

ProxySQL para MySQL

Mais que um simples proxy - Exemplos de uso

Marcelo Altmann

Senior Support Engineer

Percona University - São Paulo

27/04/2019



Agenda

Agenda

- Sobre Mim
- Introdução
- Features
- Caso de uso

Sobre Mim

Marcelo Altmann

- Engenheiro de Suporte @ percona
 - *MySQL DBA @ IEDR (CCTLD Irlanda)*
- Tecnólogo em Sistemas para Internet
- Oracle ACE Associate
- blog.marceloaltmann.com
- Certificações
 - *OCP, MySQL 5.7 Database Administrator*
 - *OCP, MySQL 5.6 Database Administrator*
 - *OCP, MySQL 5.6 Developer*
 - *OCP, MySQL 5 Database Administrator*
 - *OCP, MySQL 5 Developer*
 - *OCP, MySQL 5.0/5.1/5.5*

ProxySQL - Introdução

ProxySQL - Introdução

- Criado e mantido por René Cannaò
 - *MySQL Community - Contribuidor do ano (2017)*
 - *MySQL Community - Software do ano (2018)*
 - *MySQL Community Contributor Award - Oracle (2019)*
- Open Source
 - <http://www.proxysql.com/>
 - <https://github.com/sysown/proxysql>

ProxySQL - Introdução

- Instalação

- *Source Code / rpm / deb / percona repo (yum / apt-get)*

- MySQL admin interface

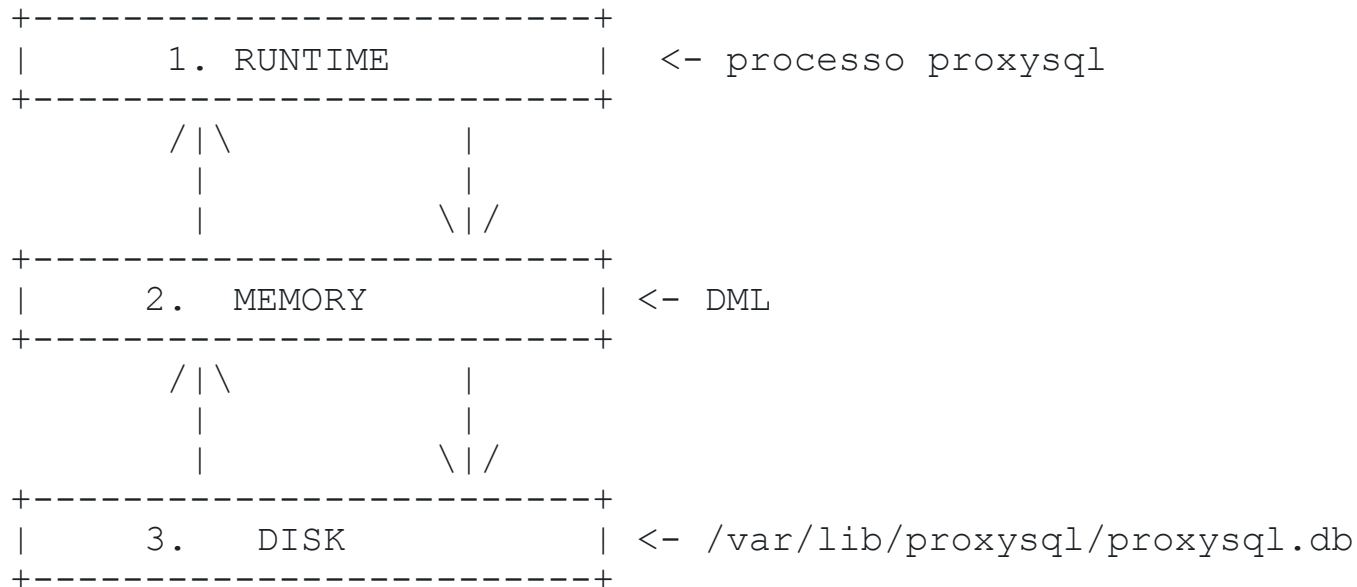
```
[root@localhost ~]# mysql -u admin -padmin -P 6032 -h 127.0.0.1 --prompt='Admin> '
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.5.30 (ProxySQL Admin Module)
. . .
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

Admin>
```


ProxySQL - Introdução

- Camadas
 - **Runtime** - Dados que o proxySQL acessa
 - **Main/Memory** - Camada onde executamos alterações e monitoramento
 - **Disk** - Camada para persistir dados. SQLite3

ProxySQL - Introdução



ProxySQL - Introdução

- Tabelas

- ***mysql_servers*** - *Contém a lista de servidores e HG (Host Group)*

```
mysql> SELECT hostgroup_id, hostname, status, comment FROM mysql_servers;
```

hostgroup_id	hostname	status	comment
10	node1	ONLINE	WRITE
11	node2	ONLINE	READ
11	node3	ONLINE	READ

ProxySQL - Introdução

- Tabelas

- *mysql_users* - Contém a lista de usuário para autenticação.

```
mysql> SELECT username, default_hostgroup FROM  
mysql_users;
```

username	default_hostgroup
proxysql_user	10

```
1 row in set (0.00 sec)
```

ProxySQL - Introdução

- Tabelas

- ***mysql_query_rules*** - *Contém a lista de regras para cache, reescrita e redirecionamento de queries.*

