Why MySQL Replication Fails, and How to Get it Back

September, 26, 2017
Sveta Smirnova

OPEN SOURCE DATABASE CONFERENCE

PERCONA
LIVE EUROPE
DUBLIN

Sveta Smirnova

- MySQL Support engineer
- Author of
  - MySQL Troubleshooting
  - JSON UDF functions
  - FILTER clause for MySQL
- Speaker
  - Percona Live, OOW, Fosdem, DevConf, HighLoad...

Thank You Sponsors!

continuent

VividCortex

facebook

Altinity

TIMESCALE

ObjectRocket
a rackspace company

Ispirer

PGDAY'17 RUSSIA

severalnines

WEBYOG

MySQL

Google Cloud Platform

GitHub

CMSWiRE

datanami
BIG DATA • BIG ANALYTICS • BIG INSIGHTS

ODBMS.org

open source .com

inside BIGDATA

insideHPC

O'REILLY

PERCONA LIVE EUROPE DUBLIN

Replication in MySQL

- Exists since very first versions

Replication in MySQL

- Exists since very first versions
- Easy to use

Replication in MySQL

- Exists since very first versions
- Easy to use
- Minimal setup

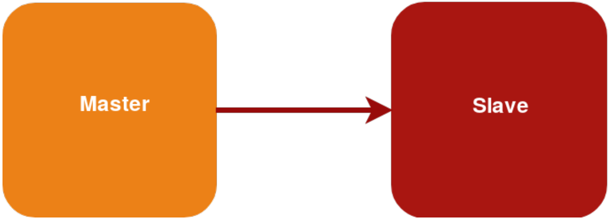
Turn Replication On

- Master
  - `--log-bin`
  - `--server-id`
  - `GRANT REPLICATION SLAVE ON *.* ...`

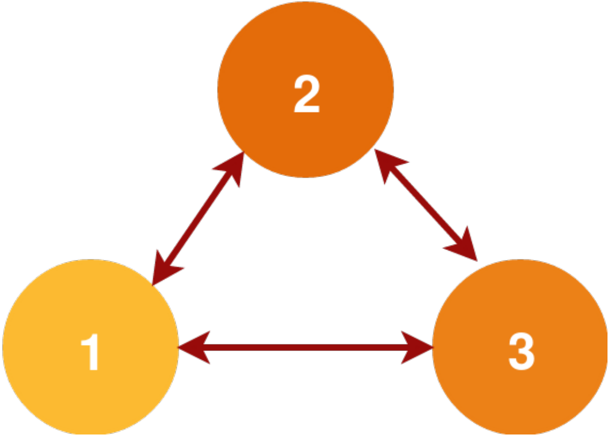
Turn Replication On

- Master
  - `--log-bin`
  - `--server-id`
  - `GRANT REPLICATION SLAVE ON *.* ...`
- Slave
  - `--server-id`
  - `CHANGE MASTER ...`
  - `START SLAVE`

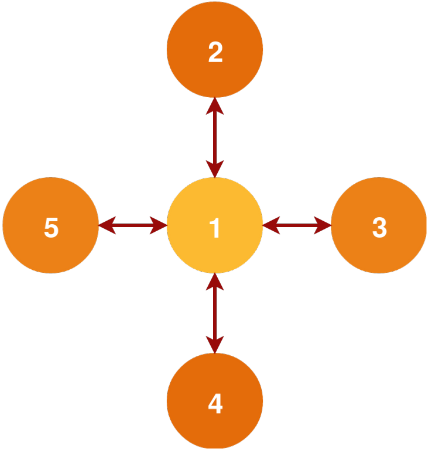
Simple



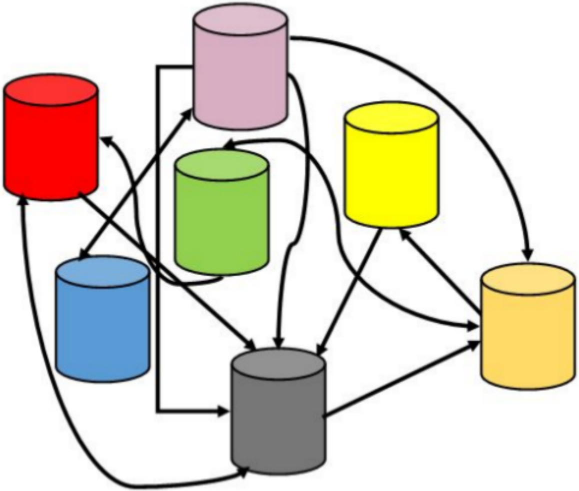
Circle



Star

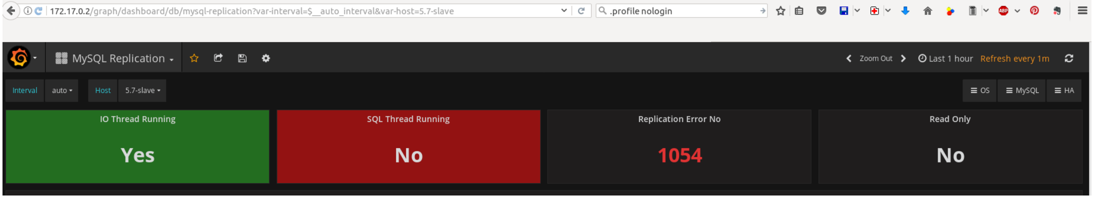


Creative



Typical Replication Errors

Replication Stopped



172.17.0.2/graph/dashboard/db/mysql-replication?var-interval=$__auto_interval&var-host=5.7-slave

