

ProxySQL: Traffic management and Performance Troubleshooting

Vlad Fedorkov

ProxySQL Tech Day, Ghent

03.10.2019



What is ProxySQL?

- Written on C++
- Running as a separate daemon
- A drop-in solution for scaling MySQL-based applications
- Industry standard for high-volume projects
- Available under terms of GNU GPL v3.0 on <http://proxysql.com/>
 - You don't need to pay anyone for download and use ProxySQL within GPL v3 terms
- Developed and supported by ProxySQL
 - We do offer support and paid feature development



What ProxySQL does?

- It's processing, forwarding and returning results for MySQL queries
- Supports several clusters
- Manages connections and sessions
- Which leads to key ProxySQL features

What ProxySQL could do?

- On-the-fly query rewrite
- Connection pooling and multiplexing
- Complex query routing
- Read/write split
- Load balancing within hostgroup
- Real time statistics
- Zero downtime upgrade

What else?

- Query caching
- Seamless failover support
- Firewall
- Query
 - Throttling
 - Mirroring
- Zero time reconfiguration
- Cluster-wide configuration support
- Galera/PXC and Group Replication support

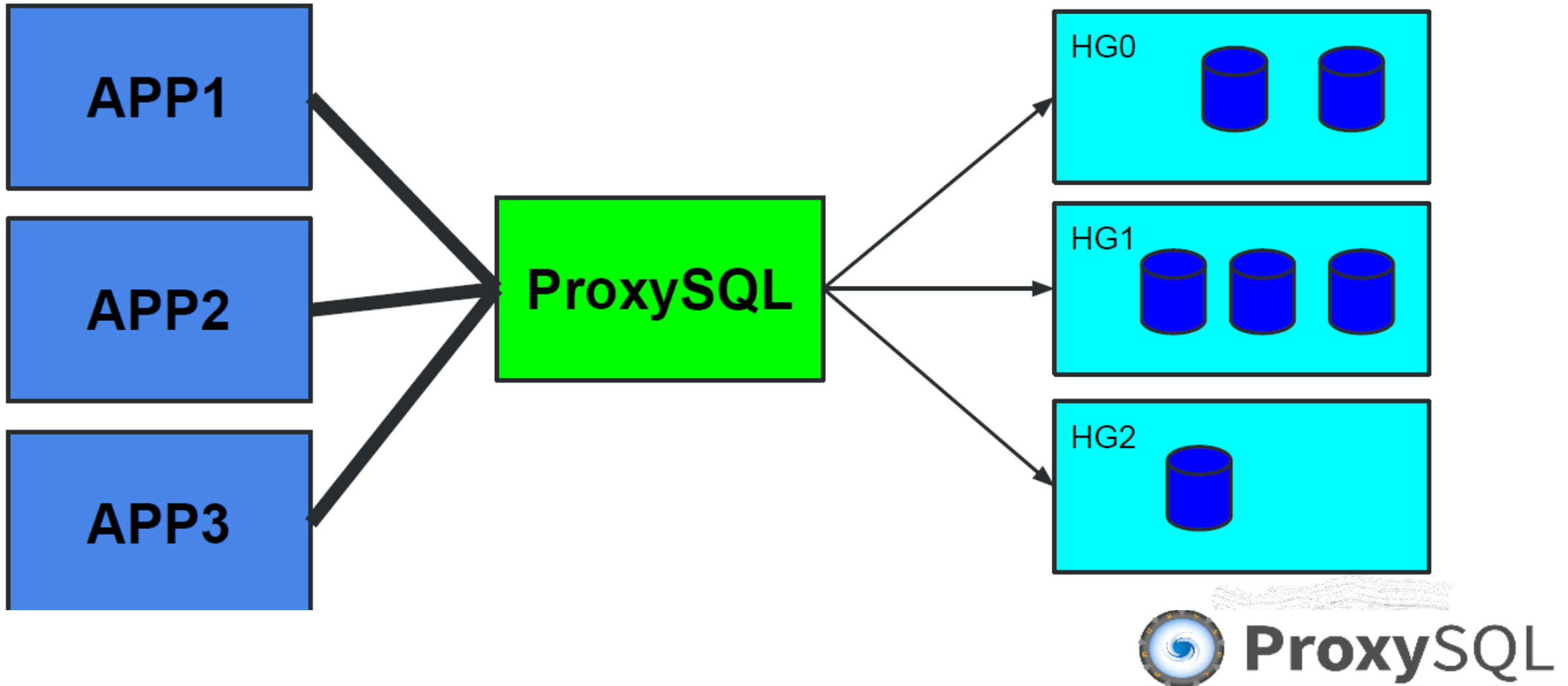
Read/Write split

- Writing to specific nodes
- Reading from everywhere
 - Can do weighted round-robin
 - Weights can be changes on the fly
- The hostgroups
 - HGxx0: Write masters
 - HGxx1: Read instances
- ProxySQL decide hostgroup based on the read_only flag
 - You need to be careful

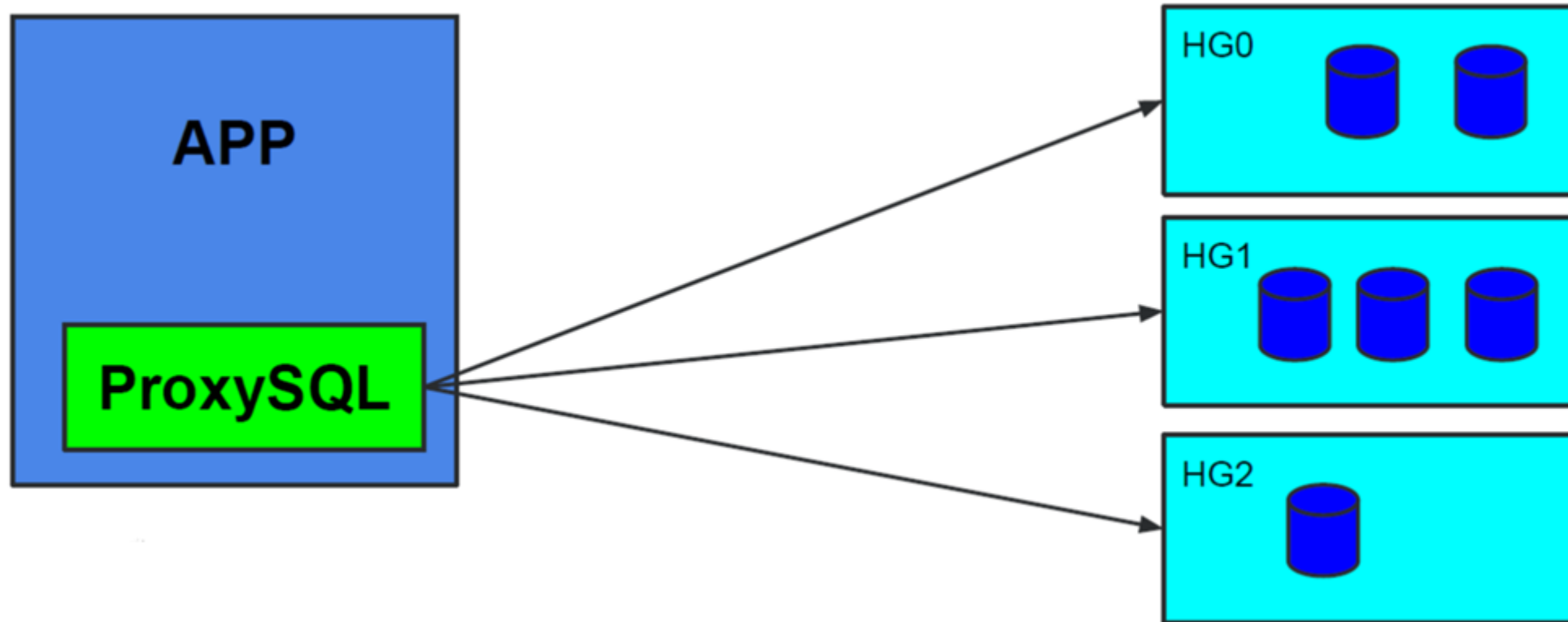
Why do we need hostgroups?

