PlanetScale

# MySQL High Availability

## Connection handling and concurrency

Matthias Crauwels

Enterprise Customer Engineer

@mcrauwel

# My Ess Que Ell

The official way to pronounce “MySQL” is “My Ess Que Ell” (not “my sequel”), but we do not mind if you pronounce it as “my sequel” or in some other localized way.

https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html

1. Brief introduction to MySQL
2. Challenges of connection management
3. How PlanetScale supports connections
4. Questions

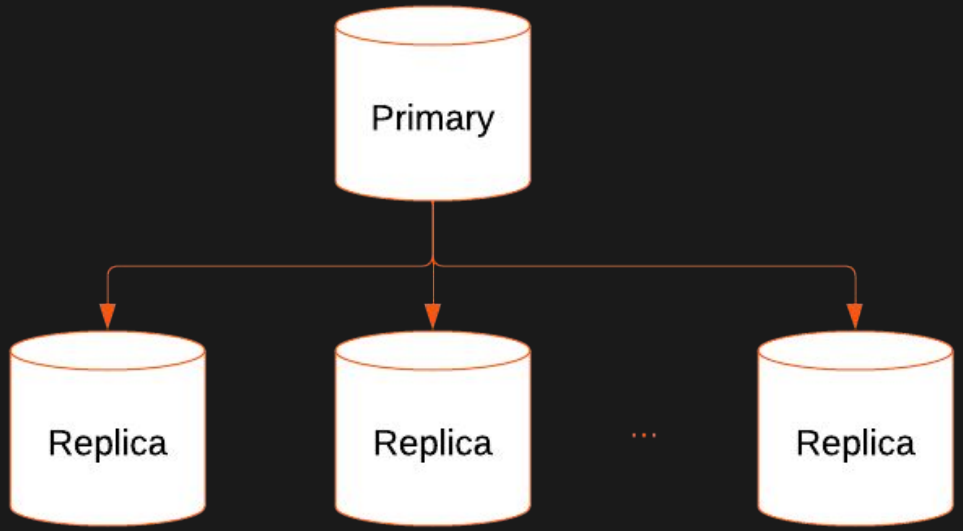
# MySQL powers the web

# Why MySQL got popular

- Free and open source software
- Easy to install
- Able to handle a lot of connections
- LAMP stacks
- Replication

# Replication

- Single primary
- Replica’s
  - Read scale out
  - HA / Failover
  - Backups
  - …

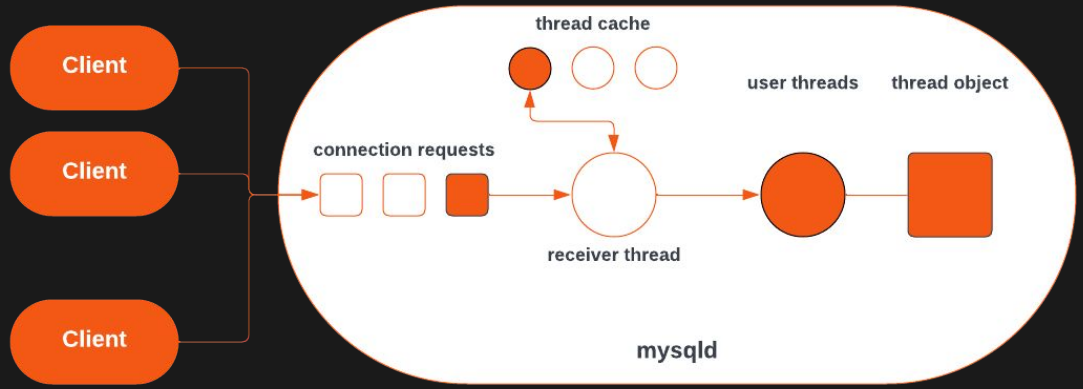


# Replication

- Traditionally asynchronous
- Semi-sync
- (virtually) synchronous
  - InnoDB Cluster
  - Galera Cluster

# MySQL connection handling

- MySQL has a thread based architecture
  - Each connection is handled by 1 user thread
- Receiver thread handles incoming connections
  - Connection requests in a queue
  - Creates the user thread

