de Canelones.

Bachelor of Science in Information Systems - ORT University. Uruguay.

Certifications

Red Hat Certified System Administrator

Canelones, Uruguay

Email
steffano.britos@percona.com

LinkedIn
www.linkedin.com/in/steffano-britos-2160661a

PERCONA

Agenda

01 Speaker Intro
Background, certifications, and experience.

02 Why Replicate
High Availability, Load Balancing, and Use Cases.

03 Source-Replica
Binary logs, relay logs, and replication threads.

04 GTIDs & Topologies
Global Transaction IDs and multi-source setups.

05 Lag & Monitoring
Measuring replication lag and troubleshooting.

06 Wrap-up & Q&A
Key takeaways and open question session.

PERCONA®

Why We Replicate: Scale, HA, and Common Use Cases



High Availability

A replica acts as a hot spare if the source server crashes.



Load Balancing

Applications can offload read queries to replicas, scaling read capacity.



Backups & Dedicated Queries

Backups and dedicated queries can be performed on replicas without impacting production traffic.



Data Distribution & Testing

Geographically distributed read-only nodes and a safe environment to test configuration changes or version upgrades.

PERCONA®

The Source-Replica Pipeline

01 First, transactions commit on the source and are written to its binary log.

02 Second, the replica's IO_thread connects to the source, downloads the binary log events, and writes them locally to the relay log.

03 Third, the replica's SQL_thread reads the relay log and executes the transactions locally to keep the servers in sync.

04 Because replication is asynchronous, replicas can stop downloading or executing logs and seamlessly resume where they left off.

PERCONA

Binary Log Formats: Row-Based vs. Statement-Based

Statement-Based Replication (SBR)

- Logs actual SQL statements executed on the source
- Highly sensitive to data discrepancies between systems
- Non-deterministic functions can cause replica drift

Row-Based Replication (RBR)

- Logs the actual row image before and after changes
- Prevents replica drift and significantly reduces locking
- Results in larger binary log sizes due to detailed logging

Mixed-Mode Replication (MBR)

- Defaults to Statement format and automatically switches to Row format for statements deemed unsafe for replication.
- DDLs (Data Definition Language) are always logged in Statement format.

PERCONA

Replication Topology Setups

Simple Setups

Replication can be configured in various topologies to suit architectural needs.

Replication Chains

A replica acts as an intermediate source for another replica, requiring log_bin=1 and log_replica_updates=1. Intermediate replicas should use mixed-mode binary logging to support checksums.

Multi-Source Replication

Allows a single replica to be configured to replicate from multiple sources simultaneously.

Group Replication

For advanced high availability, provides multi-primary or single-primary setups utilizing Paxos consensus for conflict detection.

PERCONA

Understanding ACID Properties

Fundamental database guarantees ensuring reliability and transactional integrity, with a focus on Durability.

Atomicity

Ensures all operations in a transaction execute completely or roll back entirely ("all or nothing").

Consistency

Guarantees that database constraints are strictly enforced, moving data from one valid state to another.

Isolation (Independence)

Prevents concurrent transactions from interfering with each other, executing independently.

Durability (Key Focus)

Guarantees committed transactions persist on disk and survive OS or power failures.

PERCONA

Balancing Durability and Speed

Configuring database durability requires balancing maximum safety against write throughput.

Maximum Durability

```
sync_binlog = 1
innodb_flush_log_at_trx_commit = 1
```

- Ensures writes are flushed to disk on **every transaction commit**.
- Prevents data loss in the event of OS crashes or power failures.
- **Trade-off: Limits write IOPS performance.**

High Performance

```
sync_binlog = 0
innodb_flush_log_at_trx_commit = 2
```

- Flushes writes to disk **periodically** instead of instantly.
- Significantly increases database write speed and throughput.
- **Risk: Up to one second of data loss during a crash.**

PERCONA

Accelerating Playback: Improving Parallel Efficiency

Multi-threaded replicas improve parallel replication efficiency by executing events concurrently via multiple worker threads.
Variable: replica_parallel_type

DATABASE Parallelism

- Allows updates to different databases to run in parallel.
- **Ineffective if all tables reside in a single database.**

LOGICAL_CLOCK Parallelism

Introduced in MySQL 5.7

- Tags transactions with sequence_number and last_committed values.
- Allows dependent writes to execute concurrently as soon as their dependencies complete.
- Vastly improves playback speed for intra-schema workloads.

PERCONA

The Replica Read Bottleneck

Locking Conflicts

Heavy, long-running SELECT reporting queries on the replica create locking conflicts, blocking the SQL thread and stalling replication.

Primary Key Requirements

Row-Based Replication requires PKs for lookups. Missing keys trigger full table scans on every modified row, severely lagging replicas.

Lag Vectors

Concurrent analytical query workloads and missing keys compound, driving massive replication lag and impacting performance.

