

Percona Live 2017 ProxySQL Sharding

Marco tusa

Santa Clara, California

April 24th – 27th, 2017



PERCONA
LIVE EUROPE
DUBLIN

ProxySQL Sharding

Marco (the Grinch) Tusa
September 2017 Dublin



PERCONA
LIVE EUROPE
DUBLIN

About me



- Marco “**The Grinch**”
 - Open source enthusiast
 - In love with ProxySQL
 - Consulting team leader
 - Consultant myself

Agenda

Example of ProxySQL/MySQL setup

Understand Query Rules

Examples:

IP/port

Schema using field=value

Comment -> schema or host

Use combination of host/schema

Real life case of need to reduce sharding

A lot of code so you can reuse

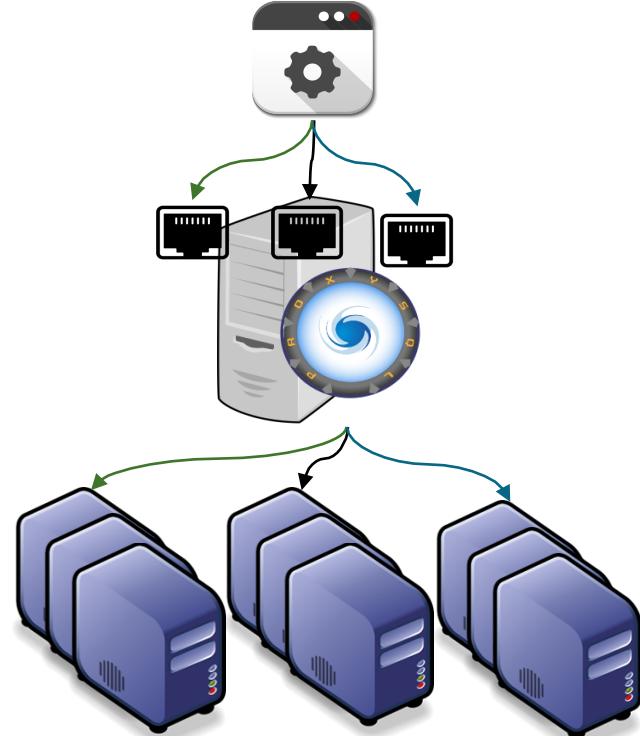
Sharding ...

Examples will use world DB.

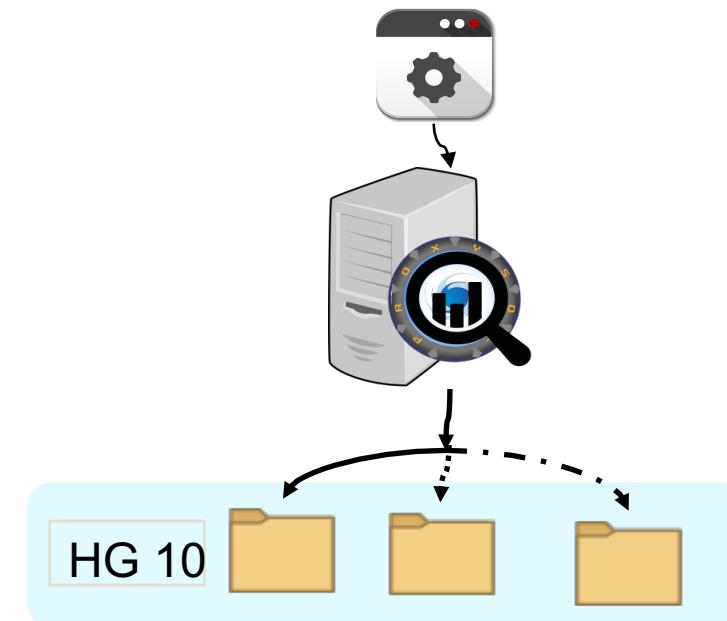


<http://downloads.mysql.com/docs/world-db.tar.gz>

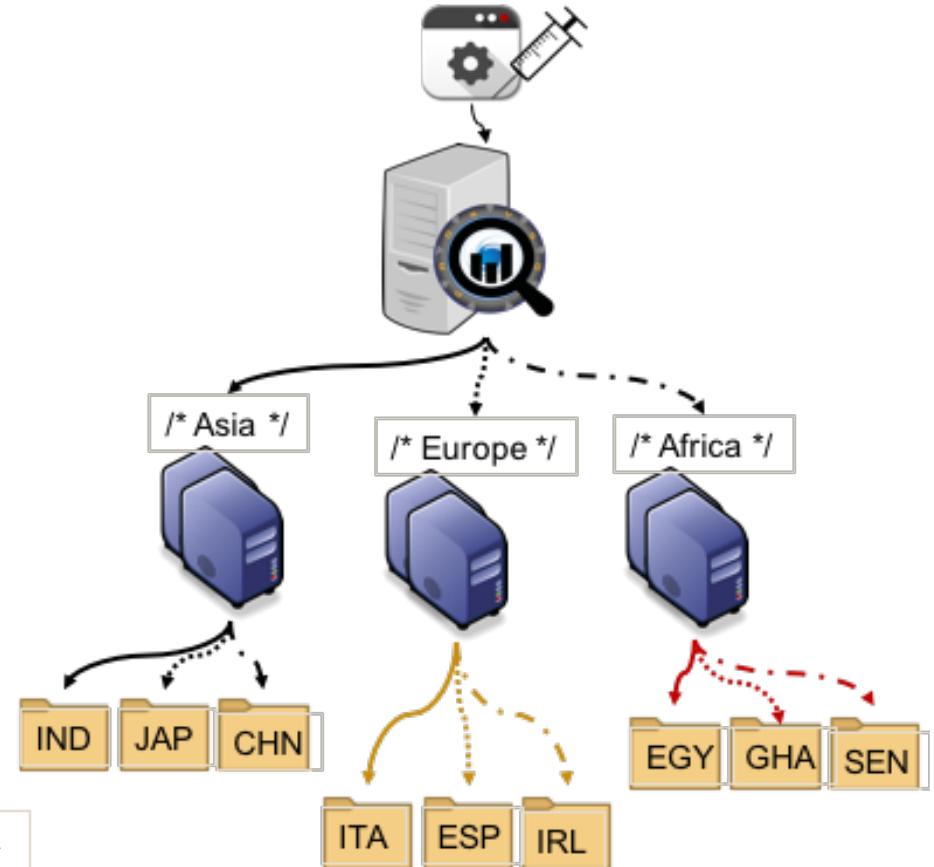
Way of Sharding



Basic By IP/Port



By parameter capture or
comment injection
Performing query rewrite



Comment Injection/ HG
Direct.
Query analysis and rewrite

Prepare MySQL

You will need:

3 hosts: 192.168.0.[5-21-231]

4 schemas: Continent X + world schema

1 user : user_shardRW

3 hostgroups: 10, 20, 30

Understand the basics

HostGroup (HG) & Replication HG

Logical group of server

Unless specify HG receives reads and writes

Properties:

id; Server list

Special Replicaton Host Group (RHG)

Manage the separation between Write and Read HG.

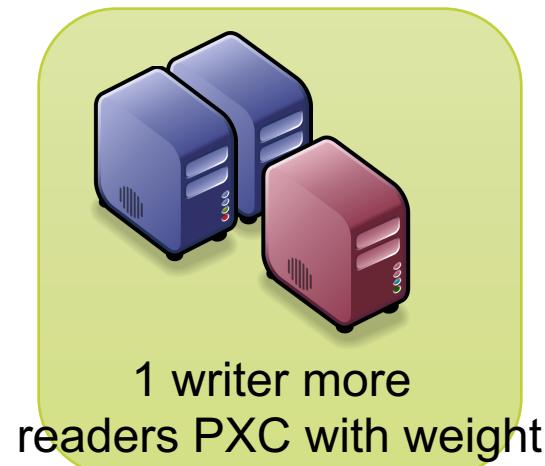
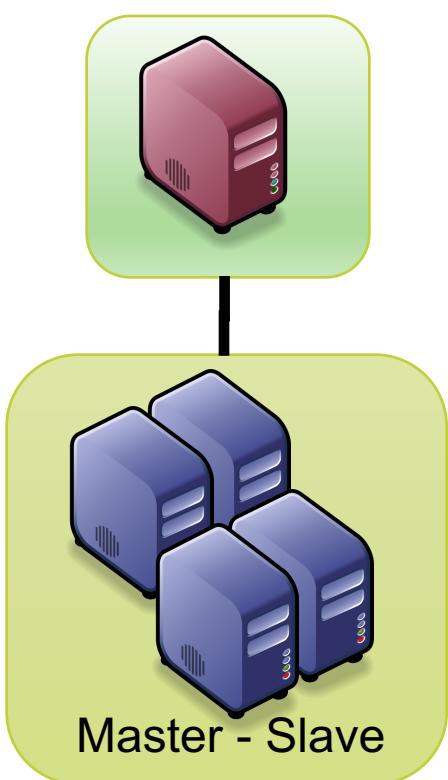
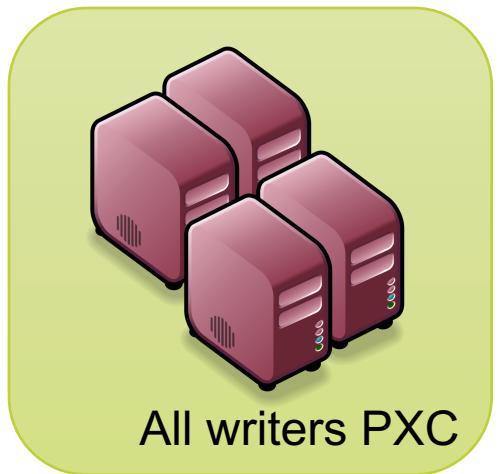
Properties:

