ProxySQL at Shopify
Who we are

Jordan Wheeler
Senior Production Engineer @ Shopify

René Cannaò
Founder @ ProxySQL
ProxySQL Architecture Overview

Data gateway
Clients connect to ProxySQL
Requests are evaluated
Actions are performed
Some of the most interesting features:

- on-the-fly rewrite of queries
- caching reads outside the database server
- connection pooling and multiplexing
- complex query routing and read/write split
- load balancing
- real time statistics
- monitoring
- data masking
- multiple instances on same ports
Some of the most interesting features:

- high availability and scalability
- seamless failover
- firewall
- query throttling
- query timeout
- query mirroring
- runtime reconfiguration
- scheduler
- support for Galera/PXC and Group Replication
Some of the most interesting features:

- support for millions of users
- support for tens of thousands of database servers
- native ProxySQL Clustering Solution
- support for ClickHouse as a backend
- support for Aurora
- SSL support for frontend
- SSLv1.2
- native Support for Galera
- causal reads using GTID
Multiplexing:

Reduce the number of connections against mysqld (configurable)
Many clients connections (tens of thousands) can use few backend connections (few hundreds)
Tracks connection status (transactions, user variables, temporary tables, etc)
Order by waiting time
600K Active Merchants
80K RPS Peak
$26B GMV 2017
1000 Developers
40 Deploys/Day
kyliejenner SURPRISE! The #WETSET is dropping on KylieCosmetics.com in 1 hour! @kyliecosmetics if you haven’t tried it before.. it’s about to be your new best friend!

Load more comments
elsyendypatty_Fb
elsyendypatty_Cb
syifaazzahraa__lb
syifaazzahraa__fb
syifaazzahraa__cb
yungmanok_Lb
_htthuy03_ Llibb
saisharma9990 💗💕🌈💖💖💖💖💖💖
_marvelous_Lb
bilnabilaahhAAajr__Lb

990,395 likes
JUNE 22

Log in to like or comment.
No ProxySQL:

ProxySQL:
No ProxySQL:

ProxySQL:
No ProxySQL:

ProxySQL:
11:14 AM

fucking job friends

that was definitely the smoothest flash sale at that volume i've ever seen
Cost of MySQL’s one-thread-per-connection

Too many software threads per hardware thread
CPU registries save/restore and context switching
  Mutexes/locks contentions
  CPU cache almost useless
High cost for access to memory
Avoid having a central bottleneck
Thread pool in MySQL
Thread pool in MySQL

Threads in ProxySQL are known as "MySQL Threads"
Fixed number of worker threads (configurable)
All threads listen on the same port(s)
Client connections are not shared between threads
All threads perform their own network I/O
Uses "poll()"... (does it scale?)
Threads never share client connections

Pros:
- Thread contention is reduced
- No need for synchronization
- Each thread calls "poll()"

Cons:
- Possibly imbalanced load
poll() vs. epoll()

"poll()" is O(N)
"epoll()" is O(1)
"epoll()" scales better than "poll()"

Why does ProxySQL use "poll()"?
It is faster than "epoll()" for fewer connections (~1000)
Performance degrades when there are a lot of connections
ProxySQL Auxiliary Threads

Each worker thread has an auxiliary thread. Worker thread uses "poll()" and auxiliary thread uses "epoll()". Worker thread passes idle connections to auxiliary thread. When a connection becomes active, auxiliary thread passes connection to the worker thread.

Solution scales to 1 million connections.
MySQL Thread Overview

All other Modules:
- Query Processor
- Query Cache
- Hostgroups Manager
- Authentication
- Others

For low contention, threads independently:
- Track internal metrics
- Store values for mysql-XXX variables
- Store a copy of the defined query rules
Contention on MyHGM

MyHGM is a shared resource so it can cause contention when accessed by MySQL Threads.
Thread Connection Cache

Each MySQL Thread has a connection cache that is reset before calling poll().
Kubernetes Service
shard123-db1

MySQL

Taiji

Writer | Instances
---------|----------

Hostgroup 0 | Hostgroup 1
-----------|-----------

shard123-db1
Questions?