



MySQL Cluster und MySQL Proxy

Alles Online

- Diese Slides gibt es auch unter:
 - <http://rt.fm/s4p>

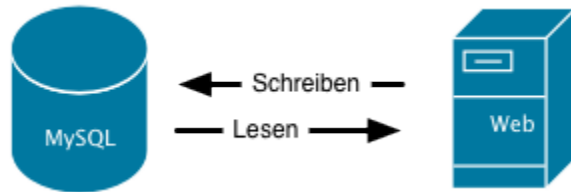


Agenda

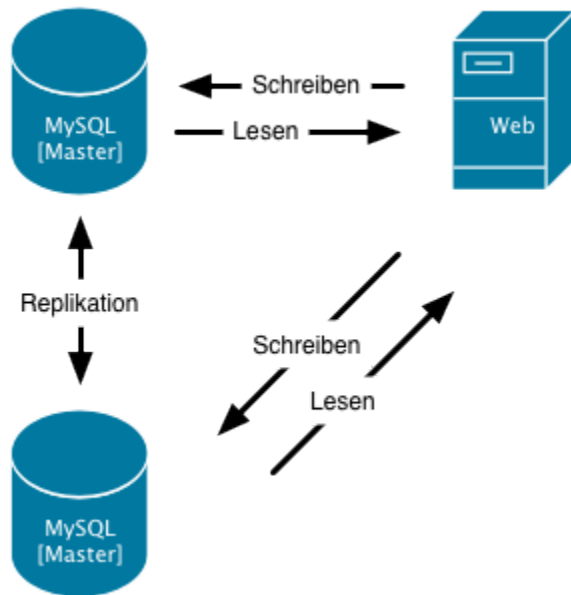
- (Don't) Panic
 - Web- und MySQL-Server
 - MySQL Master-Master Cluster
 - MySQL Proxy und Cluster
- MySQL Master-Slave/Master
 - Konfiguration
 - Replication Manager
- MySQL Proxy
 - Aufgaben
 - LUA
 - Konfiguration
- Live Demo

(Don't) Panic

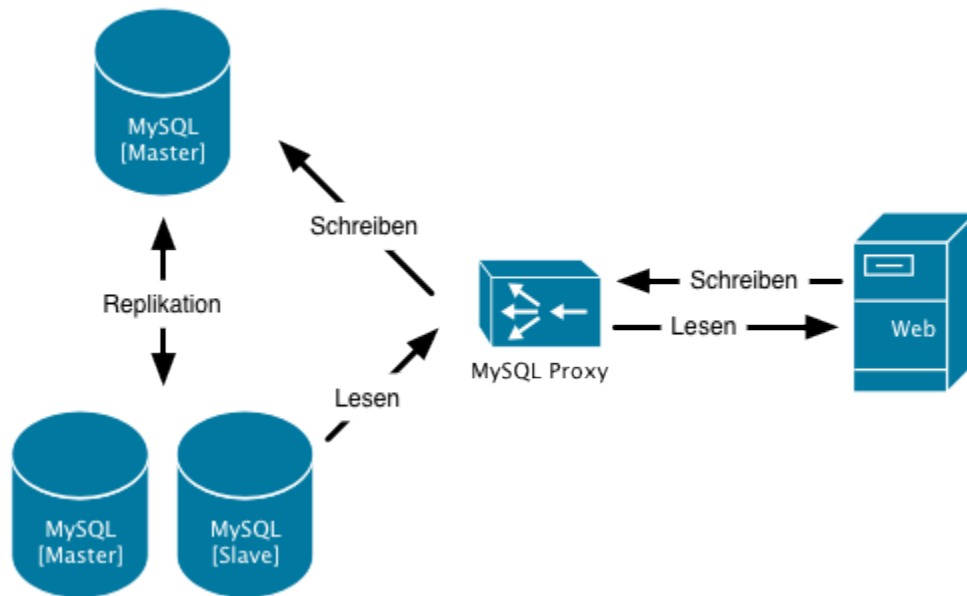
Web- und MySQL-Server



MySQL Master-Master Cluster



MySQL Proxy und Cluster



MySQL Master-Slave / Master

Pre-Konfiguration

- Empfehlung:
 - leere MySQL Server ohne Datenbank
 - ggf. Datenbank Sicherung und auf allen Cluster-Nodes einspielen

```
$ mysqldump --opt --allow-keywords --add-drop-database  
--hex-blob --quote-names --lock-tables  
<dbname>
```