writer_hostgroup ID

reader_hostgroup ID

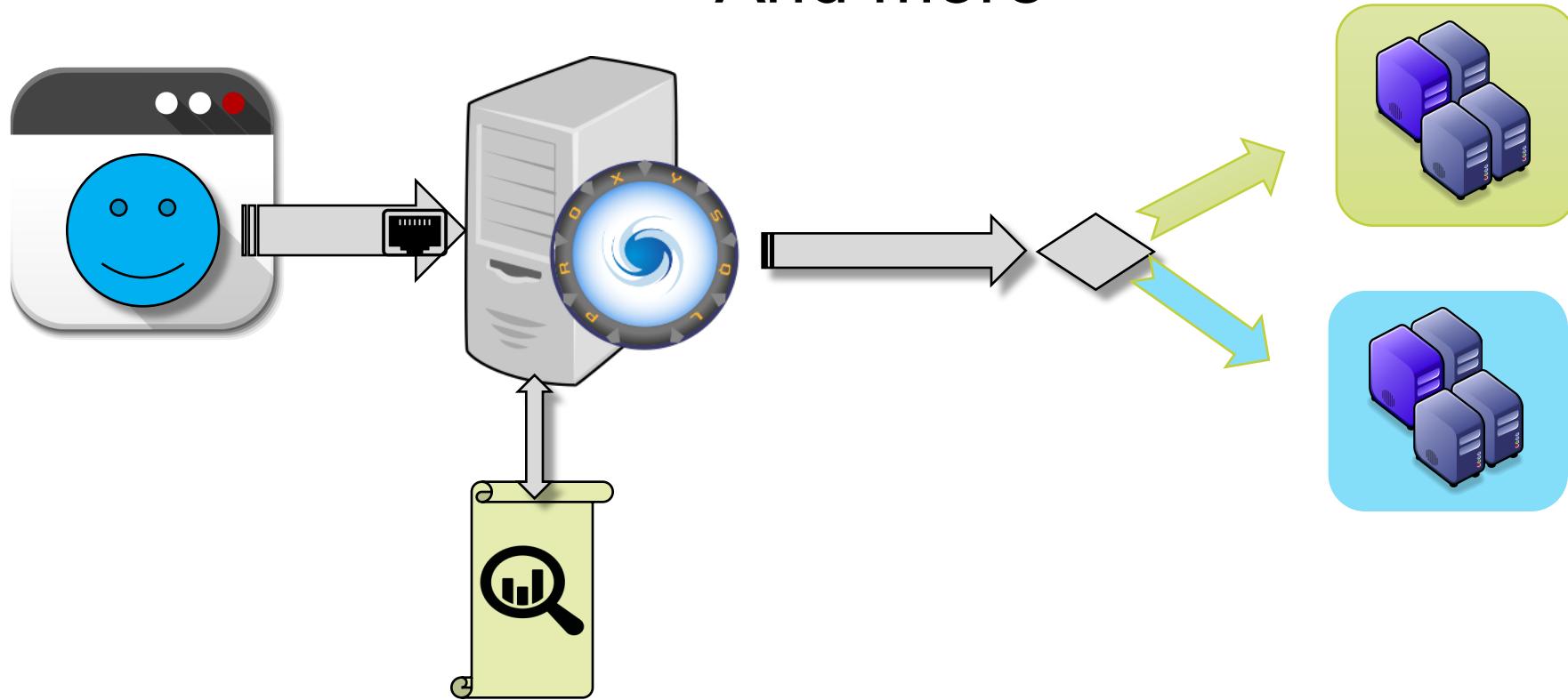
Understand the basics

HostGroup (HG) & Replication HG



What is a Query rule?

Is the logic in ProxySQL that link Users/Schema/IP/Port/SQL to (Replication) Host Group.
OR That allow SQL rewrite.
And more



Prepare MySQL

Load DB test World

Create schemas Asia|Europe|North_America|Africa

For each continent:

```
create table {continent}.City as select a.* from world.City a join Country on a.CountryCode = Country.code where Continent='{continent}' ;
```

```
create table {continent}.Country as select * from world.Country where Continent='{continent}' ;
```

End for

```
grant all on *.* to user_shardRW@'%' identified by 'test';
```

Prepare Proxysql

What you need to do is:

Set the user

```
insert into mysql_users
(username,password,active,default_hostgroup,default_schema,transaction_persistent)
values ('user_shardRW', 'test',1,10,'test',1);
LOAD MYSQL USERS TO RUNTIME;SAVE MYSQL USERS TO DISK;
```

Set variable:

```
update global_variables set Variable_Value=0 where Variable_name='mysql-
query_cache_size_MB';
update global_variables set Variable_Value=0/20 where Variable_name='mysql-
query_processor_iterations';

load mysql variables to run;save mysql variables to disk;
```

Prepare Proxysql

Create basic set of server to play with like:

```
INSERT INTO mysql_servers (hostname,hostgroup_id,port,weight)
VALUES ('192.168.0.5',10,3306,100);
```

```
INSERT INTO mysql_servers (hostname,hostgroup_id,port,weight)
VALUES ('192.168.0.21',20,3306,100);
```

```
INSERT INTO mysql_servers (hostname,hostgroup_id,port,weight)
VALUES ('192.168.0.231',30,3306,100);
```

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

Prepare Proxysql

Add Ports to play later with:

```
update global_variables set  
Variable_Value='0.0.0.0:6033;0.0.0.0:6034;0.0.0.0:6035;0.0.0.0:60  
36;' where Variable_name='mysql-interfaces';
```

save mysql variables to disk;



Exit and restart proxysql

Sharding all the logic is in:

MySQL Query Rules Table

Sharding by:

- username
- schemaname
- client_addr
- proxy_addr
- proxy_port
- match_digest
- match_pattern