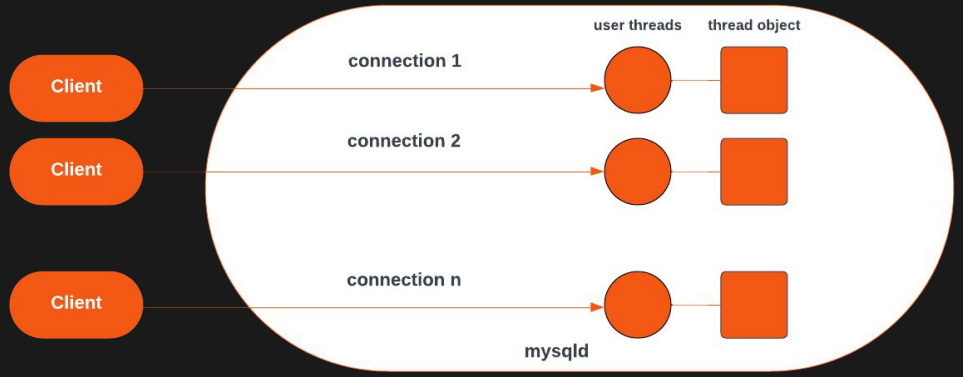


# Short lived connections

- Connections is only open for a short period of time
- Most common for PHP applications
- Typical flow
  - Connect
  - Query
  - Query
  - ...
  - Disconnect
- Thread object get deallocated at the disconnect

# Long lived connections

- Connections that are open “indefinitely”
- Typically opened at the start of the application process
- Kept open until the application is stopped
- One thread object per connection, never deallocated until connection is terminated



# Challenges

- Threads will keep on executing instructions until:
  - they need to wait for something
  - the OS scheduler decides it’s timeshare is used
- What can threads need to wait for
  - Mutex
  - Locks
  - I/O (disk, network, ...)

# Memory utilization

- Thread object holds per-connection buffers
- Short lived connections have a lot of overhead with regards to thread and TCP connection handling
- Long lived connections can create memory pressure

# Service Discovery

- Where is the primary?
- Which replica’s are available

# Connection pooling

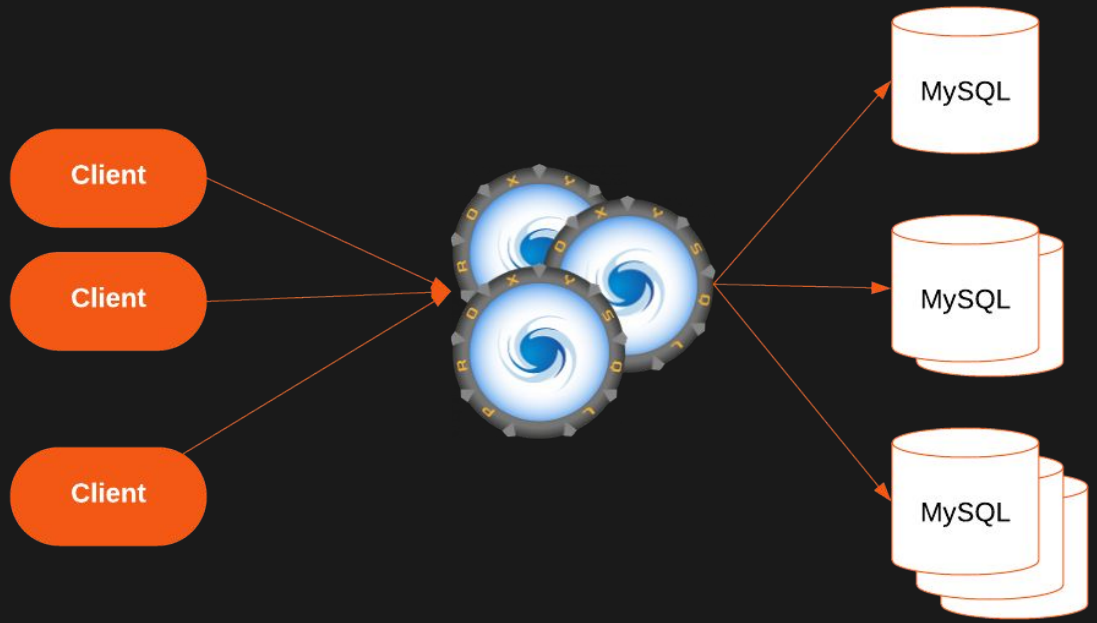
# What is connection pooling?

