ORACLE

30 Years of MySQL: Scaling Smarter with ProxySQL

A Look at How ProxySQL Future-Proofs MySQL Deployments

René Cannaò
CEO
ProxySQL LLC
July 22 , 2025

Meet the Creator – René Cannaò

- Founder & CEO of ProxySQL
- Over 25 years of experience in database architecture & performance optimization
- Former MySQL consultant and performance expert
- Creator of ProxySQL to solve real-world MySQL scalability challenges
- Passionate about high-performance databases, open-source development, and community collaboration

- #OptimizingMySQLAtScale

Agenda

1 MySQL through the decades

2 Scaling problems and introduction to ProxySQL

3 Problems and how ProxySQL solves them

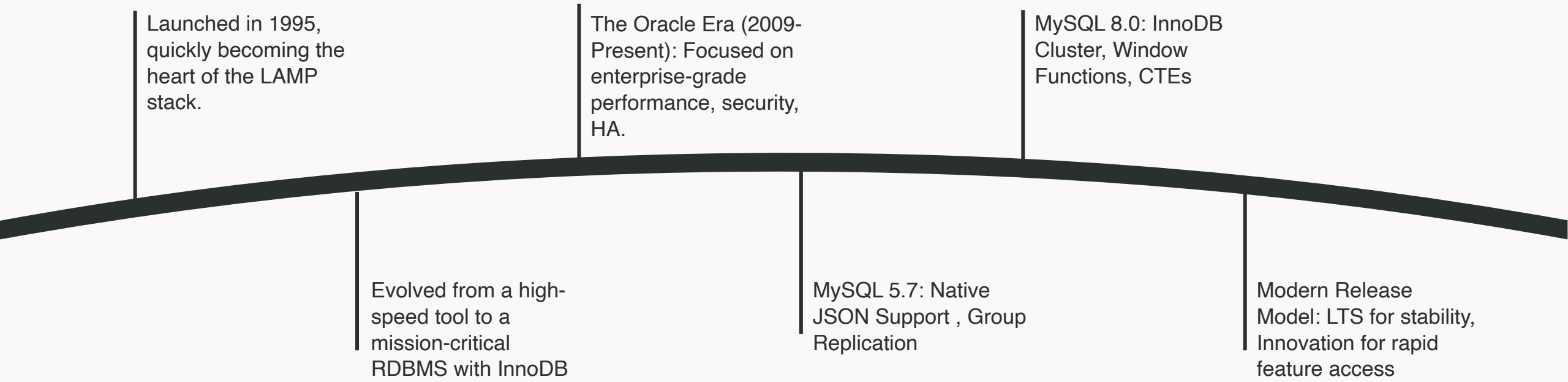
4 Failover automation

5 ProxySQL features for scalability

6 Key takeways

MySQL at 30 years — A Legacy of Reliability

MySQL Through the Decades: From Disruptor to Enterprise Standard



The Scaling Problem Today

- Explosive growth in user and data volume
- High concurrency = massive connection pressure
- Query latency affects UX and business KPIs
- Read/write contention & replica lag
- Complex topologies + HA demands

The Architectural Hurdles of Scale

Growing Workloads, Growing Pains

- **The Connection Bottleneck:** MySQL's "one-thread-per-connection" model struggles with modern, high-concurrency applications.
- **Resource Exhaustion:** Thousands of idle connections consume gigabytes of memory and cause CPU context-switching overhead.
- **Replication Lag:** Asynchronous replication can lead to stale reads, caused by long-running transactions, DDL locks, or overloaded replicas.
- **The Burden of Application-Level Read/Write Split:** Forcing read/write split logic into the application creates tight coupling and makes failover a manual, high-risk process.

Enter ProxySQL — A Smart SQL-Aware Proxy

Meet ProxySQL: Built for Modern MySQL Infrastructures

- A high-performance, SQL-aware proxy created to solve MySQL’s scaling limitations.
- Sits transparently between applications and the MySQL backend cluster.
- Intelligently manages the complete query lifecycle: routing, multiplexing, caching, and rewriting.
- Open-source, production-ready, and fully compatible with Oracle's HA solutions

Purpose-Built for Scaling MySQL

A high-performance SQL-aware proxy for MySQL

- Connection pooling & multiplexing
- Query-aware load balancing & routing
- Advanced failover automation
- Query caching, rewrites, and traffic shaping
- Fine-grained observability & access control

Connection Management Superpowers

Taming Thousands of Connections with Multiplexing

Problem:

Every client connection = dedicated MySQL thread

Idle connections still consume memory/CPU

Spikes in traffic = CPU/memory exhaustion

unresponsiveness are common

Solution:

Connection Multiplexing: Manages thousands of application connections while maintaining only a small pool of connections to the database.

Drastic Resource Savings: Frees up gigabytes of memory and reduces CPU context-switching on the database server.

Prevents Connection Storms: Absorbs traffic spikes from application restarts or scaling events, protecting the database from overload.

Session Integrity: Intelligently tracks transactions and session state to ensure multiplexing is applied safely.