ProxySQL - Features - Load Balance

ProxySQL - Features - Load Balance

■ Usuários de autenticação

```
Admin> INSERT INTO mysql_users (username,password,default_hostgroup) VALUES ('proxysql_user','Pr0xySQL!',10);
```

```
Query OK, 1 row affected (0.00 sec)
```

```
Admin> LOAD MYSQL USERS TO RUNTIME ;
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
Admin> SAVE MYSQL USERS FROM RUNTIME; -- Hack para o plain-text password
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
Admin> SAVE MYSQL USERS TO DISK ;
```

```
Query OK, 0 rows affected (0.01 sec)
```

ProxySQL - Features - Load Balance

- Redirecionamento do tráfego para lista de servidores

```
Admin> INSERT INTO mysql_servers (hostname, hostgroup_id, comment) VALUES  
('node1',10,'WRITE'),('node2',11,'READ'),('node3',11,'READ');
```

```
Query OK, 3 rows affected (0.00 sec)
```

```
Admin> LOAD MYSQL SERVERS TO RUNTIME;
```

```
Query OK, 0 rows affected (0.01 sec)
```

```
Admin> SAVE MYSQL SERVERS TO DISK;
```

```
Query OK, 0 rows affected (0.02 sec)
```

```
Admin> SELECT hostgroup_id, hostname, status, comment FROM mysql_servers;
```

hostgroup_id	hostname	status	comment
10	node1	ONLINE	WRITE
11	node2	ONLINE	READ
11	node3	ONLINE	READ

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


ProxySQL - Features - Load Balance

- Dividir leituras e escritas

```
Admin> INSERT INTO mysql_query_rules (username,destination_hostgroup,active,match_digest,apply)
```

```
-> VALUES
```

```
-> ('proxysql_user',10,1,'^SELECT.*FOR UPDATE',1),
```

```
-> ('proxysql_user',11,1,'^SELECT ',1);
```

```
Query OK, 2 rows affected (0.00 sec)
```

```
Admin> LOAD MYSQL QUERY RULES TO RUNTIME; SAVE MYSQL QUERY RULES TO DISK;
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
Query OK, 0 rows affected (0.01 sec)
```

ProxySQL - Features - Load Balance

- Dividir leituras e escritas

```
[root@localhost ~]# mysql -u proxysql_user -pPr0xySQL! -P 6033 -h 127.0.0.1 -e 'INSERT INTO test.t1 VALUES ( @@hostname )'
```

```
[root@localhost ~]# mysql -u proxysql_user -pPr0xySQL! -P 6033 -h 127.0.0.1 -e 'SELECT @@hostname, server FROM test.t1'
```

```
+-----+-----+
| @@hostname | server |
+-----+-----+
|      node2 | node1  |
+-----+-----+
```

```
[root@localhost ~]# mysql -u proxysql_user -pPr0xySQL! -P 6033 -h 127.0.0.1 -e 'SELECT @@hostname, server FROM test.t1'
```

```
+-----+-----+
| @@hostname | server |
+-----+-----+
|      node3 | node1  |
+-----+-----+
```

ProxySQL - Features - Failover

ProxySQL - Features - Failover

- Replicação
 - Mysql_replication_hostgroups - Validar a variável read_only no MySQL

```
Admin> INSERT INTO mysql_replication_hostgroups  
(writer_hostgroup, reader_hostgroup) VALUES (10, 11);
```

```
Admin> LOAD MYSQL SERVERS TO RUNTIME; SAVE MYSQL SERVERS  
TO DISK;
```

ProxySQL - Features - Failover

- Percona XtraDB Cluster / Galera
 - Scheduler - Script que roda a cada X milisegundos
 - <https://github.com/percona/proxysql-admin-tool>
 - Modos
 - *Single Write*
 - *Load Balance*

ProxySQL - Features - Failover

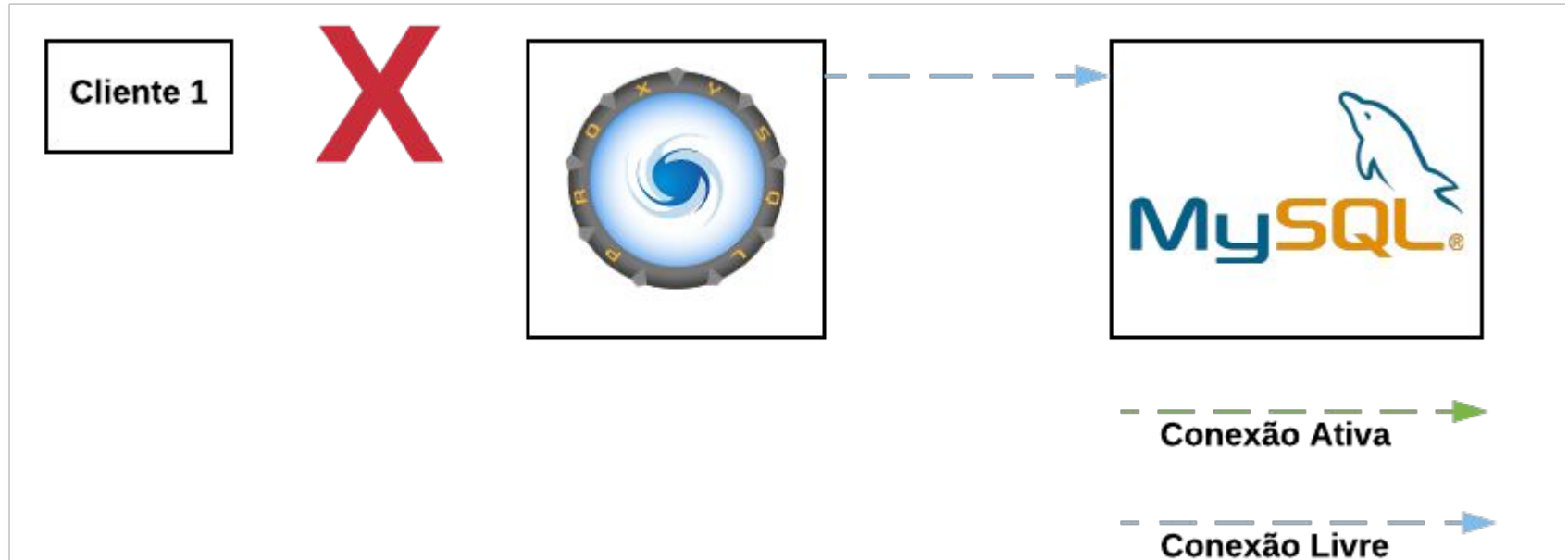
- Mysql_galera_hostgroups (proxysql 2.0)
 - *Writer_hostgroup*
 - *Backup_writer_hostgroup*
 - *Reader_hostgroup*
 - *Offline_hostgroup*