- An “application” that maintains your database connections
- Short lived becomes long lived
- Some application servers have it built in (think Tomcat/Java)
- Sometimes you need an external application

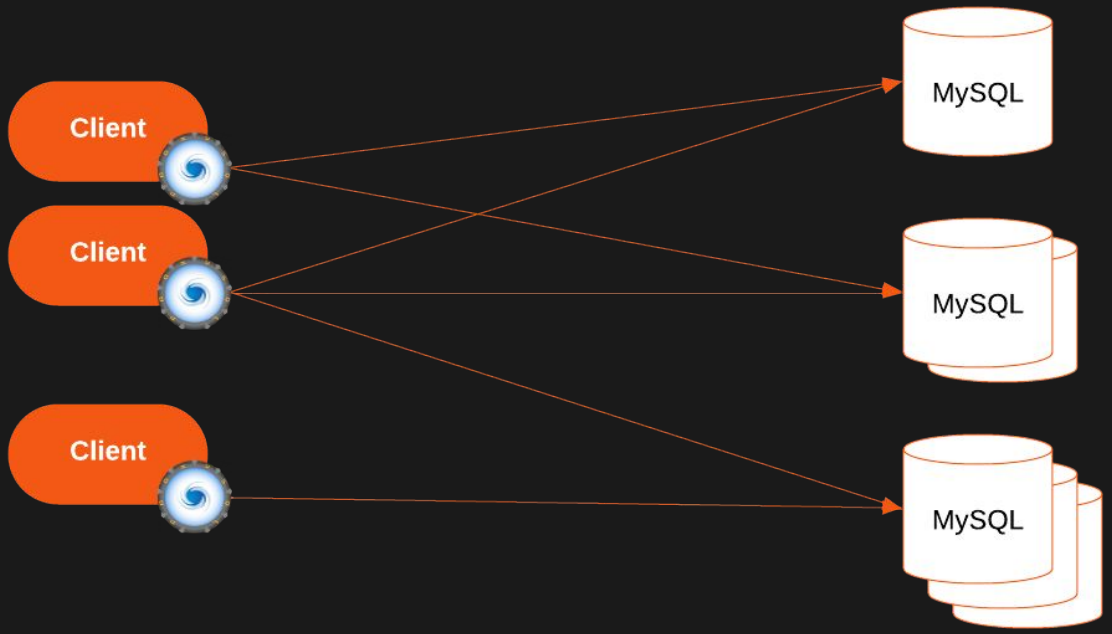
# ProxySQL

- High performance proxy server for MySQL
- Speaks the MySQL protocol
- Can provide intelligent load balancing of application requests onto multiple servers
- Understands the underlying database topology
- Knows whether instances are up or down
- Can be configured as a connection pooling application
- Full end-to-end SSL support