Sharding Targets

Destination_hostgroup

Pointing to different HG



Replace_pattern

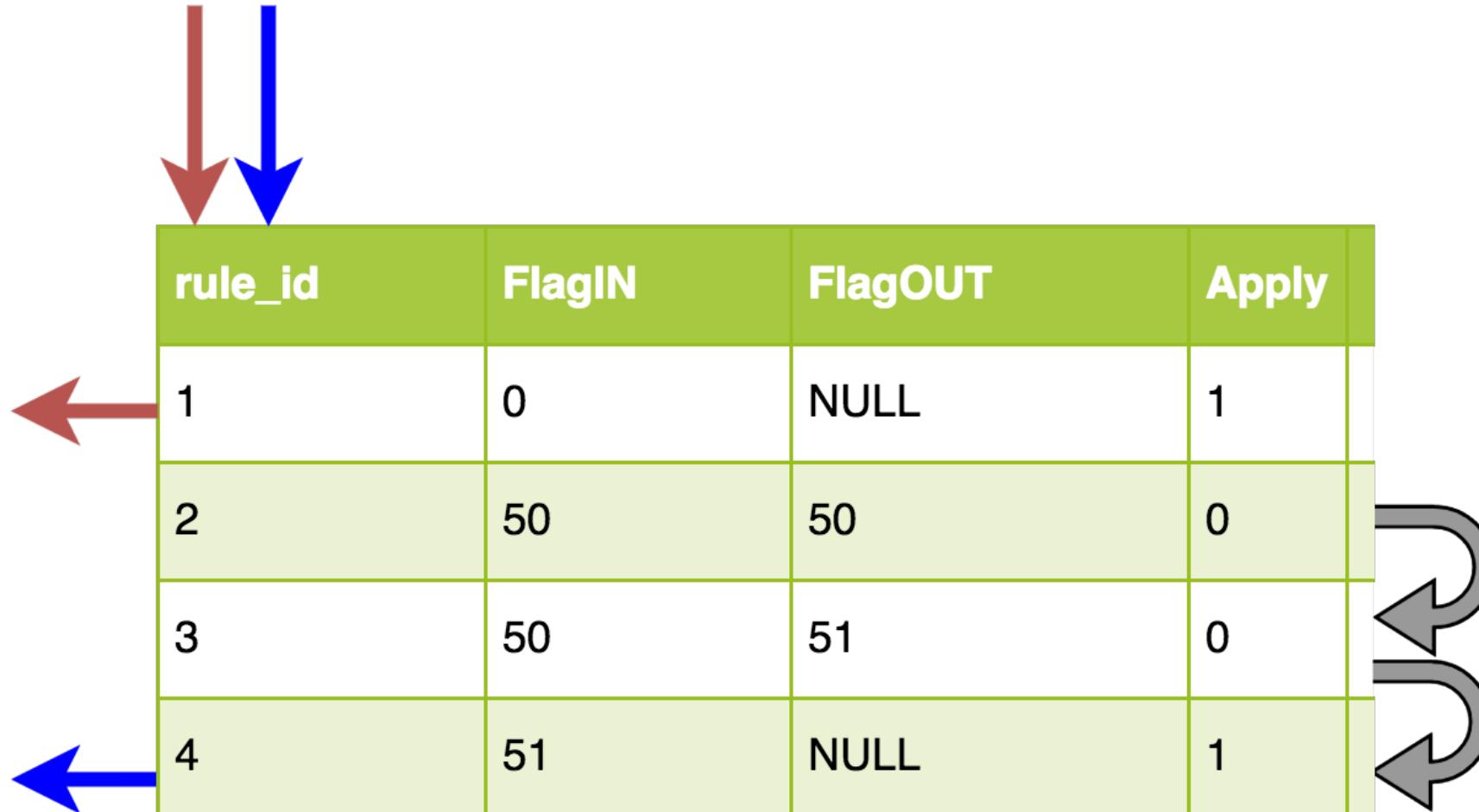
Rewriting the Query

Or if firewalling blocking the query

Sharding Examples Query rules

Sh type	Destination	rule_id	active	username	proxy_port	destination_hostgroup	re_modifiers	match_pattern	replace_pattern	apply	FlagOUT	FlagIN
Port	HG	10	1	user_shardRW	6034		10	CASELESS	NULL	NULL	1	NULL
Port	HG	11	1	user_shardRW	6035		20	CASELESS	NULL	NULL	1	NULL
Port	HG	12	1	user_shardRW	6036		30	CASELESS	NULL	NULL	1	NULL
Comment	HG	20	1	user_shardRW	NULL		10	CASELESS,GLOBAL	\\\s*shard_host_HG= *Asia\\s*\\".	NULL	0	NULL
Comment	HG	21	1	user_shardRW	NULL		20	CASELESS,GLOBAL	\\\s*shard_host_HG= *Europe\\s*\\".	NULL	0	NULL
Comment	HG	22	1	user_shardRW	NULL		20	CASELESS,GLOBAL	\\\s*shard_host_HG= *Africa\\s*\\".	NULL	0	NULL
Value	Schema	31	1	user_shardRW	NULL	NULL	CASELESS	^SELECT\\s*(.+)\\s*from\\s*world.(\\S*)\\s(.+).*Continent='(\\S*)'\\s*(\\s+.+)\$	SELECT \\1 from \\4.\\2 WHE	1	NULL	0
Comment	Schema	32	1	user_shardRW	NULL	NULL	CASELESS	"\\s*\\s*\\\\s*\\s*continent= *Asia\\s*\\".*"	NULL	0	23	0
Comment	Schema	33	1	user_shardRW	NULL	NULL	CASELESS,GLOBAL	'world.'	'Asia.'	0	23	23
Comment	Schema	34	1	user_shardRW	NULL	NULL	CASELESS,GLOBAL	"\\s*\\s*\\\\s*\\s*continent= *Europe\\s*\\".*"	NULL	0	24	0
Comment	Schema	35	1	user_shardRW	NULL	NULL	CASELESS,GLOBAL	'world.'	'Europe.'	0	24	24
Comment	Schema	36	1	user_shardRW	NULL	NULL	CASELESS,GLOBAL	"\\s*\\s*\\\\s*\\s*continent= *Africa\\s*\\".*"	NULL	0	25	0
Comment	Schema	37	1	user_shardRW	NULL	NULL	CASELESS,GLOBAL	'world.'	'Africa.'	0	25	25

Understand FlagIN FlagOUT



Sharding Examples

IP/port

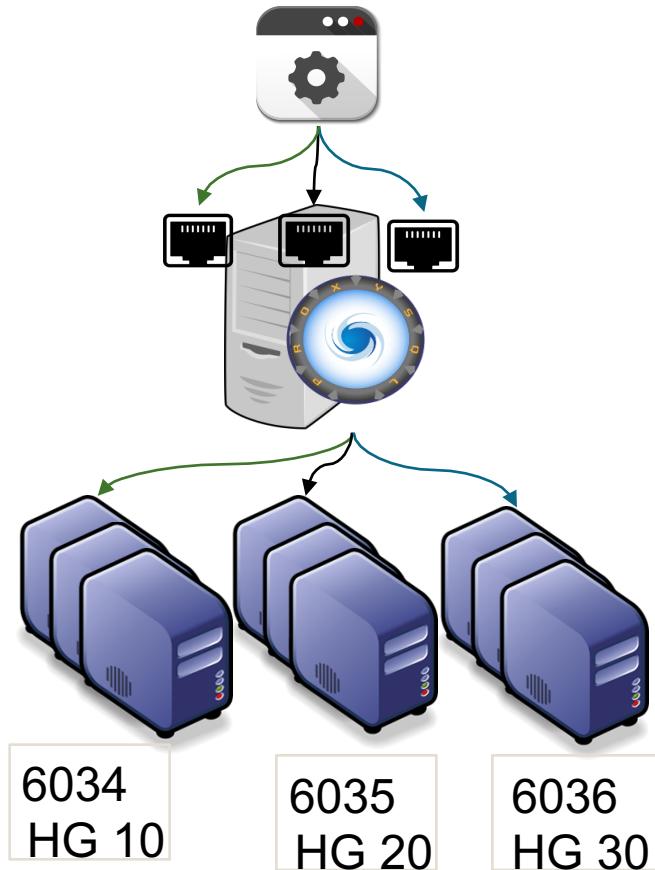
Schema using field=value

Comment -> schema or host

Use combination of host/schema

Sharding Examples

IP/port



Basic By IP/Port

Sharding Examples – IP or Port settings

For \$port in 6034 6035 6036 do

```
INSERT INTO mysql_query_rules
(rule_id,active,username,proxy_port,destination_hostgroup,apply)
VALUES (10/11/12,1,'user_shardRW' , $port,10/20/30,1);
```

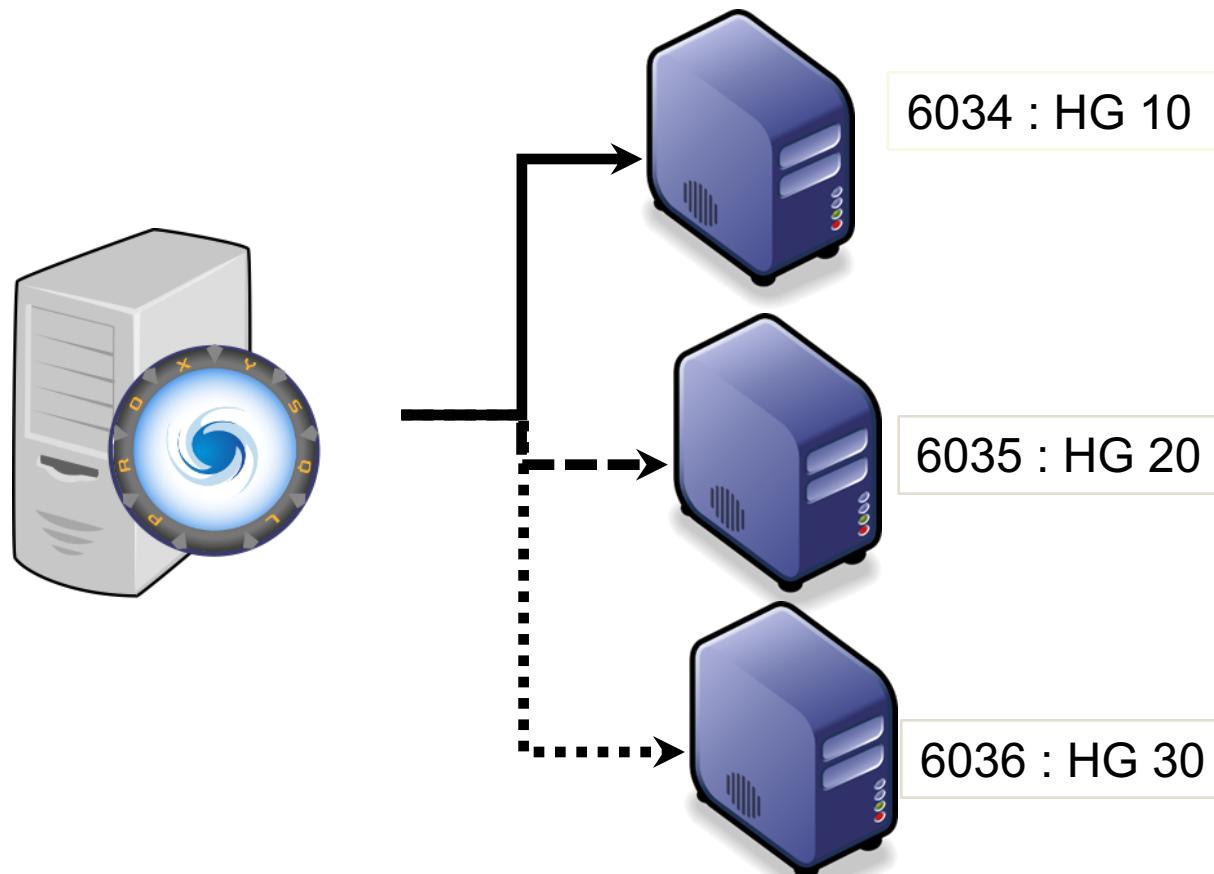
Done

LOAD MYSQL QUERY RULES TO RUNTIME ; SAVE MYSQL QUERY RULES TO DISK;

Sh type	Destination	rule_id	active	username	proxy_port	destination_hostgroup	re_modifiers	match_pattern	replace_pattern	apply	FlagOUT	FlagIN
Port	HG	10	1	'user_shardRW'	6034		10 CASELESS	NULL	NULL	1	NULL	0
Port	HG	11	1	'user_shardRW'	6035		20 CASELESS	NULL	NULL	1	NULL	0
Port	HG	12	1	'user_shardRW'	6036		30 CASELESS	NULL	NULL	1	NULL	0

Sharding Examples – IP or Port Result

rule_id	hg	username	schemaname	flagIN	flagOUT	apply	proxy_addr	proxy_port
41	10	user_shardRW	NULL	0	NULL	1	NULL	6034
42	20	user_shardRW	NULL	0	NULL	1	NULL	6035
43	30	user_shardRW	NULL	0	NULL	1	NULL	6036

