



MySQL Replikation und GTID

DOAG K & A 2017, Nürnberg

Oli Sennhauser

Senior MySQL Berater, FromDual GmbH

oli.sennhauser@fromdual.com



Über FromDual GmbH

Beratung



Schulung



remote-DBA



Support



Inhalt

MySQL Replikation und GTID

- › **Geschichte der MySQL Replikation**
- › **Statement Based Replikation**
- › **Row Based Replikation**
- › **Probleme**
- › **GTID basierte Replikation**

Geschichte der Replikation

- **Erster öffentlicher MySQL-Release**
 - 31. August 1996
- **MySQL Replikation**
 - Release 3.23.15 (08 May 2000)
 - Replication between master and slaves.
- **MySQL Row Based Replikation (RBR)**
 - Changes in MySQL 5.1.5 (2006-01-10)
 - Replication: Added the `binlog_format` system variable that controls whether to use row-based or statement-based binary logging.
- **MySQL GTID basierte Replikation**
 - Changes in MySQL 5.6.5 (2012-04-10, Milestone 8)
 - Replication with GTIDs
Important Change; Replication: This release introduces global transaction identifiers (GTIDs) for MySQL Replication.

Am Anfang war ...

- Statement Based Replikation (SBR)
 - Logische Replikation

```
mysql> SHOW MASTER STATUS;
+-----+-----+
| File           | Position |
+-----+-----+
| binlog.000001 |      154 |
+-----+-----+

mysql> INSERT INTO test VALUES (NULL, CONCAT('Some data: ', RAND()), SYSDATE());
Query OK, 1 row affected, 1 warning (0.00 sec)
...
mysql> INSERT INTO test VALUES (NULL, CONCAT('Some data: ', RAND()), SYSDATE());
Query OK, 1 row affected, 1 warning (0.00 sec)

mysql> SELECT * FROM test;
+-----+-----+-----+-----+
| id | data                                     | ts                |
+-----+-----+-----+-----+
| 1  | Some data: 0.6192293740988051          | 2017-11-19 22:01:10 |
| 2  | Some data: 0.4439282272420155          | 2017-11-19 22:01:16 | → +6s!!!
+-----+-----+-----+-----+
```

Binary Log

```
shell> mysqlbinlog binlog.000001
...
# at 154
#171119 21:55:33 server id 3391  end_log_pos 531
BEGIN
SET INSERT_ID=1/*!*/;
SET @@RAND_SEED1=743706528, @@RAND_SEED2=996423492/*!*/;
use `test`/*!*/;
SET TIMESTAMP=1511124933/*!*/;
INSERT INTO test VALUES (NULL, CONCAT('Some data: ', RAND()), SYSDATE());
# at 531
#171119 21:55:33 server id 3391  end_log_pos 562
COMMIT/*!*/;
# at 562
#171119 21:55:36 server id 3391  end_log_pos 939
BEGIN;
SET INSERT_ID=2/*!*/;
SET @@RAND_SEED1=6317607, @@RAND_SEED2=1002741132/*!*/;
SET TIMESTAMP=1511124936/*!*/;
INSERT INTO test VALUES (NULL, CONCAT('Some data: ', RAND()), SYSDATE());
# at 939
#171119 21:55:36 server id 3391  end_log_pos 970
COMMIT/*!*/;
...
```

Problem mit SBR

- Zu einfacher Ansatz da:
- **SHOW WARNINGS** und **MySQL Error Log**:

```
[Warning] Unsafe statement written to the binary log using statement format
since BINLOG_FORMAT = STATEMENT. Statement is unsafe because it uses a
system function that may return a different value on the slave. Statement:
INSERT INTO test VALUES (NULL, CONCAT('Some data: ', RAND()), SYSDATE())
```

- **Auf Slave:**

```
mysql> SELECT * FROM test;
+-----+-----+-----+-----+
| id | data | ts |
+-----+-----+-----+-----+
| 1 | Some data: 0.6192293740988051 | 2017-11-19 22:01:10 |
| 2 | Some data: 0.4439282272420155 | 2017-11-19 22:01:30 | → +20s!!!
+-----+-----+-----+-----+
```