ProxySQL - Features - Connection Pool

ProxySQL - Features - Connection Pool



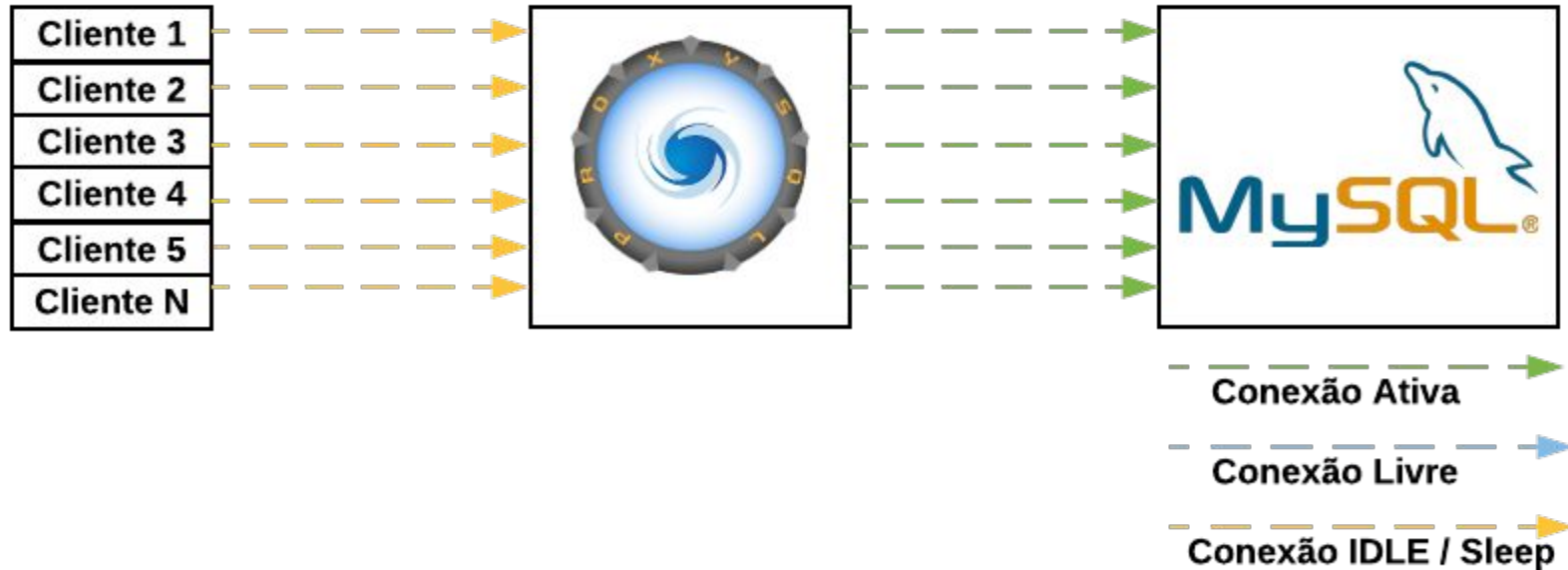
ProxySQL - Features - Connection Pool



ProxySQL - Features - Connection Pool

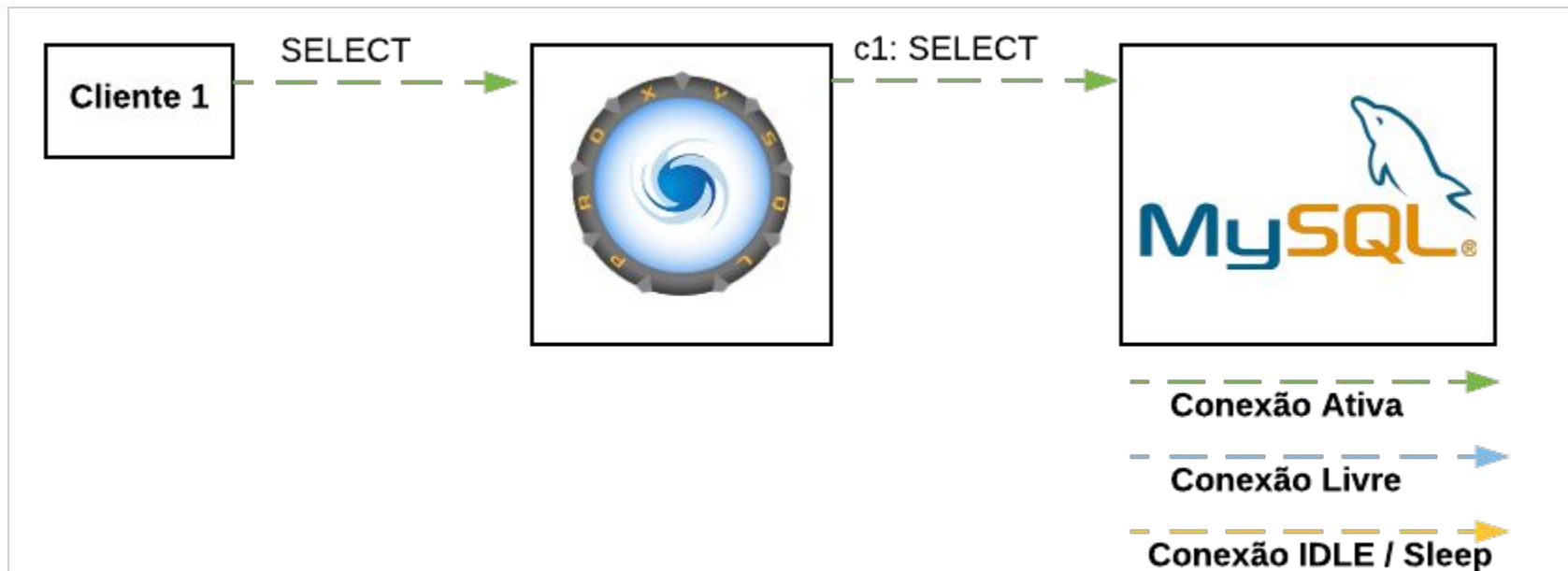


ProxySQL - Features - Connection Pool

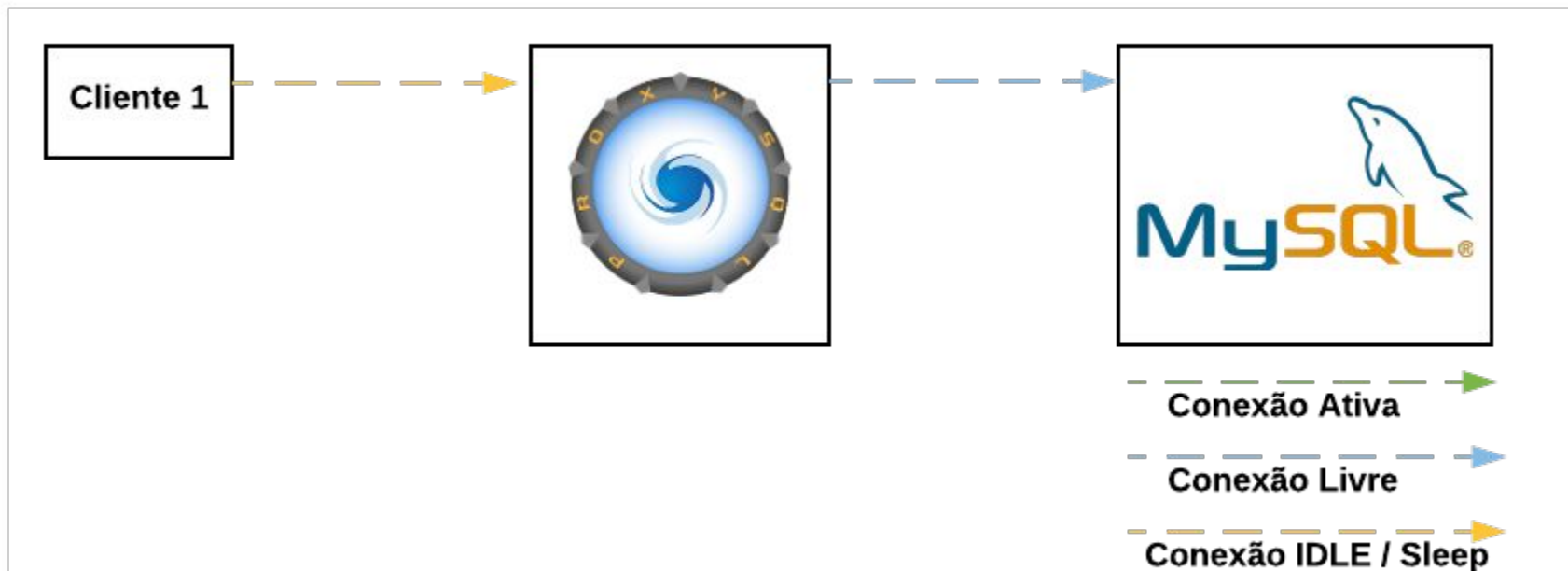


ProxySQL - Features - Multiplexing

ProxySQL - Features - Multiplexing



ProxySQL - Features - Multiplexing



ProxySQL - Features - Multiplexing



ProxySQL - Features - Stats

ProxySQL - Features - Stats

- SHOW TABLES FROM stats;
 - **Stats_mysql_commands_counters** - Contador baseado em comandos
 - **Stats_mysql_connection_pool** - Conexoes por servidor
 - **Stats_mysql_global** - Estatisticas globais
 - **Stats_mysql_processlist** - SHOW PROCESSLIST
 - **Stats_mysql_query_digest** - Contador agrupado por digest
 - **Stats_mysql_query_rules** - Contador baseado nas query rules