Konfiguration - my.cnf

```
[mysql]
# Server IDs should be consecutive (1, 2, 3, ...)
# among all master servers
server_id          = 1
# The increment should be set to the number of
# master servers involved (default: 1)
auto_increment_increment = 2
# The offset should be set to the local server-id (default: 1)
auto_increment_offset = 1
# Using those, the NEXT_INSERT_ID is calculated as follows:
# (NOTE: all values are integral, thus a division is whole-numbered as well)
# AI_SEQUENCE = (AI_CURRENT + AI_INCREMENT - AI_OFFSET) / AI_INCREMENT
# NEXT_INSERT_ID = AI_SEQUENCE * AI_INCREMENT + AI_OFFSET
# The binary log (master role)
log_bin            = /var/log/mysql/mysql-bin.log
log_bin_index      = /var/log/mysql/mysql-bin.log.index
# The relay log where binary log data from the master is written to
# by the IO thread before being read by the SQL thread (slave role)
relay_log          = /var/log/mysql/mysql-relay-bin
relay_log_index    = /var/log/mysql/mysql-relay-bin.index
# Purge binary logs older than this many days
expire_logs_days  = 10
# Maximum size of single binary log files
max_binlog_size   = 100M
# Log updates received as a slave from a master to the own
# binary log (used for circular master-master replication)
log_slave_updates = 1
```

Konfiguration – Benutzer

- Anlegen eines Benutzers mit Replikationsrechten
- Host-Rechte sollten ggf. eingeschränkt werden

```
mysql-01> GRANT SUPER, REPLICATION CLIENT, REPLICATION SLAVE ON *.* \  
          TO 'replication'@'192.168.0.%' IDENTIFIED BY 'secret';  
mysql-02> GRANT SUPER, REPLICATION CLIENT, REPLICATION SLAVE ON *.* \  
          TO 'replication'@'192.168.0.%' IDENTIFIED BY 'secret';
```

Konfiguration - Slave

- Festlegen des Master Servers
- SSL von Vorteil

```
mysql-01> CHANGE MASTER TO master_host='masterserver',  
                             master_port=3306,  
                             master_user='replication',  
                             master_password='secret',  
                             master_ssl = 1,  
                             maser_ssl_ca = '/etc/ssl/certs/class3.pem';
```

MySQL Status Master

```
mysql> show master status;
```

File	Position	Binlog_Do_DB	Binlog_Ignore_DB
MysqlMYSQL01-bin.000008	410	adam	

1 row in set (0.00 sec)

Slave

```
mysql> show slave status\G;
```

***** 1. row *****

Slave_IO_State: Waiting for master to send event
Master_Host: mysqlserver
Master_User: replication
Master_Port: 3306
Connect_Retry: 60
Master_Log_File: Mysql1MYSQL02-bin.000008
Read_Master_Log_Pos: 410

[...]

Multi-Master Replication Manager

Was ist MMM?

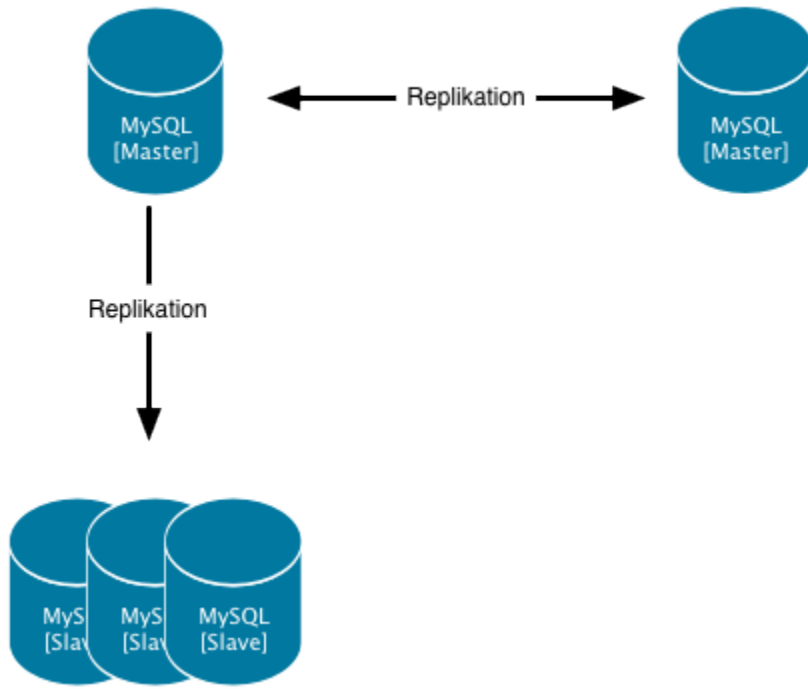
- HA Lösung mit Failover Möglichkeit
- Automatisches Failover für Slave-Server

Was ist MMM nicht?