- Flexibility
- HG0: write nodes (RW)
- HG1: production reads (RO)
- HG2: reporting replicas
- HG3: table group (A,B,C)
- HG4: table group (D,E,F)
- HG5: single huge table H

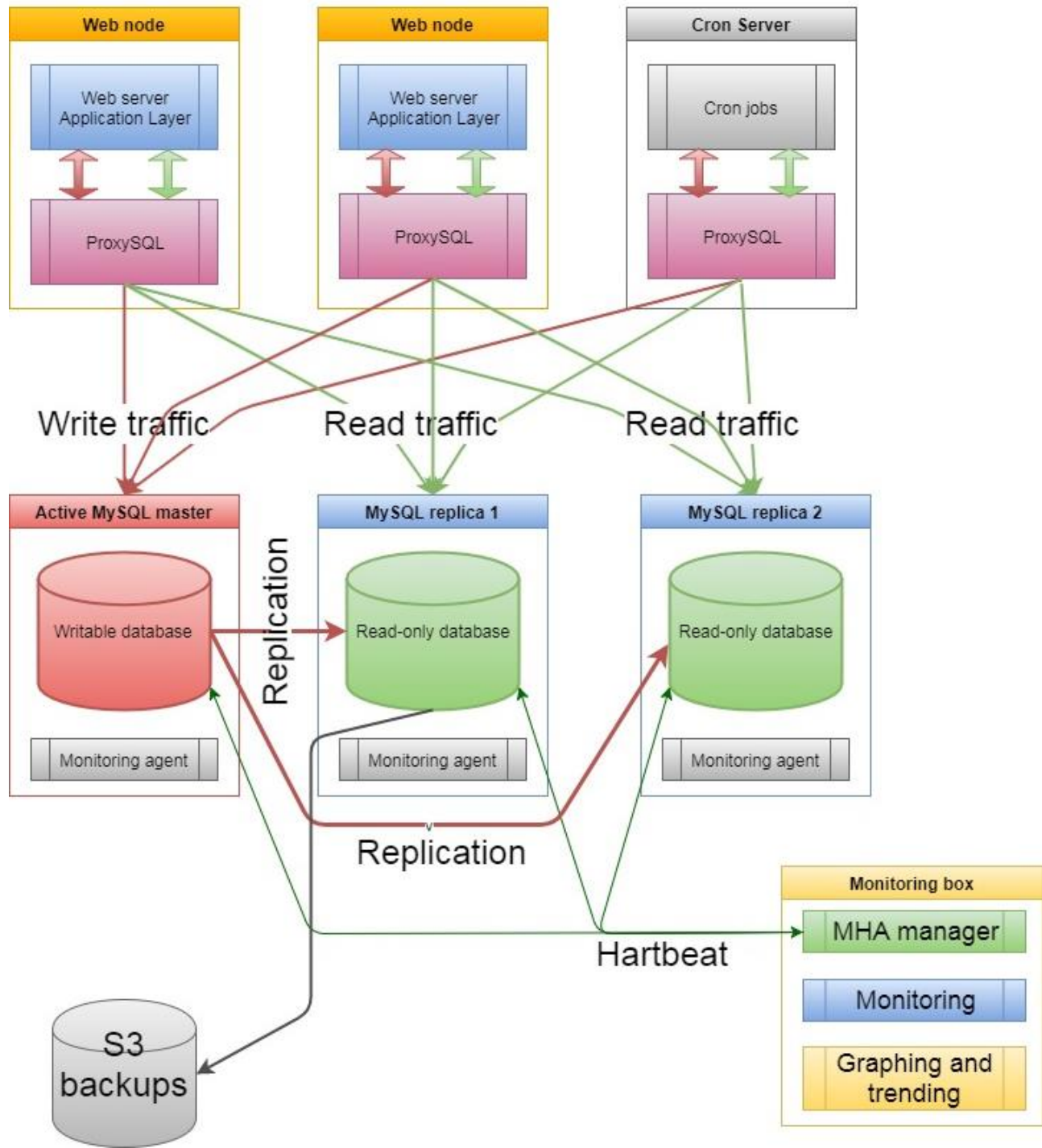
How does it work?