ProxySQL - Features - Reescrita de Query

ProxySQL - Features - Reescrita de query

■ stats_mysql_query_digest

```
Admin> SELECT digest, digest_text, sum_time FROM stats_mysql_query_digest ORDER BY sum_time DESC LIMIT 1;
```

```
+-----+-----+-----+
| digest          | digest_text                                     | sum_time |
+-----+-----+-----+
| 0xD69E622A5052289E | SELECT * FROM world.city WHERE Name = ? | 7016461 |
+-----+-----+-----+
```

```
Admin> INSERT INTO mysql_query_rules (rule_id,active,match_pattern, replace_pattern) VALUES (3,1,
'^SELECT \* FROM world\.city WHERE Name = (.*)$',
'SELECT Population FROM world.city WHERE Name = \1');
```

```
Query OK, 1 row affected (0.00 sec)
```

```
Admin> LOAD MYSQL QUERY RULES TO RUNTIME;
```

```
Query OK, 0 rows affected (0.01 sec)
```

```
Admin> SAVE MYSQL QUERY RULES TO DISK;
```

```
Query OK, 0 rows affected (0.01 sec)
```

ProxySQL - Features - Reescrita de query

```
[root@localhost ~]# mysql -u proxysql_user -pPr0xySQL! -P 6033 -h 127.0.0.1 -e "SELECT * FROM world.city WHERE Name = 'São Paulo'"
```

```
+-----+  
| Population |  
+-----+  
|      9968485 |  
+-----+
```

ProxySQL - Features - Firewall

ProxySQL - Features - Firewall

- Bloquear queries - SQL INJECTION!
 - **Original query:** *SELECT Name FROM world.city WHERE Name = '?'*
 - **SQL Injection:** *? = São Paulo' OR ID > 0; --*
 - *SELECT Name FROM world.city WHERE Name = 'São Paulo' OR ID > 0; --'*

```
Admin> SELECT username, digest, digest_text FROM stats_mysql_query_digest WHERE digest_text LIKE '% OR ID %'  
ORDER BY first_seen DESC LIMIT 1;
```

```
+-----+-----+-----+  
| username | digest | digest_text |  
+-----+-----+-----+  
| application | 0xD8AF41BF32707ABD | SELECT Name FROM world.city WHERE Name = ? OR ID > ? |  
+-----+-----+-----+  
1 row in set (0.00 sec)
```

ProxySQL - Features - Firewall

```
Admin> INSERT INTO mysql_query_rules (rule_id, active, digest, error_msg, apply) VALUES  
(4,1,'0xD8AF41BF32707ABD','Suspeita de SQL Injection',1);  
Query OK, 1 row affected (0.00 sec)
```

```
Admin> LOAD MYSQL QUERY RULES TO RUNTIME;SAVE MYSQL QUERY RULES TO DISK;  
Query OK, 0 rows affected (0.01 sec)  
Query OK, 0 rows affected (0.01 sec)
```

```
[root@localhost ~]# mysql -u application -papp -P 6033 -h 127.0.0.1 -e "SELECT Name FROM  
world.city WHERE Name = 'São Paulo' OR ID > 0; --' "  
ERROR 1148 (42000) at line 1: Suspeita de SQL Injection
```

ProxySQL - Features - Espelhamento de Query

ProxySQL - Features - Mirror

- Espelhar queries em outros servidores
 - *Testar configurações*
 - *Standby Master - Manter buffer pool quente*
 - *Testar workload em versões diferentes - Upgrades*
 - *Troubleshooting*

ProxySQL - Features - Mirror

- Esquentar buffer pool

```
Admin> INSERT INTO mysql_servers (hostgroup_id, hostname) VALUES (12, 'node4');
```

```
Admin> LOAD MYSQL SERVERS TO RUNTIME; SAVE MYSQL SERVERS TO DISK;
```

```
Admin> SELECT rule_id, match_digest, destination_hostgroup, mirror_hostgroup FROM  
mysql_query_rules WHERE rule_id=2;
```

rule_id	match_digest	destination_hostgroup	mirror_hostgroup
2	^SELECT	11	NULL

```
Admin> UPDATE mysql_query_rules SET mirror_hostgroup=12 WHERE rule_id=2;
```

```
Admin> LOAD MYSQL QUERY RULES TO RUNTIME;
```

```
Admin> SAVE MYSQL QUERY RULES TO DISK;
```