Performance Remediation

Isolate analytical workloads to dedicated replicas, enforce strict PK usage, and break up long-running transactions.

PERCONA

Diagnosing Lag & Replication Failures

Diagnosing and resolving replication failures is a core DBA task.

Common Failures & Remedies

- Skip common duplicate key errors (Error 1062) using: `pt-replica-restart --error-numbers 1062`.
- Always monitor for bottleneck vectors, including network latency, single-threaded execution bottlenecks, and disk I/O constraints on the replica.
- Is recommended to run pt-table-checksum

Observability in Action: Replication Monitoring

Percona Monitoring and Management (PMM) Integration

Leverage rich interactive dashboards for complete visualization of key database performance metrics, including **replication throughput**, **active threads**, **disk queues**, and **replication delay** to ensure comprehensive, real-time database observability.

Seconds_Behind_Source (Traditional Check)

Accessed easily via **SHOW REPLICA STATUS**, but introduces critical monitoring blindspots:

- Prone to severe inaccuracy during long-running transactions.
- Reports false zeros and fails to show delay during idle periods.

pt-heartbeat (High-Precision Tracking)

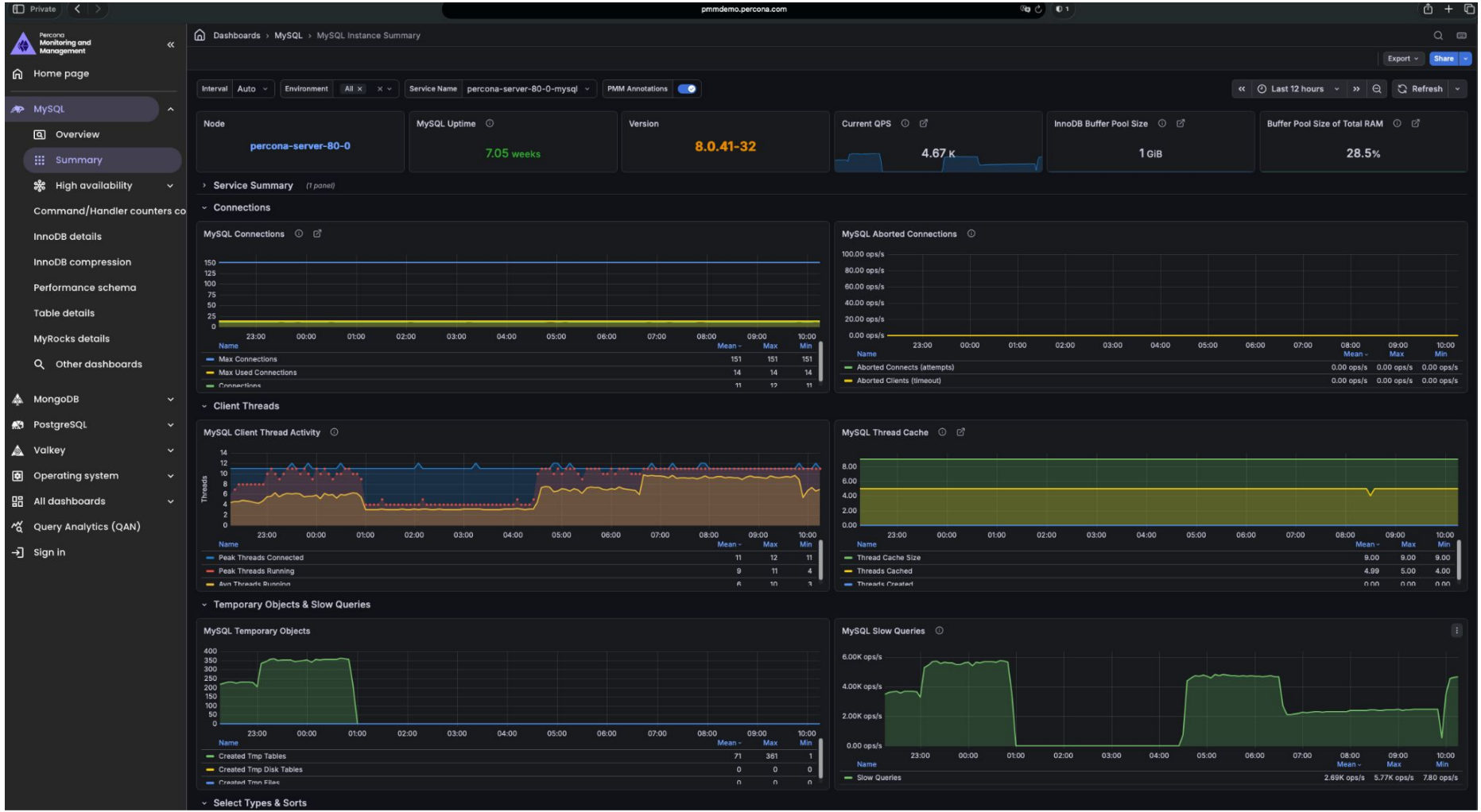
An advanced monitoring tool from Percona Toolkit that offers precise delay measuring:

- Writes a continuous, highly reliable heartbeat timestamp on the source.
- Monitors exact replication delay down to sub-second granularity.