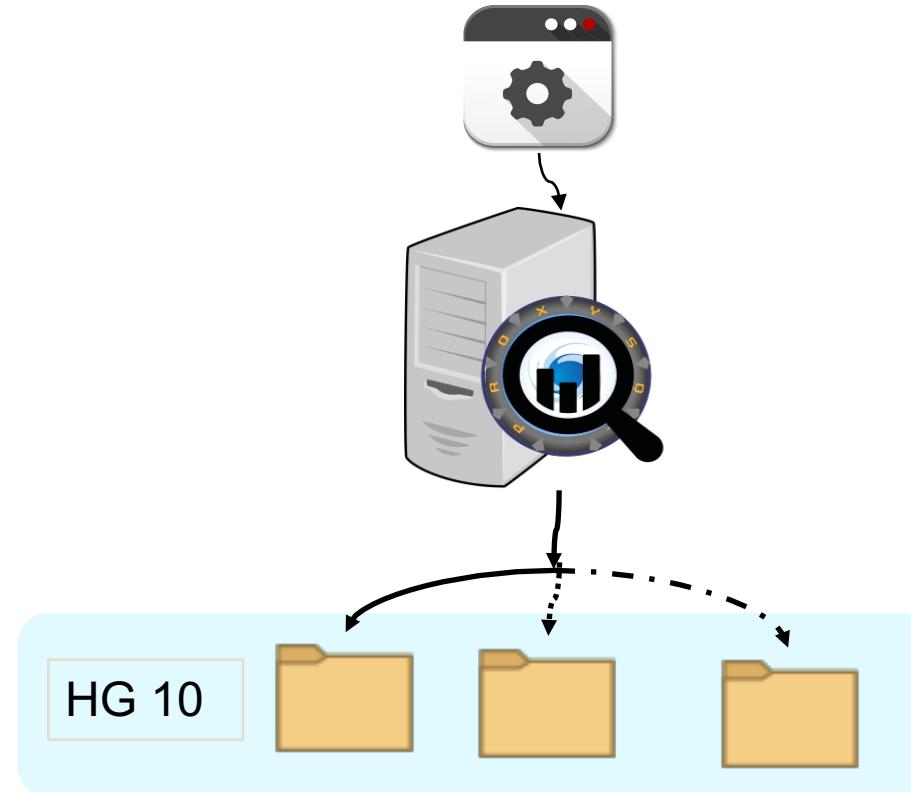


```
for port in 6034 6035 6036;do  
echo "---- $port ---"; mysql -  
uuser_shardRW -ptest -h  
192.168.0.11 -P $port -e "show  
global variables like  
'wsrep_node_name';";done  
  
---- 6034/5/6 ---  
+-----+-----+  
| Variable_name | Value |  
+-----+-----+  
| wsrep_node_name | node1/2/3  
+-----+-----+
```



Sharding Examples

Schema using
field=value
Query ReWrite



By parameter
analysis and
rewrite

Sharding Examples Schema – Value - How

Define a simple query rule using regular expression

Query: `SELECT name,population from world.City WHERE Continent='Asia' and name like 'B%' order by population desc limit 1;`

match_pattern:

`"^SELECT\s*(.*)\s*from\s*world.(\S*)\s*(.*) .*Continent='(\S*)'\s*(\S*.*)$"`

Replace_pattern: `SELECT \1 from \4.\2 WHERE 1=1 \5`

Value	Schema	31	1:user_shardRW	NULL	NULL	CASELESS	<code>^SELECT\s*(.*)\s*from\s*world.(\S*)\s*(.*) .*Continent='(\S*)'\s*(\S*.*)\$</code>	<code>:SELECT \1 from \4.\2 WHERE 1=1 \5</code>	1:NULL	0
-------	--------	----	----------------	------	------	----------	--	---	--------	---

Sharding Examples Schema - Value

ProxySQL can read your query and identify a value

```
SELECT name, population from world.City WHERE  
Continent='Asia' and CountryCode='IND' order by population  
desc limit 1;
```

Become

```
SELECT name, population from Asia.City WHERE ?=? and  
CountryCode=? order by population desc limit ?
```

Sharding Examples Schema – Value - How

Given a query

Original: `SELECT name, population from world.City WHERE Continent='Asia' and name like 'B%' order by population desc limit 1`

Result: `SELECT name, population from Asia.City WHERE ?=? and name like ? order by population desc limit ?`

Sharding Examples Schema – Value

Test

```
for $continent in Asia Europe Africa  
North_America ;do  
echo "---- $continent ---";  
mysql -uuser_shardRW -ptest -h 192.168.0.11 -P6033  
-e "  
SELECT name, population  
from world.City WHERE Continent='$continent' and  
name like 'B%'  
order by population desc limit 1;"  
;done
```

Sharding Examples Schema: Value - Check

To check what ProxySQL processed:

```
    hits: 4 ←Because I run the script for 4 continents
    rule_id: 31
    match_pattern:
^SELECT\s*(.*\s*)from\s*world.(\S*)\s(.*)\.*Continent='(\S*)'\s*(\s*.*)
$  
replace_pattern: SELECT \1 from \4.\2 WHERE 1=1 \5
    cache_ttl: NULL
    apply: 1
select active,hits, mysql_query_rules.rule_id, match_digest, match_pattern, replace_pattern, apply
FROM mysql_query_rules NATURAL JOIN stats.stats_mysql_query_rules ORDER BY
mysql_query_rules.rule_id\G
```

Sharding Examples

Comment Injection

Causing:

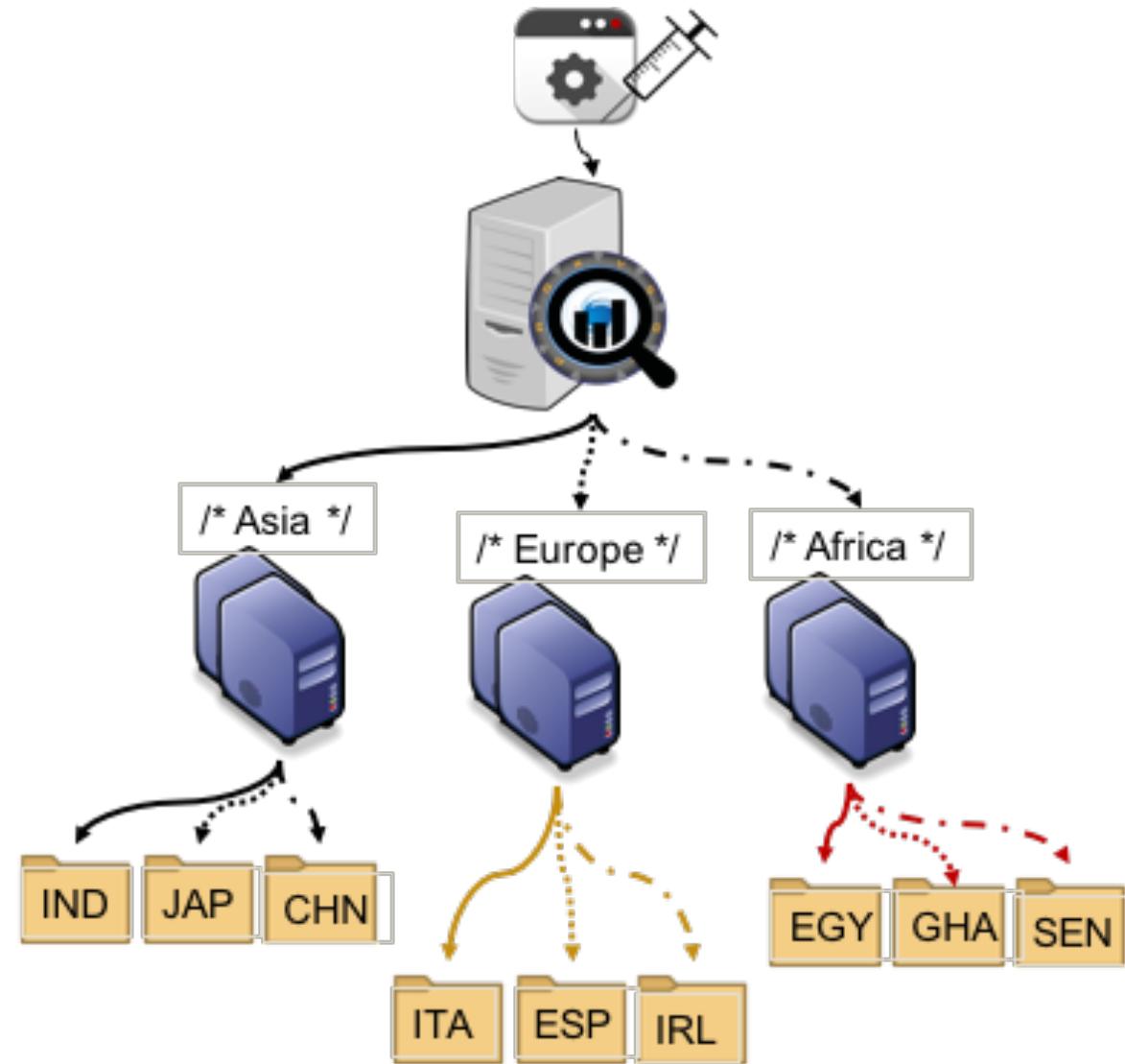
- Query ReWrite

Or

- HG redirection

Or

- Both



Sharding Examples - Comment

Comments are the easiest thing to implement in the application

QueryRules chains (FlagIN – FlagOUT)