Problem mit SBR

- Warum?
 - Sog. **nicht-deterministische** Queries...
 - Warum `RANDOM()` richtig?
- **Achtung: `binlog_format = MIXED` ist ein wenig besser aber nicht 100% perfekt!**
- **Der einzig sichere Weg ist...**

Row Based Replication

- Row Based Replication (RBR)
 - Physische Replikation

```

shell> mysqlbinlog binlog.000002
# at 154
#171119 22:41:12 server id 3391  end_log_pos 219
/*!50718 SET TRANSACTION ISOLATION LEVEL READ COMMITTED*//*!*/;
# at 219
#171119 22:41:12 server id 3391  end_log_pos 351
SET TIMESTAMP=1511127672*//*!*/;
BEGIN;
# at 351
#171119 22:41:12 server id 3391  end_log_pos 425
BINLOG '
ePoRWhM/DQAANAAAF8BAAAAA0EAAAAAAAAEABHRlc3QABHRlc3QAAwMPEQNAAAACfawuQg==
ePoRWh4/DQAASgAAAKkBAAAAA0EAAAAAAAAEAAgAD//gBAAAAHVNvbWUgZGF0YToGMC44MjU0MTU3
Mzg2OTU2ODgzWhH6eFML7hw='*//*!*/;
# at 425
#171119 22:41:12 server id 3391  end_log_pos 456
COMMIT*//*!*/;
    
```

Konfiguration der RBR

- Crash-safe (ja, war es früher nicht!)
- Master:

```

server_id          = 1
log_bin           = binary-log
expire_logs_days  = <n>           # Default: 0 → NIE löschen
binlog_format     = row           # Default in 5.7
sync_binlog       = 1            # Default in 5.7
binlog_checksum   = crc32        # Default in 5.7
innodb_support_xa = 1            # Default in 5.7