Replica Lag Detection & Read Routing

Never read from a stale replica again

Problem:

Replicas lag behind the primary

Async replication = eventual consistency

Read-after-write anomalies possible

Inconsistent data = broken user experience

Solution:

Real-time monitoring of replication lag

Only routes reads to healthy, up-to-date replicas

Avoids stale reads during failovers or lag spikes

Works with async and semi-sync replication

Load Balancing & Query Distribution

Keep replicas hot—but not overheated

Problem:

- Not all replicas are equal (hardware, lag, load)
- App usually round-robins blindly
- One replica becomes a bottleneck
- Performance degrades unpredictably

Solution:

- Distributes queries across replicas using weights or performance
- Prevents hot spots and under-utilized nodes
- Automatically adapts to backend health and performance
- Supports custom weighting and dynamic rebalancing

Intelligent Query Routing & Distribution

Smarter Traffic, Better Performance, No Code Changes

Problem:

- Application code must decide: read or write?
- Complex logic spread across microservices
- Data inconsistency or performance hits
- Difficult to maintain as team grows

Solution:

- Hostgroups: Logically group servers by role (e.g., writers, readers)
- Query Rules: A powerful, ordered rules engine to control traffic flow
- Dynamic Read/Write Split: Route queries by type (read/write), regex, schema, user, or metadata
- Fine-grained control over traffic distribution
- Supports thousands of clusters simultaneously
- Decouples Application from Topology: DBAs can change routing rules at runtime without redeploying application code

High Availability Made Practical

Simplifying Failover and Redundancy

- Monitors MySQL nodes with health checks
- Automated failover and failback
- Integration with Group Replication, Galera, Aurora, RDS, etc
- Zero-downtime maintenance possible

Failover Automation

Stay online—no matter which server fails

- Monitors backend server health and replication status
- Automatically detects promoted replica and reconfigures routing
- Keeps client connections alive during failover
- Relies on external scripts for fencing, orchestration, or custom recovery
- Built-in mechanism to prevent split brain scenarios (multiple writers)
- Rerouting
- Seamless Transition

DBA's Secret Weapon: Caching & Rewriting

Optimize and Patch Queries On the Fly

Problem:

- Many identical queries to read stale data
- Bad queries
- slow, unsafe, or legacy queries
- Queries accessing sensitive data
- Burst of requests

Solution:

- High-Performance in-memory Query Cache
  - Invalidate by TTL
- Query Rewriting
  - On the fly query rewrite
- Apply traffic policies by user, time, or origin
- Simple, or advanced firewall
- Throttling
- A powerful tool for incident response and tactical optimization.

Observability & Real-Time Stats

See everything, act faster

- Built-in performance counters & query analytics
- Tracks latency, QPS, backend load, cache hits, and more
- Real-time query digest with histograms and patterns
- Integrates with Grafana, PMM, or custom dashboards

Layer of abstraction

Decoupling Applications from MySQL Infrastructure

- Developers focus on business logic
- No need to hardcode topology, failover, or routing logic
- No visibility into replica lag or backend health required
- CI/CD becomes easier: apps talk to ProxySQL, not directly to MySQL
- Ops can make changes transparently: scale, failover, migrate

The Premier Pairing: ProxySQL + InnoDB Cluster

- MySQL InnoDB Cluster: Oracle's complete, integrated HA solution.
  - Group Replication: Provides automated, consensus-based failover and data consistency.
  - MySQL Shell: The unified command center for easy cluster management.
  - MySQL Router: Lightweight proxy for transparent connection routing.
- ProxySQL is designed to work seamlessly with InnoDB Cluster, complementing its strengths.

ProxySQL vs MySQL Router

- ProxySQL has no limit on number of connections , vs 50k in Router
- ProxySQL has no limit on number of clusters , vs only 1 in Router
- ProxySQL supports all clustering solutions
- ProxySQL supports multiple backend versions
- ProxySQL can distribute traffic unevenly
- ProxySQL has more efficient/complex connections sharing/multiplexing
- ProxySQL provides powerful traffic control: caching, rewrite, throttling, firewall, etc
- ProxySQL provides observability & real-rime stats

Why Choose ProxySQL ?

Scaling MySQL? Start here!!

✅ Scalability – Handle hundreds of thousands of connections effortlessly

✅ Performance – Reduce query latency and backend load

✅ Reliability – Built-in failover and high availability

✅ Security – Advanced authentication , traffic filtering and data masking

✅ Observability – Real-time query analytics and monitoring

✅ Easy to adopt—no application changes needed

✅ Trusted by companies from startups to Fortune 500s

Key Takeaways

Scaling MySQL Smarter in the Real World

- At 30 years, MySQL is more relevant than ever, but modern workloads and infrastructures demand a modern scaling strategy.
- ProxySQL bridges the gap, solving connection scalability, routing complexity, and performance challenges without application changes.
- The combination of MySQL InnoDB Cluster and ProxySQL creates a premier, enterprise-grade HA architecture that is automated, resilient, and performant.
- You can start simple and adopt more advanced features as you grow, future-proofing your infrastructure.

Thank you

Join our community

https://github.com/sysown/proxysql

https://proxysql.com/

https://proxysql.com/documentation/