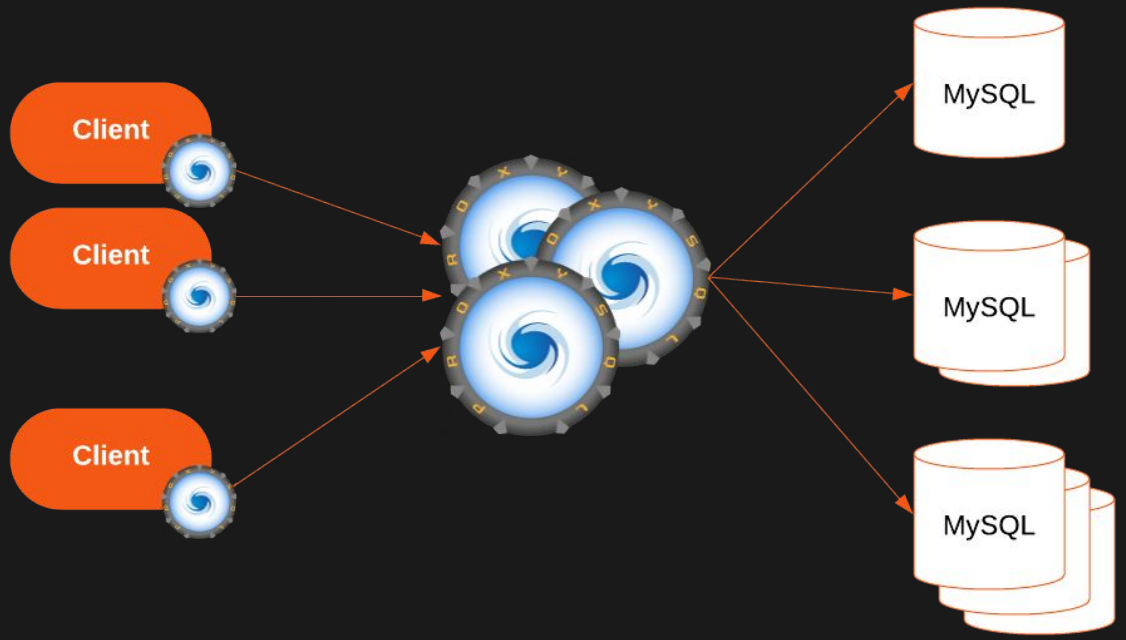
# ProxySQL architecture



# ProxySQL architecture



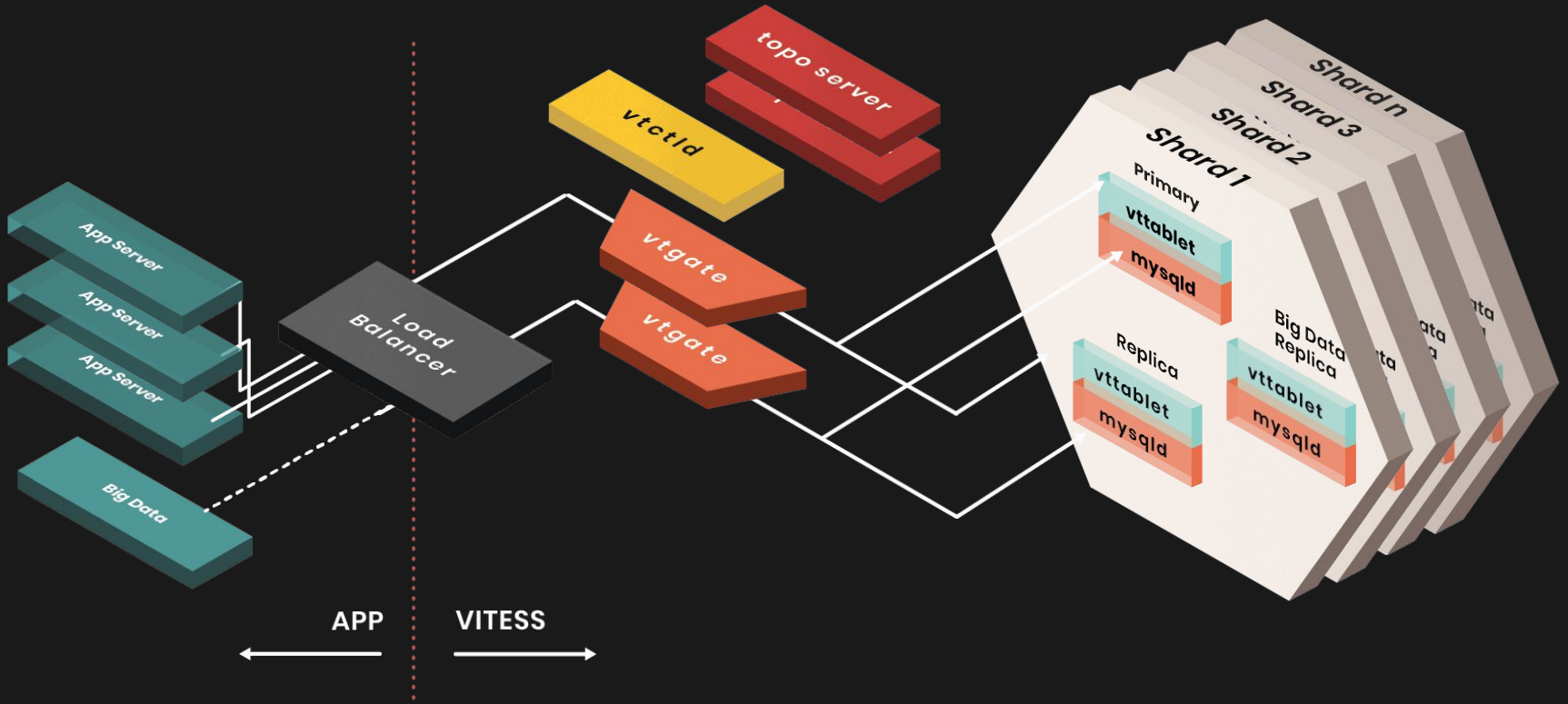
# ProxySQL architecture



# PlanetScale

- Built on Vitess
- In cloud (GCP / AWS) managed service
- Scalable
- Highly Available
- Sharding