PERCONA

Observability in Action: Replication Monitoring




Private
pmmdemo.percona.com
Percona Monitoring and Management
Home page
MySQL
Overview
Summary
High availability
Command/Handler counters co
InnoDB details
InnoDB compression
Performance schema
Table details
MyRocks details
Other dashboards
MongoDB
PostgreSQL
Valkey
Operating system
All dashboards
Query Analytics (QAN)
Sign in
Dashboards › MySQL › MySQL Instance Summary
Export
Share
Interval Auto
Environment All
Service Name percona-server-80-0-mysql
PMM Annotations
Last 12 hours
Refresh
Node
percona-server-80-0
MySQL Uptime
7.05 weeks
Version
8.0.41-32
Current QPS
4.67 K
InnoDB Buffer Pool Size
1 GiB
Buffer Pool Size of Total RAM
28.5%
Service Summary (1 panel)
Connections
MySQL Connections
Name Mean Max Min
Max Connections 151 151 151
Max Used Connections 14 14 14
Connections 11 12 11
MySQL Aborted Connections
Name Mean Max Min
Aborted Connects (attempts) 0.00 ops/s 0.00 ops/s 0.00 ops/s
Aborted Clients (timeout) 0.00 ops/s 0.00 ops/s 0.00 ops/s
Client Threads
MySQL Client Thread Activity
Threads
Name Mean Max Min
Peak Threads Connected 11 12 11
Peak Threads Running 9 11 4
Avg Threads Running 6 10 3
MySQL Thread Cache
Name Mean Max Min
Thread Cache Size 9.00 9.00 9.00
Threads Cached 4.99 5.00 4.00
Threads Created 0.00 0.00 0.00
Temporary Objects & Slow Queries
MySQL Temporary Objects
Name Mean Max Min
Created Tmp Tables 71 361 1
Created Tmp Disk Tables 0 0 0
Created Tmp Files 0 0 0
MySQL Slow Queries
Name Mean Max Min
Slow Queries 2.69K ops/s 5.77K ops/s 7.80 ops/s
Select Types & Sorts

Orchestrating High Availability

Semi-Sync Replication

Blocks the commit on the source until at least **N replicas** acknowledge receipt of the transaction to their relay log.

Controlled by rpl_semi_sync_source_timeout to manage graceful fallback.

Automatic Failover

Failover tools like **Orchestrator** continuously monitor clusters to detect sudden source failures.

Elects an optimal candidate replica, gracefully promotes it, and automatically repoints remaining replicas.

Topology Healing

Automatically resolves dangerous **split-brain** scenarios to guarantee strict data integrity.

Seamlessly recovers crashed replicas and reintegrates them back into the active topology without manual intervention.

Modern Replication: Key Advancements in MySQL 8.4 LTS

5-Year LTS Stability

As a Long-Term Support release, MySQL 8.4 is supported for 5 years with no feature additions or removals within its lifecycle.

LOGICAL_CLOCK Default

Modern multi-threaded replication now defaults to LOGICAL_CLOCK parallelism out-of-the-box, ensuring high performance for concurrent transactions.

Terminology Standardization

Complete migration to modern replication terminology, adopting the SOURCE and REPLICA syntax as the standard across the database ecosystem.

Looking Ahead: MySQL 9.0 Innovation Release

Transactional Purity

Restricts single transactions updating both InnoDB and non-transactional tables, eliminating the main cause of replication drift.

Component Semi-Sync

Transitioned from legacy plugins to Component Architecture, eliminating global lock bottlenecks and thread contention.

Strict Secure Auth

Removed mysql_native_password plugin. Replication channels strictly enforce caching_sha2_password authentication.

Enforcing transactional purity, component architecture, and modern secure authentication

PERCONA

Production Readiness Checklist

Before deploying replication in production, ensure strict adherence to best practices.

Validation & Transaction Limits

- Every table must have a Primary Key to maintain Row-Based replication performance.
- Avoid massive multi-gigabyte transactions; break them up to prevent excessive binary log generation and replica lag.

Maintenance & Drift Management

- Regularly run pt-table-checksum to detect replica drift and pt-table-sync to correct it without downtime.
- Finally, use SET SESSION SQL_LOG_BIN = 0 to execute local maintenance tasks on a replica without propagating them.

Q & A



Scan to Access Resources

Get direct access to presentation slides, technical documentation, and feedback.

Point your camera at the screen to open link instantly on mobile or tablet



PERCONA®