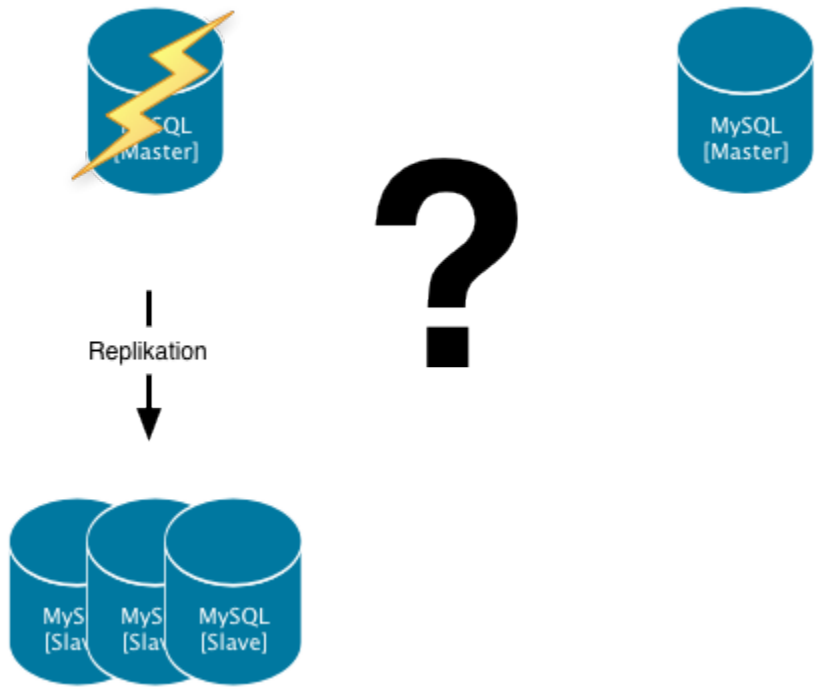
- Load Balancer
- 100% Zuverlässig (Replikation ist nicht Perfekt)