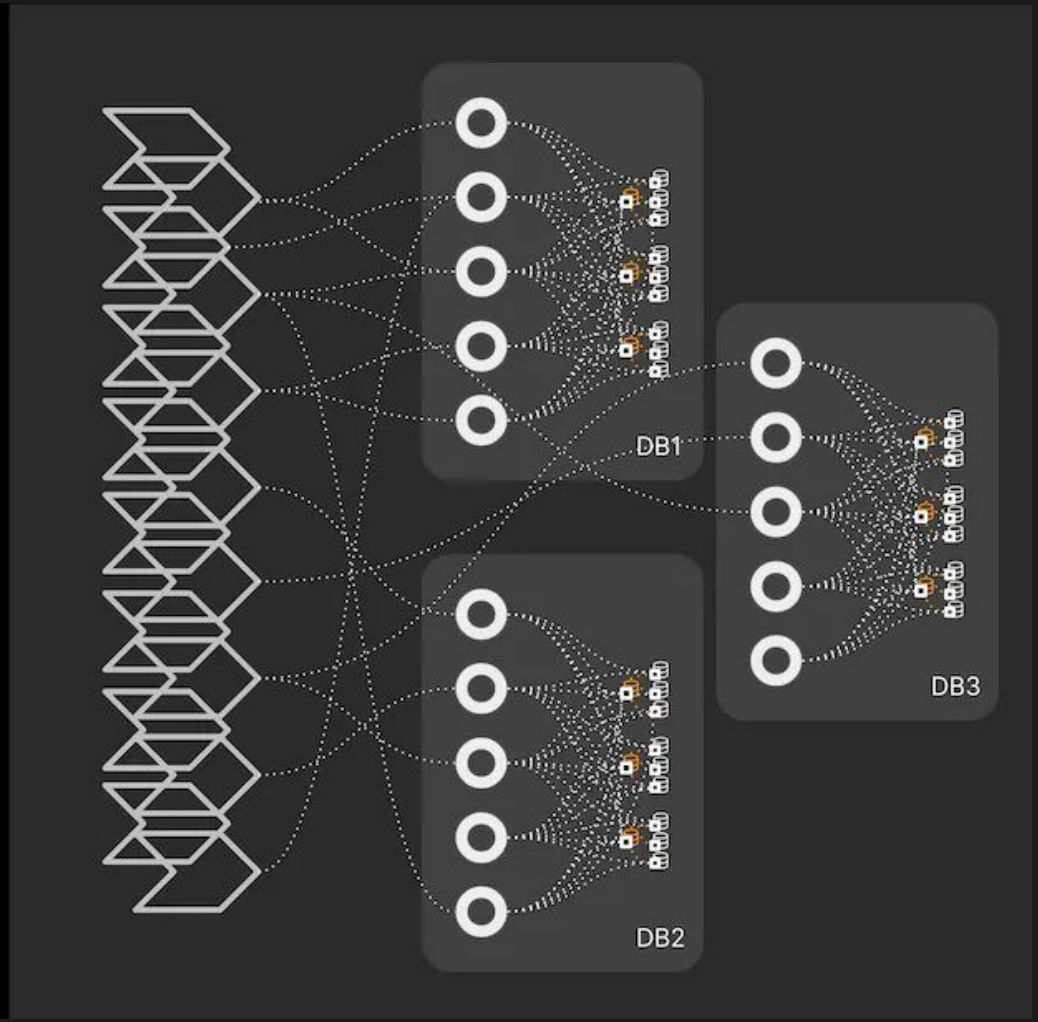
# Vitess architecture



# PlanetScale Edge infrastructure

- Works in a similar way to AWS Edge locations
- Terminates your MySQL connection in the closest edge location
- Relays the connection over the AWS/GCP internal backbone
- Also supports HTTPS queries (gRPC)

# PlanetScale Global Routing Infrastructure




MySQL instance

VTTablet sidecar process

VTGate stateless gateway

PlanetScale Edge

# One million connections


https://planetscale.com/blog/one-million-connections

# Questions?


PlanetScale