ProxySQL - Features - Query Cache

ProxySQL - Features - Query Cache

- Cache de queries baseado em TTL

```
[root@localhost ~]# sysbench --num-threads=16 --max-requests=0 --max-time=60 --test=/usr/share/doc/sysbench/tests/db/oltp.lua
--mysql-user=application --mysql-password=app --mysql-db=test --mysql-host=127.0.0.1 --mysql-port=6033 --oltp-table-size=10000
--oltp-read-only=on --oltp-skip-trx=on --oltp-point-selects=100 --oltp-simple-ranges=1 --oltp-sum-ranges=1 --oltp-order-ranges=1
--oltp-distinct-ranges=1 run | grep 'read/write requests'
```

```
read/write requests:          105664 (1744.81 per sec.)
```

```
Admin> SELECT count_star,sum_time,hostgroup,digest,digest_text FROM stats_mysql_query_digest ORDER BY sum_time DESC;
```

```
+-----+-----+-----+-----+-----+
| count_star | sum_time | hostgroup | digest          | digest_text                                     |
|-----+-----+-----+-----+-----+
| 100200     | 459147213 | 11        | 0xBF001A0C13781C1D | SELECT c FROM sbtest1 WHERE id=?             |
| 1002       | 6533622   | 11        | 0xF7D3CD60704822A0 | SELECT DISTINCT c FROM sbtest1 WHERE id BETWEEN ? AND ?+? ORDER BY c |
| 1002       | 6061540   | 11        | 0x877EEAFAADF3DDF2 | SELECT c FROM sbtest1 WHERE id BETWEEN ? AND ?+? ORDER BY c |
| 1002       | 5905677   | 11        | 0xAF7A51977DD56217 | SELECT c FROM sbtest1 WHERE id BETWEEN ? AND ?+? |
| 1002       | 5321376   | 11        | 0x3E268CF3E75DB831 | SELECT SUM(K) FROM sbtest1 WHERE id BETWEEN ? AND ?+?
```

ProxySQL - Features - Query Cache

```
Admin> INSERT INTO mysql_query_rules (rule_id,active,digest,cache_ttl,apply) VALUES (5,1, '0xBF001A0C13781C1D' ,2000,1);
```

```
Query OK, 1 row affected (0.00 sec)
```

```
Admin> LOAD MYSQL QUERY RULES TO RUNTIME;SAVE MYSQL QUERY RULES TO DISK;
```

```
[root@localhost ~]# sysbench --num-threads=16 --max-requests=0 --max-time=60 --test=/usr/share/doc/sysbench/tests/db/oltp.lua
--mysql-user=application --mysql-password=app --mysql-db=test --mysql-host=127.0.0.1 --mysql-port=6033 --oltp-table-size=10000
--oltp-read-only=on --oltp-skip-trx=on --oltp-point-selects=100 --oltp-simple-ranges=1 --oltp-sum-ranges=1 --oltp-order-ranges=1
--oltp-distinct-ranges=1 run | grep 'read/write requests'
```

```
read/write requests:                238680 (3956.42 per sec.)
```

```
Admin> SELECT count_star,sum_time,hostgroup,digest,digest_text FROM stats_mysql_query_digest ORDER BY sum_time DESC;
```

count_star	sum_time	hostgroup	digest	digest_text
140512	632180517	11	0xBF001A0C13781C1D	SELECT c FROM sbtest1 WHERE id=?
3372	18351846	11	0xF7D3CD60704822A0	SELECT DISTINCT c FROM sbtest1 WHERE id BETWEEN ? AND ?+? ORDER BY c
3372	17739689	11	0x877EEAAFADF3DDF2	SELECT c FROM sbtest1 WHERE id BETWEEN ? AND ?+? ORDER BY c
3372	17709660	11	0xAF7A51977DD56217	SELECT c FROM sbtest1 WHERE id BETWEEN ? AND ?+?
3372	15646777	11	0x3E268CF3E75DB831	SELECT SUM(K) FROM sbtest1 WHERE id BETWEEN ? AND ?+?
196688	0	-1	0xBF001A0C13781C1D	SELECT c FROM sbtest1 WHERE id=?

ProxySQL - Features - GTID Leituras Consistentes

ProxySQL 2.0 - GTID Leituras Consistentes

- Stale reads
- Requer modificações na aplicação
 - `WAIT_FOR_EXECUTED_GTID_SET`
 - *Verificar multiplos slaves*
 - *Nenhum slave em sync*

ProxySQL 2.0 - GTID Leituras Consistentes

- MySQL 5.7.5+ - session_track_gtids
- Verificar GTID no slave?
 - *Pull - Verificar slaves em intervalos*
 - *Push - Slaves notificam o ProxySQL*

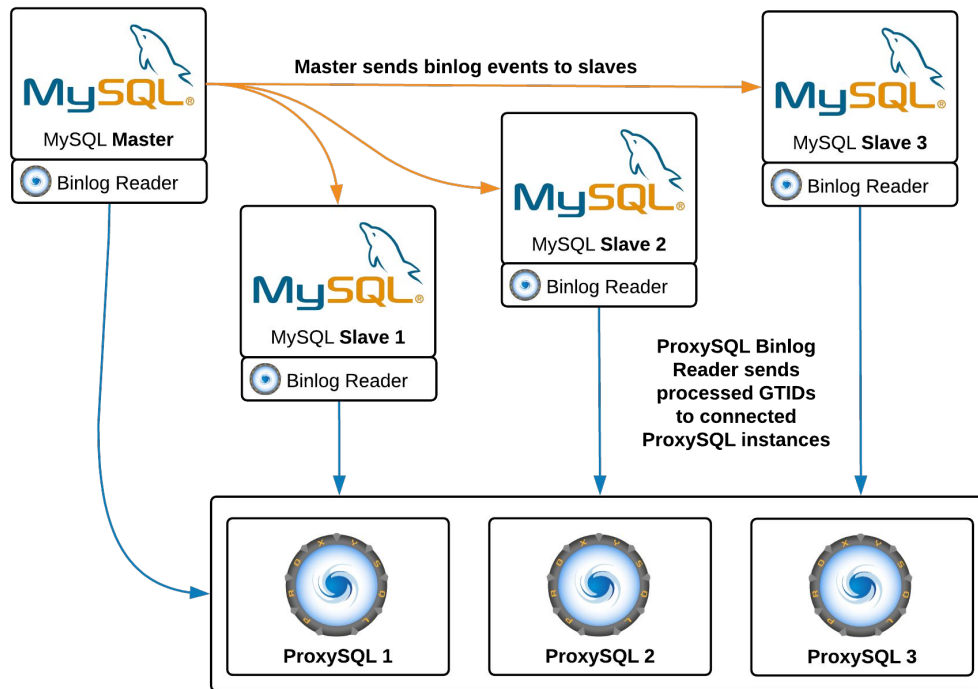
ProxySQL 2.0 - GTID Leituras Consistentes