```

- Slave:

```

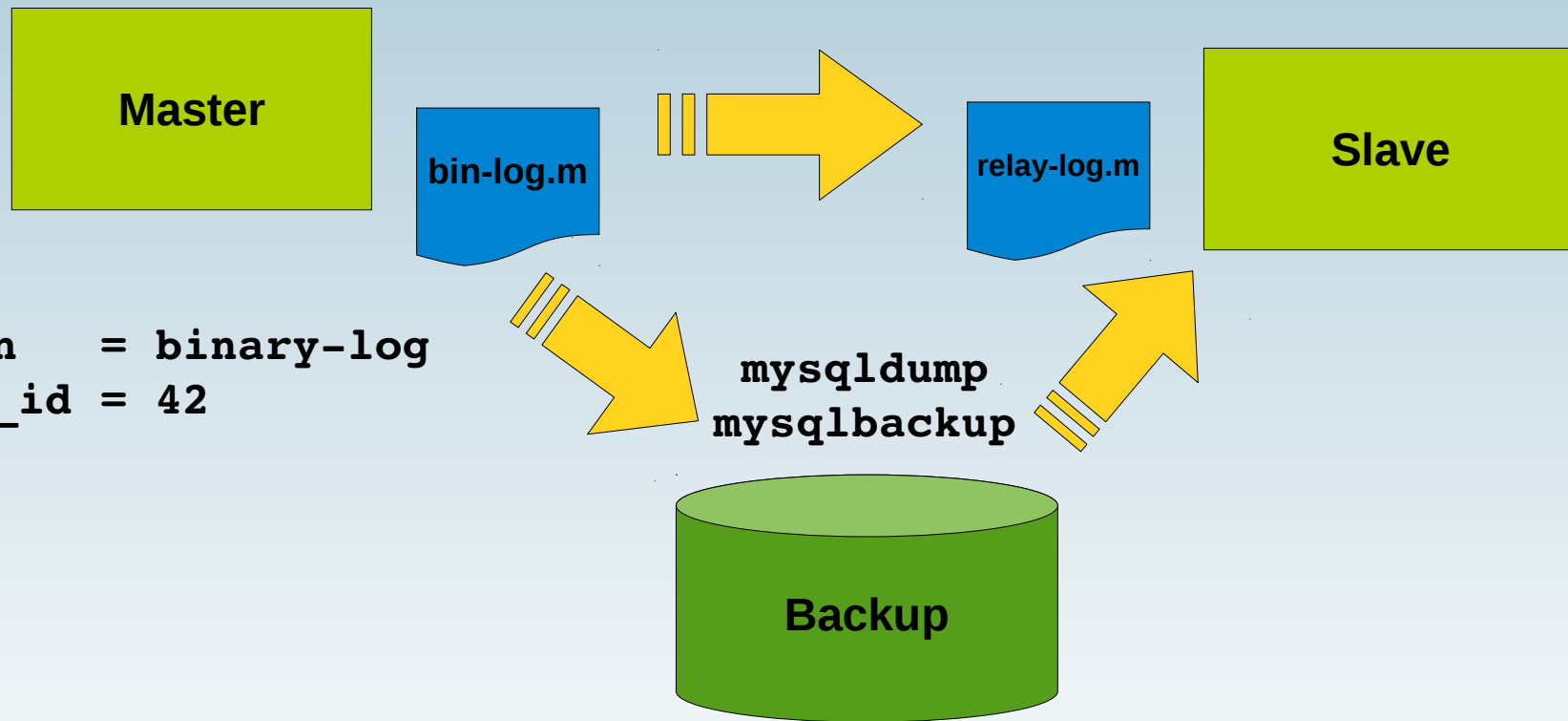
server_id          = 2           # Muss unterschiedlich sein!!!
master_info_repository = TABLE  # Default in 5.7: FILE
relay_log_info_repository = TABLE # Default in 5.7: FILE
relay_log_recovery  = ON        # Default in 5.7: OFF
relay_log_purge     = ON        # Default in 5.7
slave_sql_verify_checksum = ON   # Default in 5.7

```

Aufsetzen der Replikation

```
CREATE USER 'replication'@'192.168.1.%'
  IDENTIFIED BY 'secret';
GRANT REPLICATION SLAVE ON *.*
  TO 'replication'@'192.168.1.%';
```

```
CHANGE MASTER TO ...
  master_log_file='...'
  , master_log_pos=...;
START SLAVE;
```