What is a comment in MYSQL:

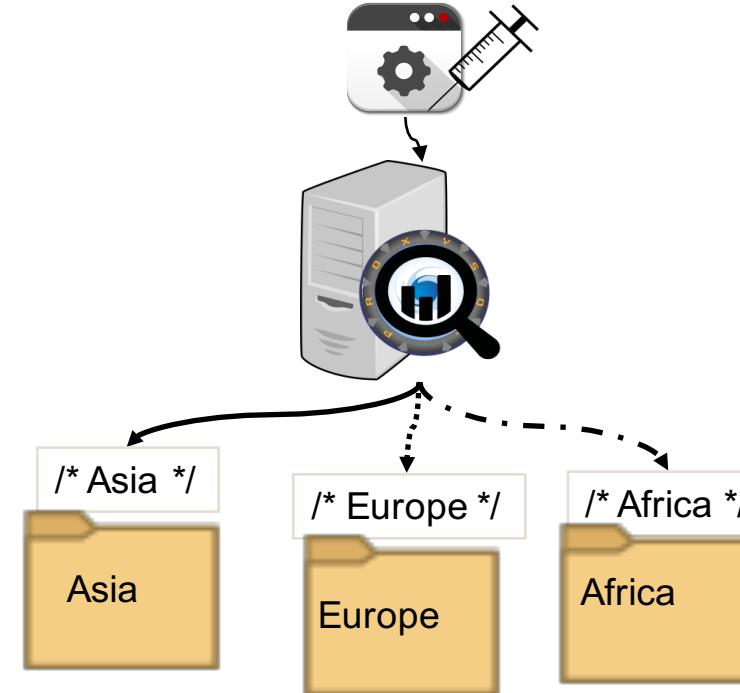
Select name /* I am a comment*/ from world.City;

If testing from mysql console do not forget the flag

“-C”

Sharding Examples - Comment

Use comments to hit Different schemas



Sharding Examples – Comment - How

Given a query:

Original : `Select /* continent=Asia */ * from world.Country
join world.City on
world.City.CountryCode=world.Country.Code where
Country.code='IND'`

Transformed: `Select * from Asia.Country join Asia.City on
Asia.City.CountryCode=Asia.Country.Code where
Country.code=?`

Sharding Examples - Comment

2 Rules:

```
INSERT INTO mysql_query_rules
(rule_id,active,username,match pattern,replace pattern,apply,FlagOUT,FlagIN) VALUES
(32,1,'user_shardRW','\S*\s*/\*\s*continent=.*Asia\s*\*.*',null,0,23,0);
```

```
INSERT INTO mysql_query_rules
(rule_id,active,username,match pattern,replace pattern,apply,FlagIN,FlagOUT) VALUES (33,1,'user_shardRW','world.','Asia.',0,23,23);
```

Comment	Schema	32	1:user_shardRW	NULL	NULL	CASELESS	"\S*\s*/*\s*continent=.*Asia\s**.*"	NULL	0	23	0
Comment	Schema	33	1:user_shardRW	NULL	NULL	CASELESS,GLOBAL	'world.'	'Asia.'	0	23	23
Comment	Schema	34	1:user_shardRW	NULL	NULL	CASELESS,GLOBAL	"\S*\s*/*\s*continent=.*Europe\s**.*"	NULL	0	24	0
Comment	Schema	35	1:user_shardRW	NULL	NULL	CASELESS,GLOBAL	'world.'	'Europe.'	0	24	24
Comment	Schema	36	1:user_shardRW	NULL	NULL	CASELESS,GLOBAL	"\S*\s*/*\s*continent=.*Africa\s**.*"	NULL	0	25	0
Comment	Schema	37	1:user_shardRW	NULL	NULL	CASELESS,GLOBAL	'world.'	'Africa.'	0	25	25

Sharding Examples – Comment - How

```
hits: 1
rule_id: 32
match_pattern: /\*\s*continent=.*Asia\s*
replace_pattern: NULL
FlagIN: 0
FlagOUT: 23
```

```
hits: 1 (old versions (1.3.x: 4))
rule_id: 33
match_pattern: world.
replace_pattern: Asia.
FlagIN: 23
FlagOUT: 23
```

```
Select hits, mysql_query_rules.rule_id, match_digest, match_pattern, replace_pattern FROM mysql_query_rules NATURAL JOIN
stats.stats_mysql_query_rules ORDER BY mysql_query_rules.rule_id\G
```

Sharding Examples – Comment -How

Two rules involved. ID: 32, 33

re_modifiers: **CASELESS,GLOBAL** (v1.4.0)*

First rule

FlagIN: 0 FlagOUT: 23

match_pattern: `\^*\s*continent=.*Asia\s*\``

replace_pattern,apply: NULL

Second rule

FlagIN: 23 FlagOUT: 23

match_pattern: `world.`

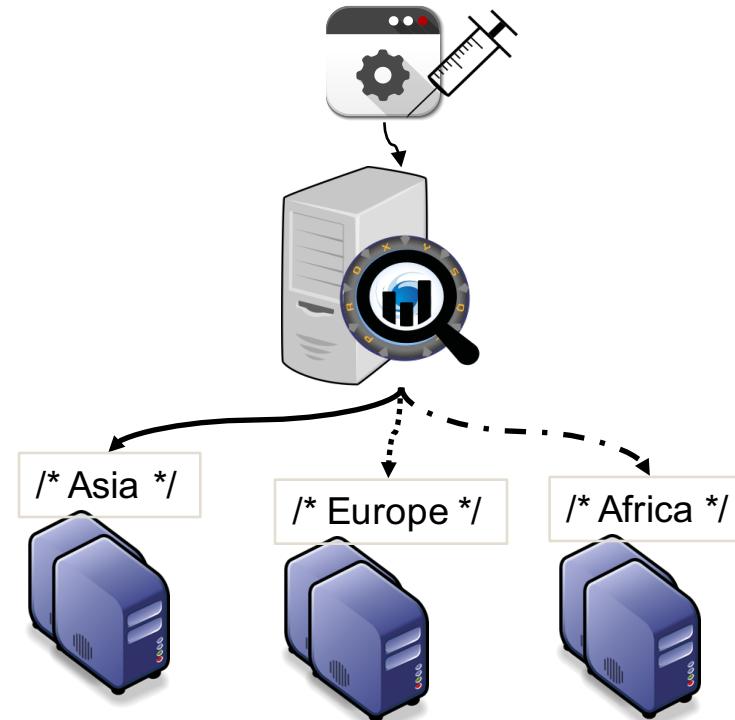
replace_pattern,apply: `Asia.`

*

Unless v1.4.0 and following you need to: update global_variables set Variable_Value=20 where Variable_name='mysql-query_processor_iterations';

Sharding Examples – Comment – To HG

Point to a HG directly is even easier



Sharding Examples – Comment – To HG

Point to a HG directly is even easier

The scope is that whatever match the rule, should go to a defined HG.

```
INSERT INTO mysql_query_rules (rule_id, active, username,
match_pattern, destination_hostgroup,apply) VALUES
(10,1,'user_shardRW','\/*\s*shard_host_HG=.*Europe\s*\*.*',
10,0);
```

Comment	HG	20	1:user_shardRW	NULL		10	CASELESS,GLOBAL	\/*\s*shard_host_HG=.*Asia\s**.	NULL	0	NULL	0
Comment	HG	21	1:user_shardRW	NULL		20	CASELESS,GLOBAL	\/*\s*shard_host_HG=.*Europe\s**.	NULL	0	NULL	0
Comment	HG	22	1:user_shardRW	NULL		20	CASELESS,GLOBAL	\/*\s*shard_host_HG=.*Africa\s**.	NULL	0	NULL	0