- ProxySQL Binlog Reader

- https://github.com/sysown/proxysql_mysqlbinlog

ProxySQL 2.0 - GTID Leituras Consistentes

ProxySQL Binlog Reader



ProxySQL 2.0 - GTID Leituras Consistentes

01:29:54.081552 IP (tos 0x0, ttl 64, id 25059, offset 0, flags [DF], proto TCP (6), length 100)

marcelo-altmann-PU-replication-1.lxd.opsession-prxy > marcelo-altmann-PU-proxysql-1.lxd.41726: Flags [P.], cksum 0x5dfd (incorrect -> 0xc8cb), seq 1:49, ack 1, win 510, options [nop,nop,TS val 792525047 ecr 430181741], length 48

```
0x0000:  4500 0064 61e3 4000 4006 7b0a ac10 025f  E..da.@.@.{...._
0x0010:  ac10 0327 0ceb a2fe c48c 6725 bf6c 4c56  ...'.....g%.lLV
0x0020:  8018 01fe 5dfd 0000 0101 080a 2f3c f8f7  ....]...../<..
0x0030:  19a4 0d6d 5354 3d64 6131 6262 3030 302d  ..ST=da1bb000-
0x0040:  3563 3963 2d31 3165 392d 3965 6537 2d30  5c9c-11e9-9ee7-0
0x0050:  3031 3633 6566 6162 6163 303a 312d 3535  0163efabac0:1-55
0x0060:  3239 390a                                299.
```

```
mysql> SHOW GLOBAL VARIABLES LIKE 'gtid_executed';
```

Variable_name	Value
gtid_executed	da1bb000-5c9c-11e9-9ee7-00163efabac0:1-55299

1 row in set (0.00 sec)

ProxySQL 2.0 - GTID Leituras Consistentes

```
01:30:26.066393 IP (tos 0x0, ttl 64, id 25060, offset 0, flags [DF], proto TCP (6), length 94)
marcelo-altmann-PU-replication-1.lxd.opsession-prxy > marcelo-altmann-PU-proxysql-1.lxd.41726: Flags [P.],
cksum 0x5df7 (incorrect -> 0x93b9), seq 49:91, ack 1, win 510, options [nop,nop,TS val 792557032 ecr 430181741],
length 42
```

```
0x0000:  4500 005e 61e4 4000 4006 7b0f ac10 025f  E..^a.@.@.{...._
0x0010:  ac10 0327 0ceb a2fe c48c 6755 bf6c 4c56  ...'.....gU.lLV
0x0020:  8018 01fe 5df7 0000 0101 080a 2f3d 75e8  ....]...../=u.
0x0030:  19a4 0d6d 4931 3d64 6131 6262 3030 3035  ...Tl=da1bb0005
0x0040:  6339 6331 3165 3939 6565 3730 3031 3633  c9c11e99ee700163
0x0050:  6566 6162 6163 303a 3535 3330 300a      efabac0:55300.
```

```
mysql> SHOW GLOBAL VARIABLES LIKE 'gtid_executed';
+-----+-----+
| Variable_name | Value                                     |
+-----+-----+
| gtid_executed | da1bb000-5c9c-11e9-9ee7-00163efabac0:1-55300 |
+-----+-----+
1 row in set (0.00 sec)
```

ProxySQL 2.0 - GTID Leituras Consistentes

```
01:30:44.388537 IP (tos 0x0, ttl 64, id 25061, offset 0, flags [DF], proto TCP (6), length 61)
    marcelo-altmann-PU-replication-1.lxd.opsession-prxy > marcelo-altmann-PU-proxysql-1.lxd.41726: Flags [P.],
cksum 0x5dd6 (incorrect -> 0xa8c8), seq 91:100, ack 1, win 510, options [nop,nop,TS val 792575354 ecr
430213726], length 9
    0x0000:  4500 003d 61e5 4000 4006 7b2f ac10 025f  E..=a.@.@.{/..._
    0x0010:  ac10 0327 0ceb a2fe c48c 677f bf6c 4c56  ...'.....g..lLV
    0x0020:  8018 01fe 5dd6 0000 0101 080a 2f3d bd7a  ....]...../=.z
    0x0030:  19a4 8a5e 4932 3d35 3533 3031 0a          ...r2=55301.
```