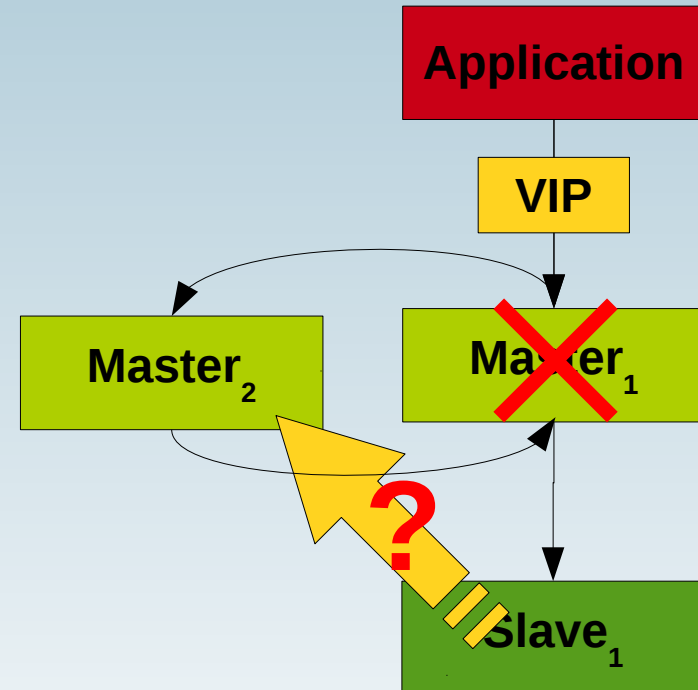
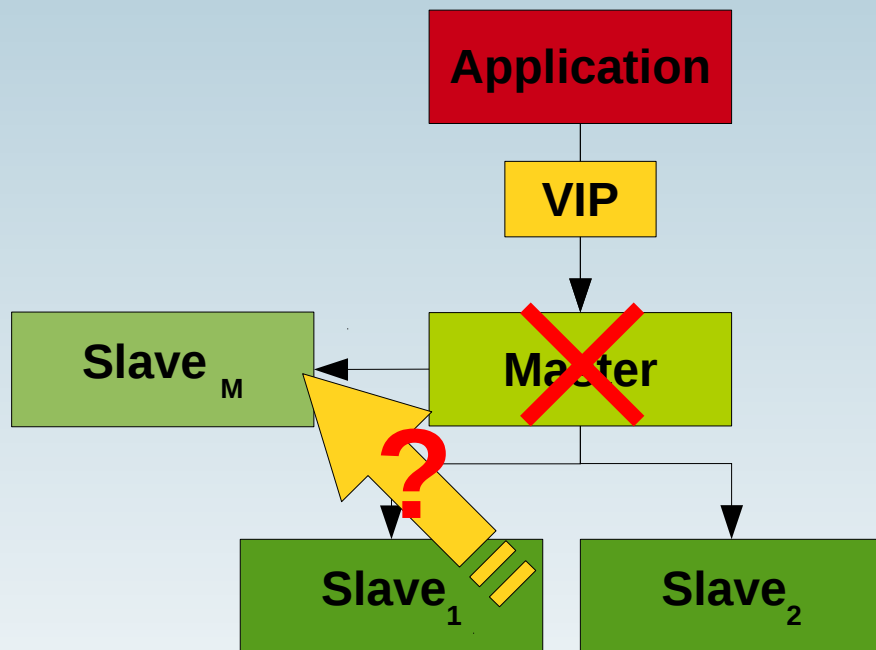


```
log_bin    = binary-log
server_id  = 42
```

Problem mit SBR & RBR

- Master/Slave/Slave

- Master/Master/Slave



Global Transaction ID

- **Server UUID (`server_uuid`)**
 - Kommt von `$datadir/auto.cnf`
 - Wird beim erstellen der Instanz generiert:

```
[Warning] No existing UUID has been found, so we assume that this is the first
time that this server has been started. Generating a new UUID:
57f8c964-cdec-11e7-86f0-28d2445cb2e9.
```

```
cat /var/lib/mysql/auto.cnf
[auto]
server-uuid=57f8c964-cdec-11e7-86f0-28d2445cb2e9
```

- **Transaktions-Sequenznummer ($1 - 2^{64}$)**
- **GTID:**
 - **57f8c964-cdec-11e7-86f0-28d2445cb2e9:32554**

Konfiguration der GTID

- Konfiguration auf Master und Slave:

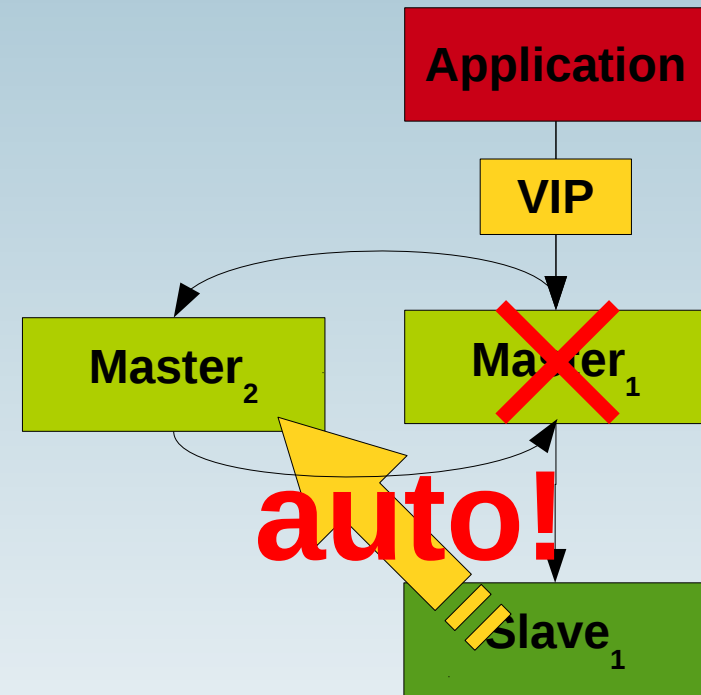
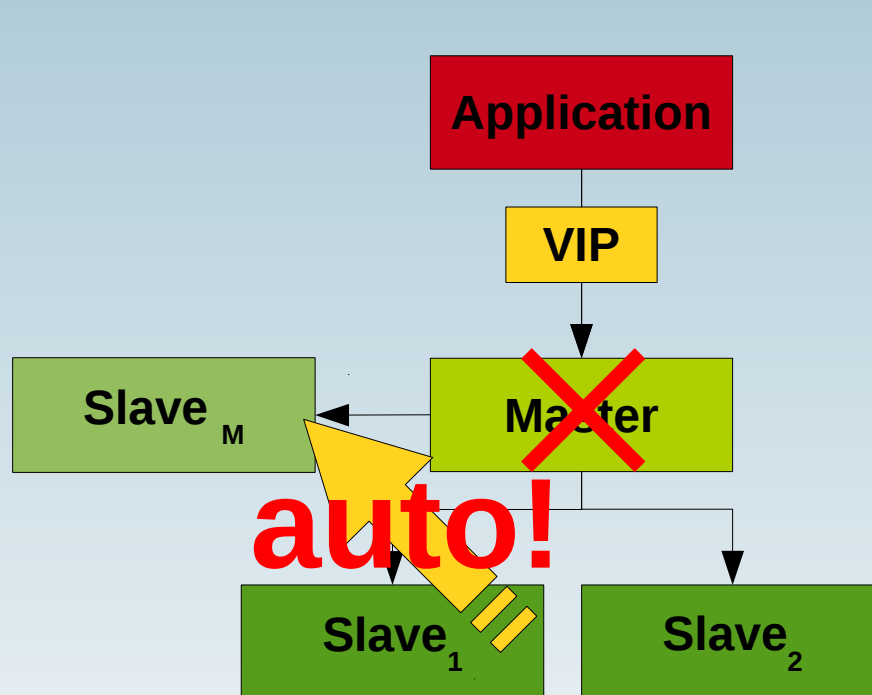
```
gtid_mode = ON  
enforce_gtid_consistency = 1
```

- User kreieren
- Backup/Restore
- Aufsetzen der Replikation

```
CHANGE MASTER TO ...  
  master_auto_position = 1;  
START SLAVE;
```

Charme von GTID

- Master/Slave/Slave
- Master/Master/Slave



CHANGE MASTER TO master_host='...';

Differenzen zw. Master/Slave

- **Beispiel: Historie auf Master löschen NICHT aber auf Reporting Slave**

```
SET SESSION sql_log_bin = 0;
```

```
DELETE FROM history  
WHERE created < DATESUB(CURRENT_DATE(), INTERVAL 90 DAY);
```

```
SET SESSION sql_log_bin = 1;
```

- **Gleich wie früher: Nicht ins Binlog...**
- **gtid_executed bleibt gleich!**

Replikationsprobleme

- **Beispiel: Ich füge Row auf Master ein, vergesse aber, dass die auf Slave schon da ist:**

```
Last_SQL_Errno: 1062
Last_SQL_Error: Could not execute Write_rows event on
                table test.test; Duplicate entry '3' for
                key 'PRIMARY', Error_code: 1062; handler
                error HA_ERR_FOUND_DUPP_KEY; the event's
                master log binlog.000072, end_log_pos 711

Master_Server_Id: 3391
Master_UUID: 57f8c964-cdec-11e7-86f0-28d2445cb2e9
SQL_Error_Timestamp: 171120 15:28:40
Retrieved_Gtid_Set: 57f8c964-cdec-11e7-86f0-28d2445cb2e9:5-99
Executed_Gtid_Set: 57f8c964-cdec-11e7-86f0-28d2445cb2e9:1-6
```

Problem auf Slave überspringen

- Alt:

```
SET GLOBAL sql_slave_skip_counter = 1;
```

geht mit GTID NICHT mehr!