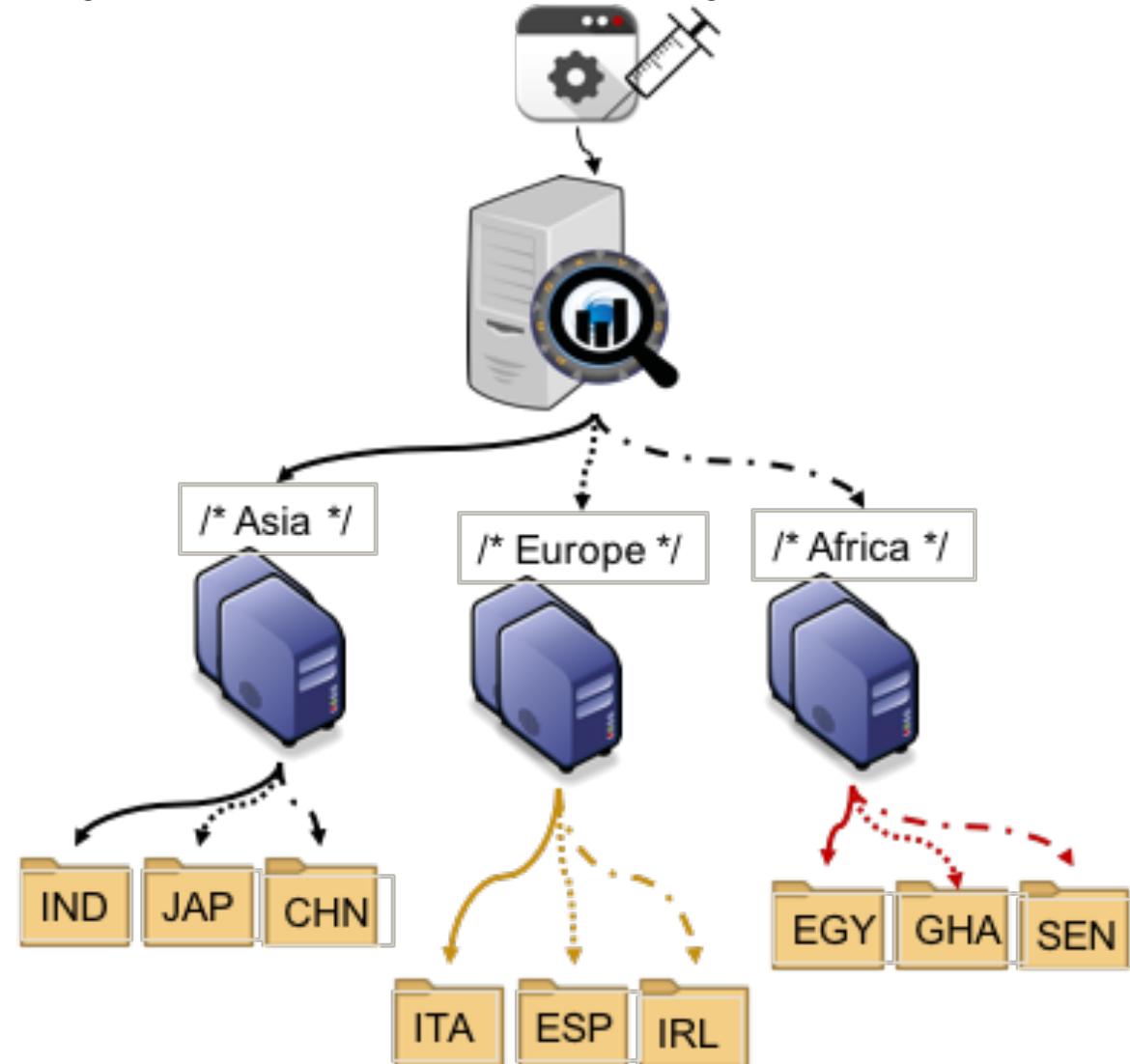
Sharding Examples – Comment - How

```
hits: 1
rule_id: 20
match_pattern: /\*\s*continent=.*Asia\s*
replace_pattern: NULL
FlagIN: 0
FlagOUT: NULL
destination_hostgroup: 10
apply: 0
```

```
Select hits, mysql_query_rules.rule_id, match_digest, match_pattern, replace_pattern FROM mysql_query_rules NATURAL JOIN
stats.stats_mysql_query_rules ORDER BY mysql_query_rules.rule_id\G
```

Sharding Examples – Comment: HG + Schema

Use both, by HG and Query Rewrite



Sharding Examples – Comment: HG + Schema

Finally is possible to mix the two things.

I want to use:

Asia on Server 192.168.1.5 -> HG 10, points to a COUNTRY schema

Europe on Server 192.168.1.6 -> HG 20, points to a COUNTRY schema

Africa on Server 192.168.1.7 -> HG 30, points to a COUNTRY schema

```
Select /* shard_host_HG=Asia */ /* country=India */ City.Name,  
City.Population from world.Country join world.City on  
world.City.CountryCode=world.Country.Code limit 5;
```

Sharding Examples - Comment

2 Rules for country :

```
INSERT INTO mysql_query_rules
(rule_id,active,username,match pattern,replace pattern,apply,FlagOUT,FlagIN) VALUES
(32,1,'user_shardRW','\S*\s*/\*\s*country=.*India\s*\*.*',null,0,23,0)
;
```

```
INSERT INTO mysql_query_rules
(rule_id,active,username,match pattern,replace pattern,apply,FlagIN,FlagOUT) VALUES (33,1,'user_shardRW','world.','India.',0,23,23);
```

Sharding Examples – Comment - How

Three different rules to achieve the result. Order is important

QR ID: 10

match_pattern: `* \s*continent=.*Asia\s*`

Target: HG

Apply: 0

QR ID: 32

match_pattern: `* \s*country=.*India\s*`

Target: Query Rule 33

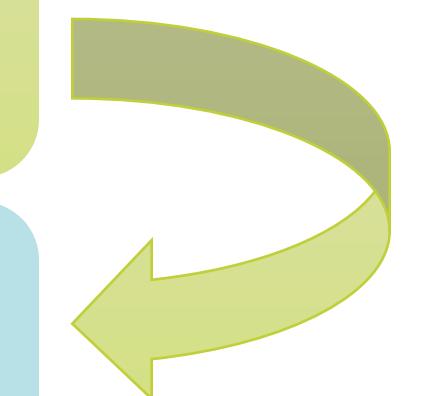
Apply: 0

QR ID: 33

match_pattern: `world.`

Target: Query Rewrite to Country schema

Apply: 0



Real case

Problem

- Customer must reduce the number of shards
- Aggregate several schemas in multitenant DB
- Collapse several servers in few
- No code change (as usual) or minimal
- Java with Hibernate

Real case

Solution

- Inject limited comment in the query
 - /*shard_ip=x.x.x.x,db=my_db,grp_id=123 */
- Create Query rules to manage
 - 4 rules to manage all + 1 for each additional server

(HG)

Real case

The query:

```
/* shard_ip=10.11.4.80,db=grp_aaaa.grp_id=23456 */  
SELECT * FROM grp_1211.users where pippo like tt;
```

Real case

The QRs:

```
INSERT INTO mysql_query_rules (rule_id,active,username,match_pattern,apply,FlagIN,FlagOUT)
VALUES (10,1,'apptest1','where',0,0,20);
```

```
INSERT INTO mysql_query_rules
(rule_id,active,username,match_pattern,replace_pattern,apply,FlagIN,FlagOUT) VALUES
(15,1,'apptest1',"(\/*.*,)grp_id=(\d*)\s*\*\/\s*(.*)\.*$","\\1 */ \\3 WHERE grp_id=\\2;",0,0,25);
```

```
INSERT INTO mysql_query_rules
(rule_id,active,username,match_pattern,replace_pattern,apply,FlagIN,FlagOUT) VALUES
(18,1,'apptest1',"(\/*.*,)grp_id=(\d*)\s*\*\/(.*)\s*(where)\s*(.*)\.*$","\\1 */ \\3 WHERE
grp_id=\\2 AND \\5;",0,20,25);
```

```
INSERT INTO mysql_query_rules
(rule_id,active,username,match_pattern,replace_pattern,apply,FlagIN,FlagOUT) VALUES
(22,1,'apptest1',"^(\/*.*,)db=(\S*),\s*\*\/\s*(.*(FROM|UPDATE|INSERT INTO))\s*(.*)\.(.*)","\\1
*/ \\3 \\2.\\6",0,25,25);
```

```
INSERT INTO mysql_query_rules
(rule_id,active,username,match_pattern,destination_hostgroup,apply,FlagIN,FlagOUT) VALUES
(30,1,'apptest1',"shard_ip=10.11.4.79",20,1,25,NULL);
```

Real case

Starting query:

```
/* shard_ip=10.11.4.80,db=grp_aaaa,grp_id=23456 */  
SELECT * FROM grp_1211.users where pippo like tt;
```

Become:

```
/* shard_ip=10.11.4.80,db=grp_aaaa, */ SELECT * FROM  
grp_1211.users where grp_id=23456 AND pippo like tt;
```

```
/* shard_ip=10.11.4.80, */ SELECT * FROM  
grp_aaaa.users WHERE grp_id=23456 AND pippo like tt;
```

Ready to be dispatch to final HG with rules above id 30

Real case

— №1 —
**REALITY
SUCKS**
———— * —————

Obviously if a customer can screw your perfect plan
... he will.

- Try to use NON standard SQL like:
INSERT... SET (MySQL extension)
- Omit the schema name in the SQL