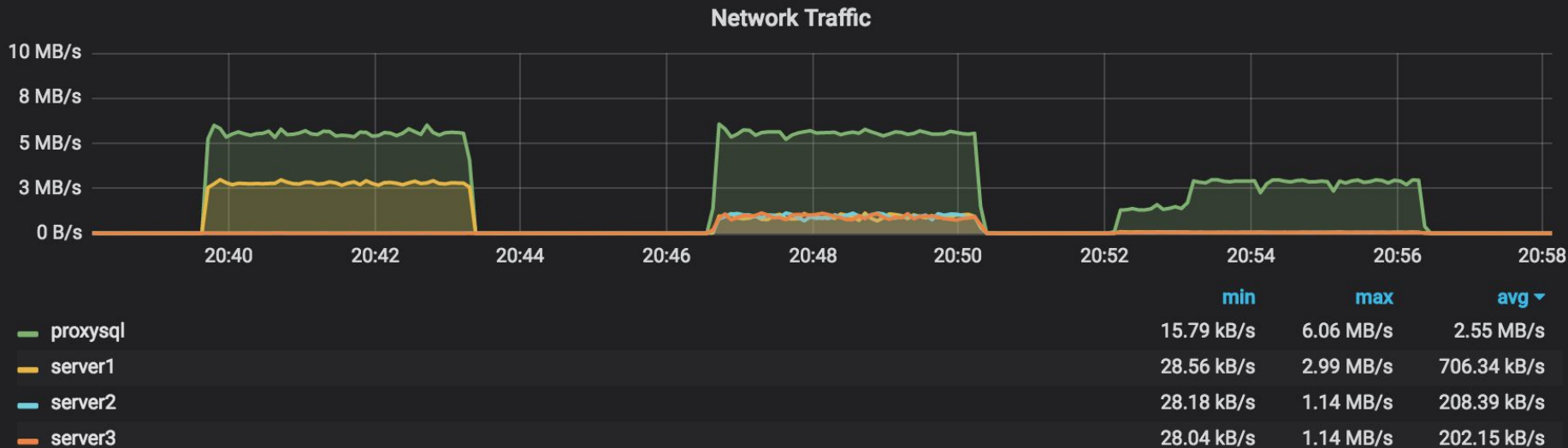
```
mysql> SHOW GLOBAL VARIABLES LIKE 'gtid_executed';
+-----+-----+
| Variable_name | Value                                     |
+-----+-----+
| gtid_executed | da1bb000-5c9c-11e9-9ee7-00163efabac0:155301 |
+-----+-----+
1 row in set (0.00 sec)
```

Caso de uso

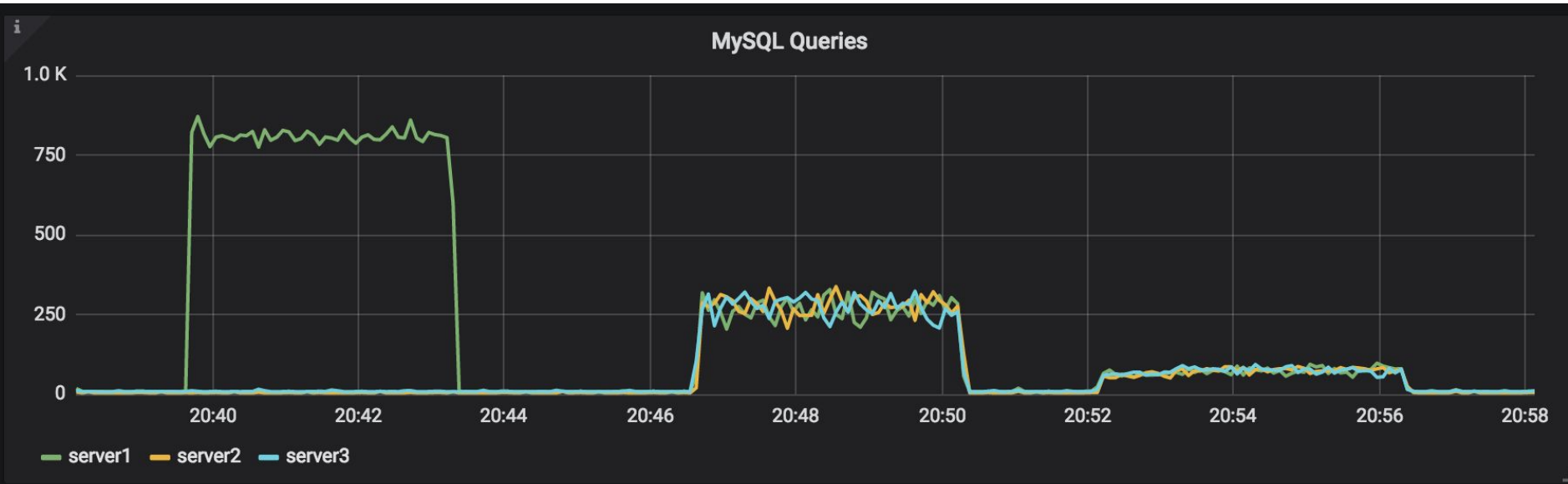
ProxySQL - Caso de uso

- Wordpress - 20 clientes conectados / 5K requests
- Teste 1 - Conectando somente em 1 servidor
- Teste 2 - Distribuindo carga entre 3 servidores
- Teste 3 - Realizando cache das top 20 queries

ProxySQL - Caso de uso



ProxySQL - Caso de uso



Perguntas?

Obrigado!



Database Performance Matters