Anleitung für die Installation: <http://mysql-mmm.org/>

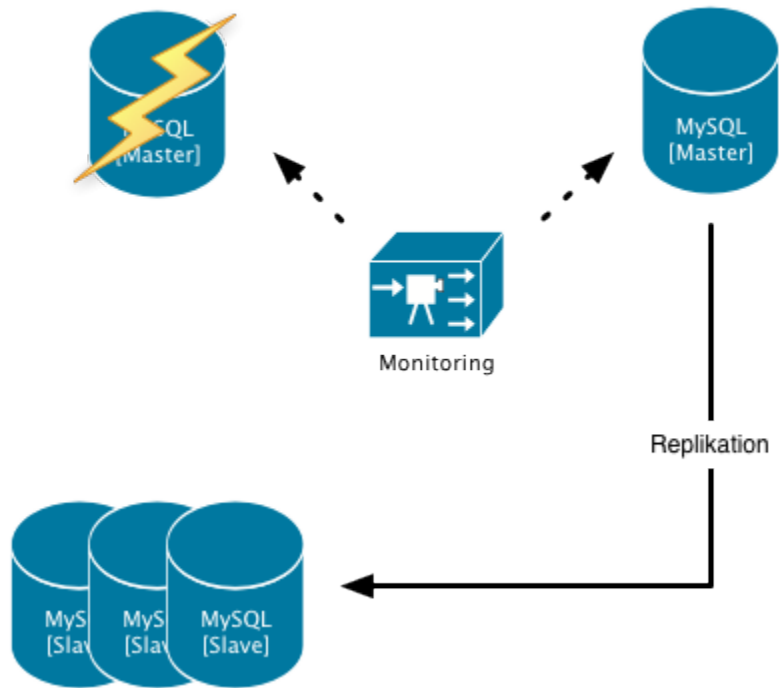
MMM Problem und Lösung



MMM Problem und Lösung



MMM Problem und Lösung

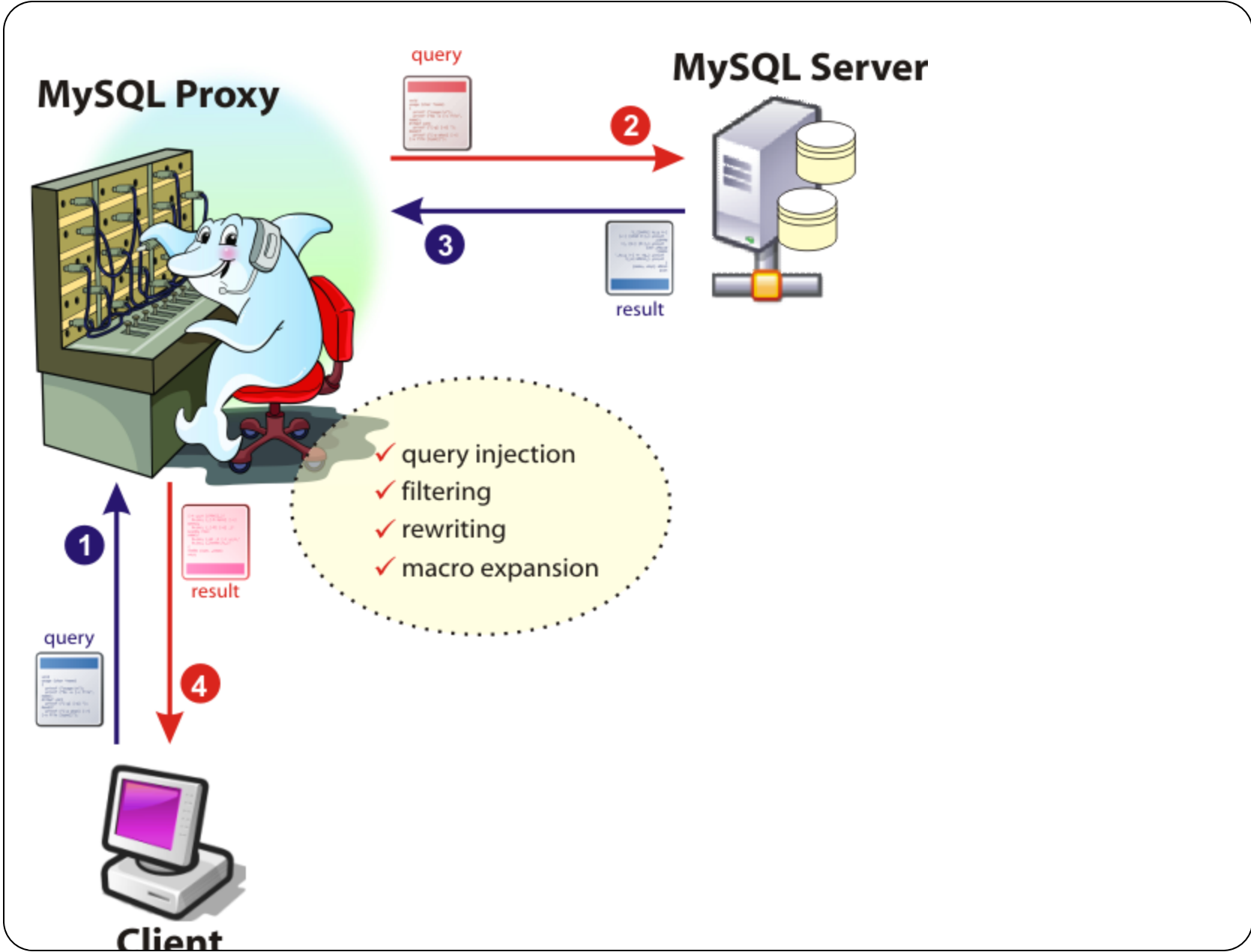


MySQL Proxy

Aufgaben

- Erstellung neuer MySQL Befehle
- Filterung von MySQL Queries
- Query Analyze
- Query Quota Support
- Ausführen von Shell Scripten
- Monitoring
- Load Balancing
- MySQL-Server Remote Steuerung (start/stop)
- Kaffee kochen

Übersicht



Lua

- Scriptsprache
- Entworfen für Embedded Systeme
- MySQL Proxy ist mit Lua-Script erweiterbar