Better like this

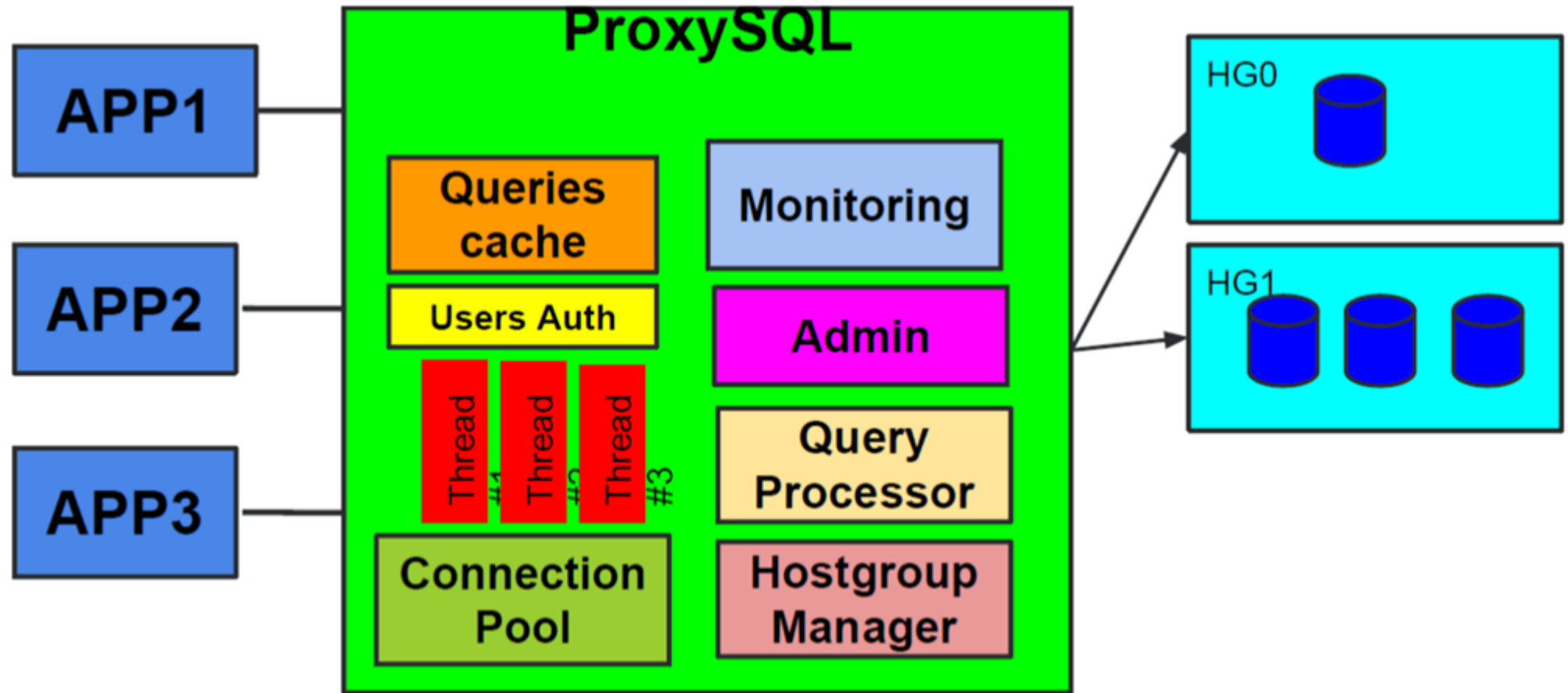


Nice looking diagram



About how cool it could've been

Architecture



Query Processor

- Based on Queries Rules
- Defines what to cache
- Defines the hostgroup target
- Timeout/delay
- Firewall
- Mirroring
- Rewrite queries