- Neu: Inject empty transactions:

```
STOP SLAVE;
```

```
SET GTID_NEXT='57f8c964-cdec-11e7-86f0-28d2445cb2e9:7';  
START TRANSACTION; COMMIT;  
SET GTID_NEXT='AUTOMATIC';
```

```
START SLAVE;
```

MySQL Variablen und GTID

- Variable: `gtid_executed`
 - Auf diesem Knoten ausgeführte Transaktionen
- Variable: `gtid_purged`
 - Alles was NICHT mehr in Binary Logs drin ist

```
mysql> SHOW GLOBAL VARIABLES LIKE 'gtid_%ed';
```

Variable_name	Value
<code>gtid_executed</code>	<code>0e18c671-d40c-11e5-9524-56847afe9799:1-7</code>
<code>gtid_purged</code>	<code>0e18c671-d40c-11e5-9524-56847afe9799:1-4</code>

- Das war der gute Fall...

Weniger schön ist...

- Weniger schön ist dieser Fall:

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

```
344161d7-01ab-11e6-9942-56847afe9799:1-32554,  
0e18c671-d40c-11e5-9524-56847afe9799:1-4,           ← das ist Meins!  
a2c9b296-7f05-11e6-b5f3-28d2445cb2e9:1-13312
```

```
mysql> SELECT @@server_uuid;  
0e18c671-d40c-11e5-9524-56847afe9799
```

- Woher kommt das?
 - Grundaussage: Transaktionen von 3 unterschiedlichen Servern!
 - Möglichkeit 1: MMS
 - Möglichkeit 2: Master wurde neu kreiert (auto.cnf) aber/und nicht resettet!!!
Typischerweise bei Backup/Restore

Betriebsbefehle und GTID

- **PURGE BINARY LOGS TO ...;**
 - `gtid_purged` wird angepasst...
- **RESET MASTER;**
 - Ganz gefährlich!
Binary Log Historie geht verloren...
--> `gtid_executed = ''`
 - Nur so ist
`SET GLOBAL gtid_purged='...'`
möglich!
 - Hiermit ist die Historie manipulierbar!
 - Wird beim Erstellen des Slaves verwendet.

Problem: Neue Instanz

- Bei Restore meist neue Instanz und somit neue UUID

```
mysqld --initialize(-insecure)
```

- Aber dann ist auch `auto.cnf` weg:

```
[ERROR] --initialize specified but the data
directory has files in it. Aborting.
```

- Somit neue GTIDs
- Also auch händischer Restore von `auto.cnf!!!`
 - IMHO gehört UUID IN die DB und nicht aufs Filesystem!!!
- Auch MySQL Enterprise Backup 4.1.0 :-(
 - Ist gemeldet bei MySQL aber noch nicht gefixed...
 - Bug #86280, 2017-05-11, Status: Verified

Rollback

- **CHANGE MASTER TO**
`master_auto_position = 0`
`, master_log_file='...'`
`, master_log_pos=...;`
- **Wo bringt es was?**
 - Bei ca. 5% der Installationen
 - M + n x S oder M/M + S
- **Sauberer/Genauerer Arbeiten erforderlich!**
- **Aber es ist die Zukunft...**

Q & A



www.fromdual.com



Fragen ?

Diskussion?

Wir haben Zeit für ein persönliches Gespräch...

- **FromDual bietet neutral und unabhängig:**
 - **Beratung**
 - **Remote-DBA**
 - **Support für MySQL, Galera, Percona Server und MariaDB**
 - **Schulung**

www.fromdual.com/presentations