Funktion in Lua

```
function f (x)
  print(x)
end
g = f
g(10)
```

MySQL Proxy – Keywords

- **connect_server()**
 - Wird beim Verbinden zum MySQL Server aufgerufen (Benötigt z.B. für Load Balancer)
- **read_query(packet)**
 - Funktion wird vor dem Senden an den MySQL Server aufgerufen
- **read_query_result(injection_packet)**
 - Diese Funktion wird vor dem Senden des MySQL Servers an den Client aufgerufen

MySQL Proxy – Query Abfangen

inform_user.lua

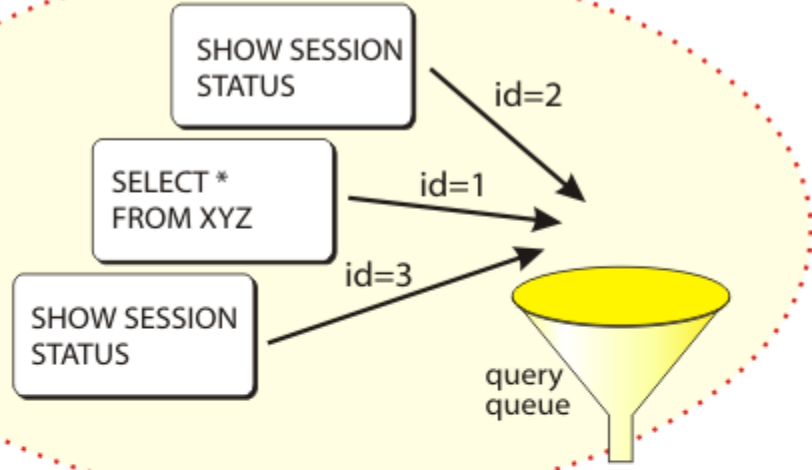
```
-- inform_user.lua
function read_query(packet)
  if string.byte(packet) == proxy.COM_QUERY then
    print("The client committed the following query: " .. string.sub(packet, 2))
  end
end
```

Ausgabe

```
The client committed the following query: show tables
The client committed the following query: select * from t
The client committed the following query: select * from t
The client committed the following query: insert into t ( c ) values ( 'blub' )
```

MySQL Proxy - Query Aufsplitten

MySQL Proxy



read_query()



SELECT * FROM XYZ



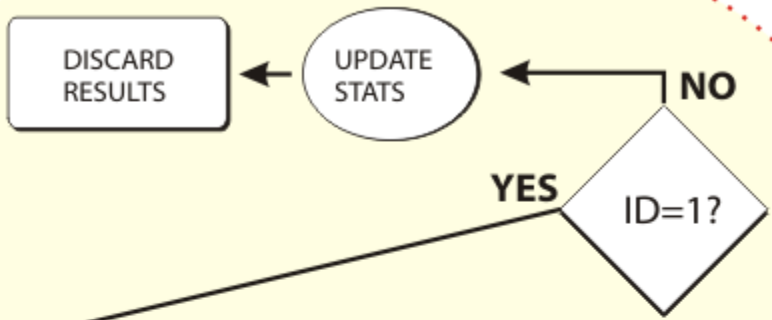
Client



MySQL Server

MySQL Proxy - Query Aufsplitten

MySQL Proxy



read_query_result()

RETURN RESULTS



Client



MySQL Server

Usage

/usr/local/sbin/mysql-proxy

```
--proxy-lua-script  
    =</path/name.lua>  
--proxy-address  
    =<host:port>  
--proxy-read-only-backend-addresses  
    =<host:port>  
--proxy-backend-addresses  
    =<host:port>
```

Das Script wird erst ab dem ersten verbundenen Client ausgeführt

Live Demo

- mysql-01: MySQL Master
- mysql-02: MySQL Master
- mysql-03: MySQL Slave
- web-01: MySQL Proxy und Webserver

Quellen

- <http://mysql-mmm.org/mmm2:guide>
- <http://forge.mysql.com/w/images/0/05/DualMasterSetupsWithMMM.pdf>
- [http://www.admin-magazin.de/\[..\]/Verteilte-Datenbank-mit-MySQL-Proxy](http://www.admin-magazin.de/[..]/Verteilte-Datenbank-mit-MySQL-Proxy)
- [http://datacharmer.org/\[..\]/mysql_proxy_oscon_2008a.pdf](http://datacharmer.org/[..]/mysql_proxy_oscon_2008a.pdf)
- http://www.fromdual.ch/sites/default/files/doag_regio_2011-03.pdf

Fragen? 42!