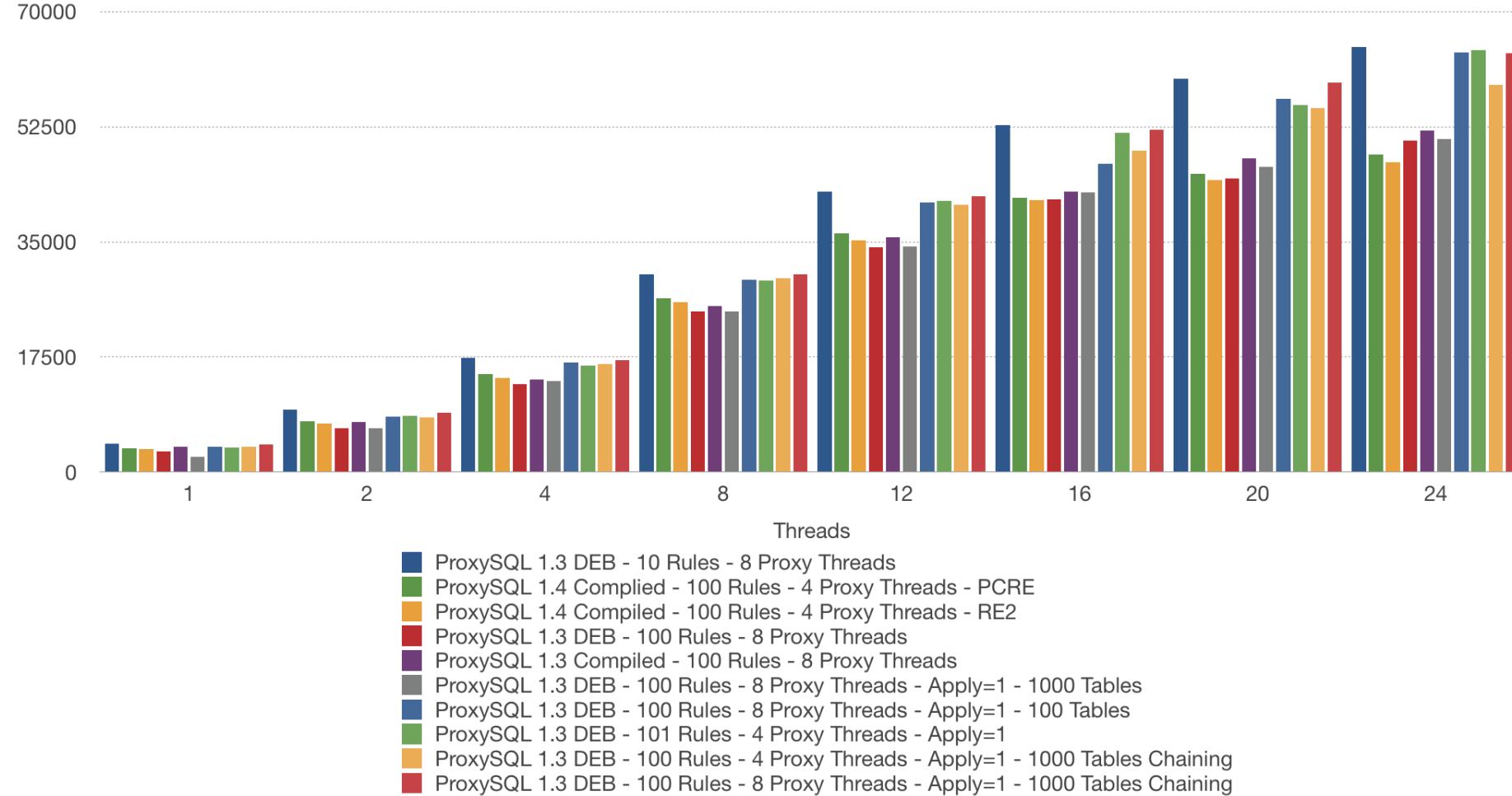
It is NOT a kind of magic

There is no magic solution for sharding.
ProxySQL is a tool to help not a sharding mechanism.

- Per port/IP (require to be define at infrastructure design time)
- Query filtering/rewrite (may impact performance if many QR, and require application to be aware of the shards)
- Cannot pass variables cross QR
- Queries are NOT spanning on multiple HG.
- Cannot aggregate resultsets from different shards



Cost in Execution (trx in 5 minutes)



Review the blogs: <https://www.percona.com/blog/2017/04/12/proxysql-applying-and-chaining-the-rules/>

<https://www.percona.com/blog/2017/04/10/proxysql-rules-do-i-have-too-many/>

Brief summary



Multiple element to use to shard:
User/default schema/IP-Port/value/Hint

Shards can be all pointing to same HG different schemas;
pointing to different HG or a Mix

Plan Your QR and chains to avoid performance degradation

To Know more



<http://www.proxysql.com/> ProxySQL

<http://www.pcre.org/> Perl Compatible Regular Expressions

<https://goo.gl/cMkngr> Sharding with ProxySQL

<https://goo.gl/vozsI> stop an offending query with ProxySQL

<https://goo.gl/OC8RbN> ProxySQL Rules: Do I Have Too Many?

<https://goo.gl/rzYDr7> Applying and Chaining the Rules

<https://www.percona.com/blog>

Some personal thanks to:

René Cannaò for ProxySQL

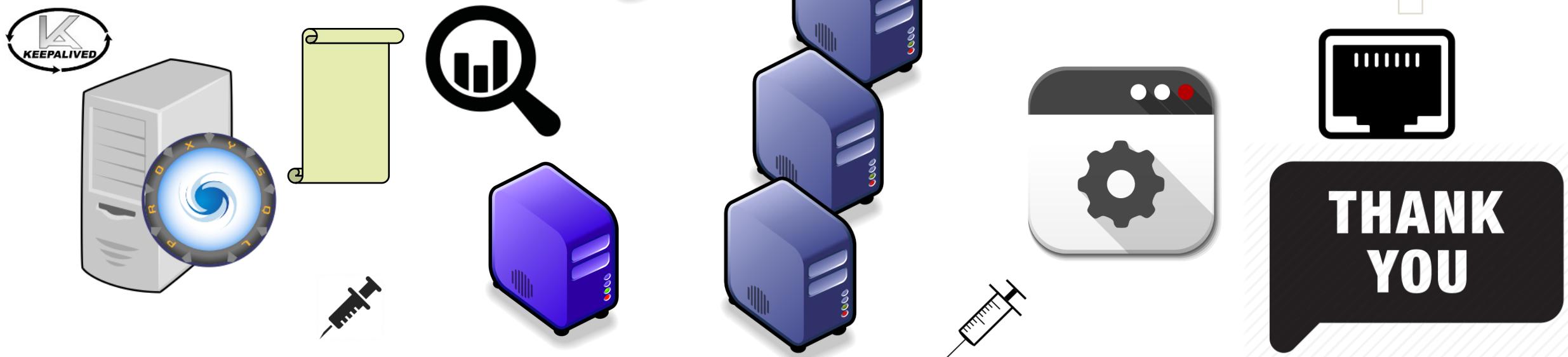
Tibor Köröcz for performance
Tests

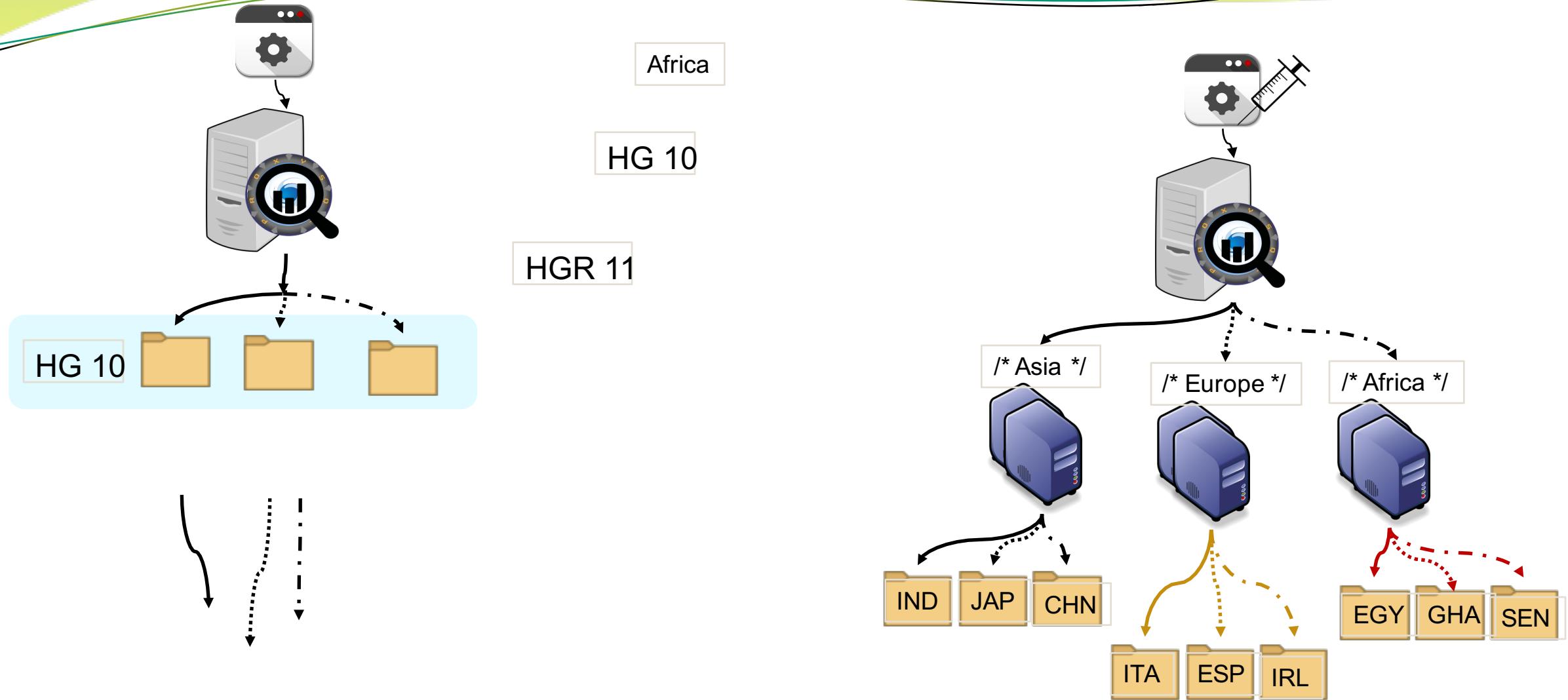


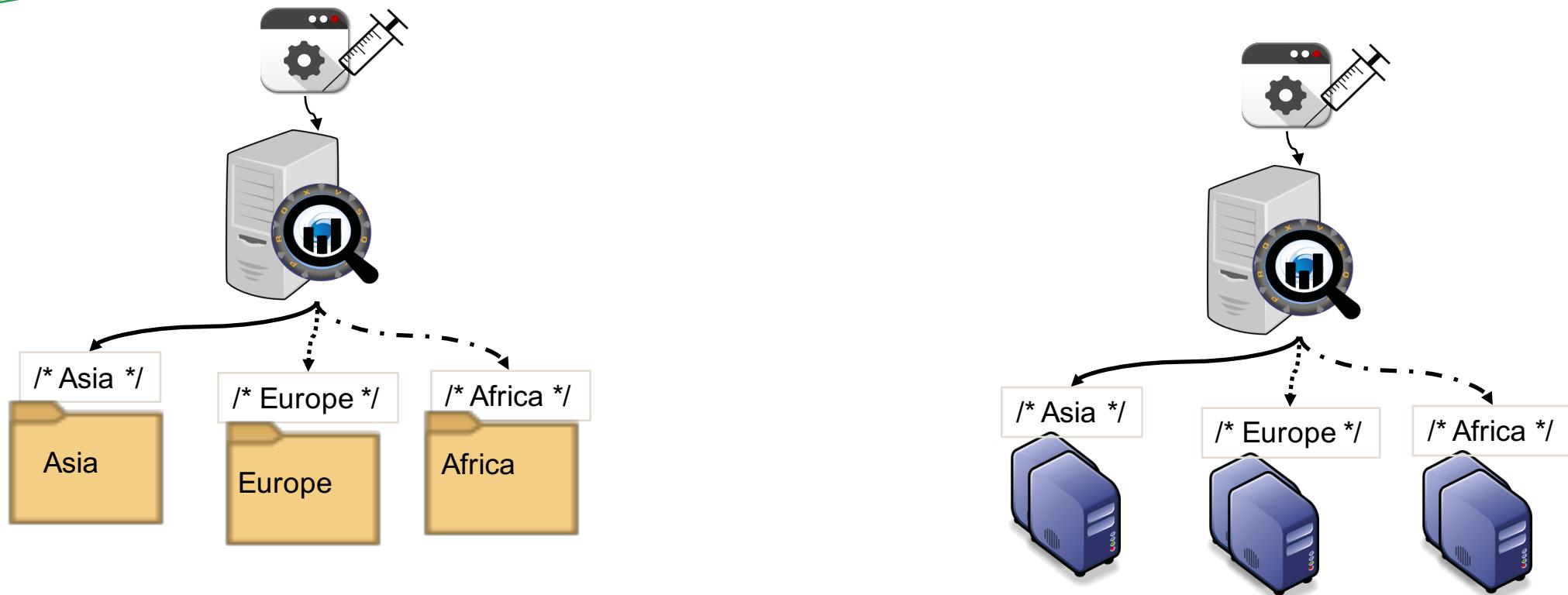
THANK
YOU

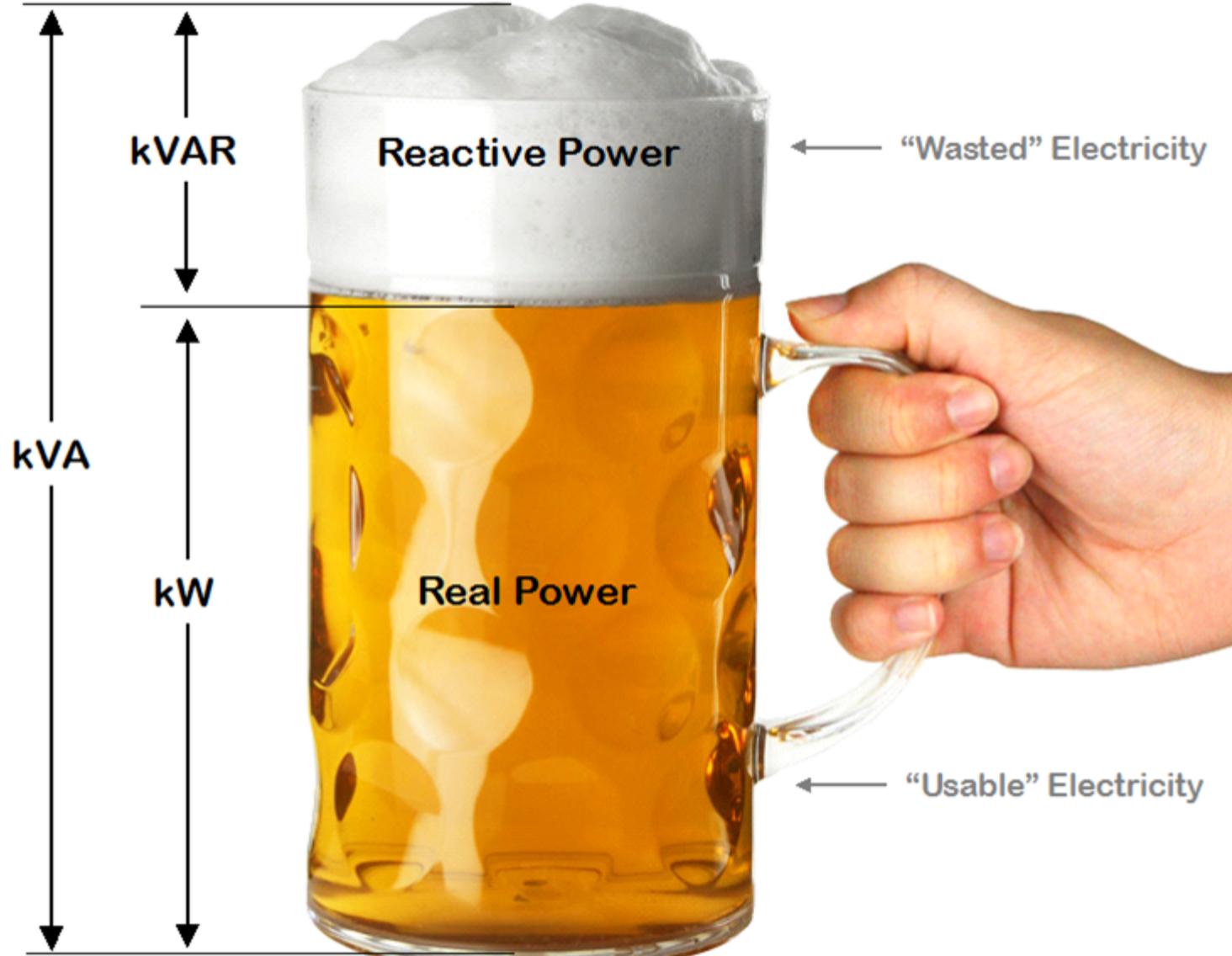
Integration with MHA

MHA









A large word cloud centered on the words "Thank You" in various languages. The words are rendered in different sizes and colors, primarily in shades of red, brown, and black. The background features a light green gradient at the top.

The word cloud includes the following words:

- Shukriya Buznyg
- ThintKo TapadhLeibh
- Blagodaram AsanteSana
- Matondo Waita
- Mercé TesekkurEdirem
- Taiku Toda
- Dakujem Batba
- Terimakasih Anikie
- Najistuke Nizzik air
- Faleminderit Aguyje
- Efharisto Blagodaria
- Gràcies Grandmercé
- Hvala Dhanyavadalu
- Arigatō Subpay
- Obrigado Spas
- Salamat Esteriksto Lankevyl
- Bayarlalaa Wado
- Obrigado Diky
- Sahamnida Marahaba
- KamSah Hamnida Takkfiri
- Spacíbo Asante
- Dankē Miigwetch
- Gracie Cámón
- Danke Misadtra
- Gracias Palías
- Dankie Akpe
- Tenki ANIKIE
- Gratias Ago Gracie
- GratiasAgo Tamirt
- GraciasAimus Akiba
- BarakAllahufiik Grazzi Nouari
- Takk Enkosi
- Murakoze Bedankt Zikomo
- FaafetaiLava Ngiyabonga
- Dhanyavaad Trugarez
- Aabhar TeşekkürEderim Sobodi
- Barkal AabharAahe
- Kiitos Mauruuru KyayTzuTimPaTe
- Grazie Chokrane GoRaibhMaithAgat
- Xièxie Dziakuju Mochchakeram
- Mèsi BarakAllahufiik
- Aciū Dhanyabead
- Dankie KopKhunKrap
- Tak Tau Nanni
- Sagolun Sagolun
- Motashakkeram KopKhunKha



Rate My Session!

Schedule
Timezone: Europe/Berlin +02:00

MON	TUE	WED
3	4	5

11:20

ClickHouse: High-Performance
Distributed
11:30 - 12:10, Matterhorn 2

TAP THE SESSION

Introducing gh-ost: triggerless, painless, trusted online schema migrations
11:20 - 12:10, Matterhorn 2

MongoDB query monitoring
11:30 - 12:10, Matterhorn 3

MySQL, Load Balancers - MaxScale, ProxySQL, HAProxy, MySQL, Router
Bamp, nginx - a close up look
11:30 - 12:10, Zurich 1

Securing your MySQL/MariaDB data
11:30 - 12:10, Zurich 2

MySQL, and Depth: A tale of two friends

← Details

Introducing gh-ost: triggerless, painless, trusted online schema migrations

⌚ 11:20 → 12:10
📍 Matterhorn 2

Rate & Review

TAP TO RATE & REVIEW

gh-ost is a MySQL tool which changes the paradigm of MySQL, online schema changes, designed to overcome today's limitations and difficulties in online migrations.

SPAKERS

 Shlomi Noach
Senior Infrastructure Engineer
GitHub

 Tom Krämer
Sr. Database Infrastructure Engg.
GitHub

Rate & Review

Tap a star to rate

☆ ☆ ☆ ☆ ☆

Feedback (optional)

Anonymously

SUBMIT

Contacts

To contact Me



Marco.tusa@percona.com



marcotusa@tusacentral.net

To follow me



<http://www.tusacentral.net/>



<http://www.percona.com/blog/>



<https://www.facebook.com/marco.tusa.94>



@marcotusa



<http://it.linkedin.com/in/marcotusa/>



“Consulting = No mission refused!”