Query rules

- Complex rules to match incoming traffic:
- regex on query
- regex on digest text
- username
- schemaname
- Source IP address
- Bind IP address/port
- digest

Queries Cache

- Caching on the fly
- Internal key/value storage
- In memory only
- Pattern based
- Expired by timeout

Query Rewrite

- Rewrite on the fly
- Regex match/replace on query on digest text
- Optionally cached or mirrored

Hostgroups Manager

- Management of servers
- Track servers status
- Tightly integrated with the connections pool

Connections Pool

- Reduced the overhead of creating new connections,
 - and are recycled when not in use
- One to many connections
- Multiplexing & maximum connections
- Auto-reconnect and automatic re-execution of queries
- Failover management

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

Monitoring Module

- It monitors backends and collects metrics
- Monitors replication lag and shun hosts
- Monitors read_only variables (replication hostgroups)
- Ping and terminates unresponsive nodes
- Built in support for asynchronous replication
- Support for any type of clustering solution via external scripts (Galera/PXC , and any heterogeneous replication setup)
- support for Group Replication from ProxySQL 1.4.0

ProxySQL configuration

- Initial configuration is text based, located in /etc
 - Read only on the first start
- Table-based
 - Stored in SQLite database
 - Created from text version if missed
- 3 configuration levels:
 1. Production (live) configuration
 2. On-disk configuration
 3. Current configuration
- Configuration switch is atomic within a host
- ProxySQL cluster configuration is supported

Admin Interface

- DBA way to change configuration on the fly
- Uses MySQL protocol
- Accessible via any MySQL client on port 6032
- `mysql -h 0 -u admin -p -P 6032`
 - Default password is: admin

Host config sample

```
mysql> select hostgroup_id, hostname, status, weight,
max_connections from mysql_servers;
```

hostgroup_id	hostname	status	weight	max_connections
1	10.1.1.163	ONLINE	2000	1000
0	10.1.0.177	ONLINE	10	1000
1	10.1.0.177	ONLINE	10	1000

```
3 rows in set (0.00 sec)
```

Host config sample

```
***** 1. row *****
  hostgroup_id: 1
    hostname: 127.0.0.1
      port: 3306
    gtid_port: 0
      status: ONLINE
    weight: 1
  compression: 0
max_connections: 1000
max_replication_lag: 0
  use_ssl: 0
max_latency_ms: 0
  comment:
```

Live query statistics

```
mysql> select * from stats_mysql_query_digest order by count_star desc limit 1\G
***** 1. row *****
  hostgroup: 0
  schemaname: rsynx
  username: rsynx
client_address:
  digest: 0x8D6AE80711B33F7F
digest_text: SELECT option_value FROM wp_options WHERE option_name = ? LIMIT ?
count_star: 1200
first_seen: 1570102810
last_seen: 1570103414
  sum_time: 250357
  min_time: 122
  max_time: 679
1 row in set (0.00 sec)
```



Live connections statistics

```
mysql> select hostgroup, srv_host, srv_port, status, Queries, Latency_us  
from stats_mysql_connection_pool;
```

hostgroup	srv_host	srv_port	status	Queries	Latency_us
0	10.1.0.177	3306	ONLINE	277995417	183
1	10.1.1.163	3306	ONLINE	215559184	663
1	10.1.0.177	3306	ONLINE	1121035	183

```
3 rows in set (0.00 sec)
```

Live connections statistics

```
mysql> mysql> select hostgroup, srv_host, ConnUsed, ConnFree,  
ConnOK, ConnERR from stats_mysql_connection_pool;
```

hostgroup	srv_host	ConnUsed	ConnFree	ConnOK	ConnERR
0	10.1.0.177	1	26	27	0
1	10.1.1.163	0	7	159	516
1	10.1.0.177	0	3	11	1

```
3 rows in set (0.00 sec)
```

Questions!

Where to get help?

- <http://proxysql.com/blog>
- <http://Facebook.com/proxysql> & <http://twitter.com/proxysql>
- Ask ProxySQL team
 - We do on-site training for DevOps and Dev teams
 - Do have both support a consulting packages

Thank you!