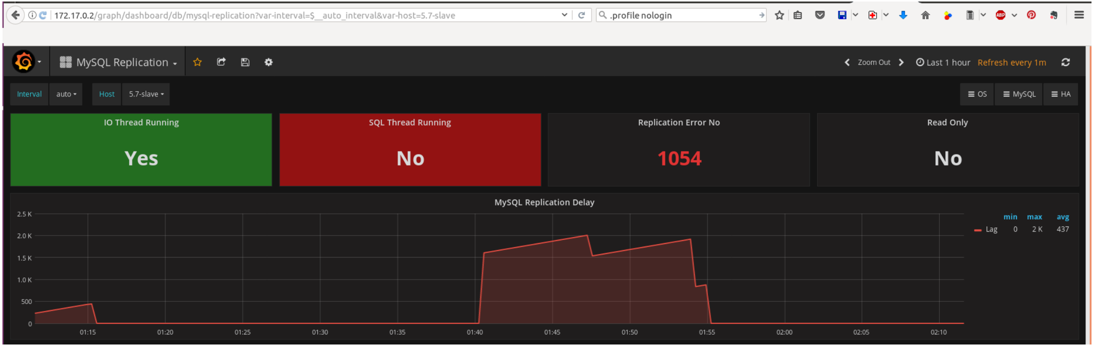
IO Thread Running: Yes

SQL Thread Running: No

Replication Error No: 1054

Read Only: No

Slave Lags from the Master



172.17.0.2/graph/dashboard/db/mysql-replication?var-interval=$__auto_interval&var-host=5.7-slave

.profile nologin

MySQL Replication

Zoom Out

Last 1 hour Refresh every 1m

Interval auto

Host 5.7-slave

OS MySQL HA

IO Thread Running: Yes

SQL Thread Running: No

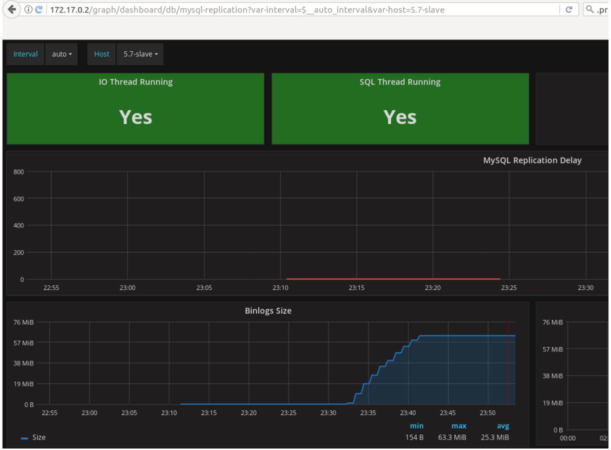
Replication Error No: 1054

Read Only: No

MySQL Replication Delay

	min	max	avg
Lag	0	2 K	437

Increased Resource Usage on Master



172.17.0.2/graph/dashboard/db/mysql-replication?var-interval=$__auto_interval&var-host=5.7-slave

Interval auto ▾ Host 5.7-slave ▾

IO Thread Running

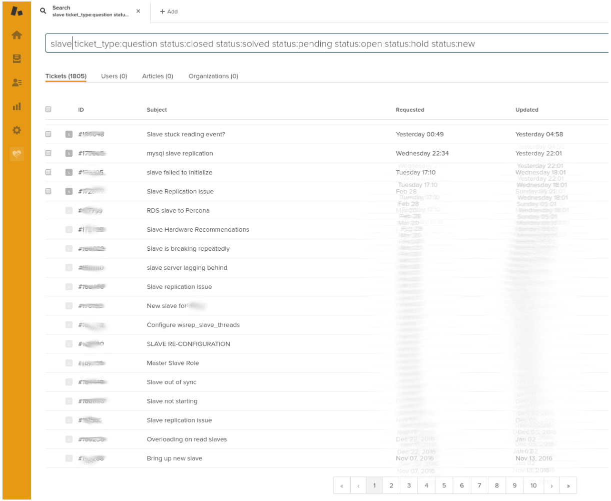
Yes

SQL Thread Running

Yes

	min	max	avg
Size	154 B	63.3 MiB	25.3 MiB

Not a Full List!



slave ticket_type:question status:closed status:solved status:pending status:open status:hold status:new

ID	Subject	Requested	Updated
[redacted]	Slave stuck reading event?	Yesterday 00:49	Yesterday 04:58
[redacted]	mysql slave replication	Wednesday 22:34	Yesterday 22:01
[redacted]	slave failed to initialize	Tuesday 17:10	Wednesday 18:01
[redacted]	Slave Replication Issue	Feb 28	Sunday 05:01
[redacted]	RDS slave to Percona	[illegible]	[illegible]
[redacted]	Slave Hardware Recommendations	[illegible]	[illegible]
[redacted]	Slave is breaking repeatedly	[illegible]	[illegible]
[redacted]	slave server lagging behind	[illegible]	[illegible]
[redacted]	Slave replication issue	[illegible]	[illegible]
[redacted]	New slave for [redacted]	[illegible]	[illegible]
[redacted]	Configure wsrep_slave_threads	[illegible]	[illegible]
[redacted]	SLAVE RE-CONFIGURATION	[illegible]	[illegible]
[redacted]	Master Slave Role	[illegible]	[illegible]
[redacted]	Slave out of sync	[illegible]	[illegible]
[redacted]	Slave not starting	[illegible]	[illegible]
[redacted]	Slave replication issue	Nov 17, 2016	Dec 05, 2016
[redacted]	Overloading on read slaves	Dec 22, 2016	Jan 02
[redacted]	Bring up new slave	Nov 07, 2016	Nov 13, 2016

MySQL Replication: Must Know

PERCONA LIVE EUROPE DUBLIN

Asynchronous

Master

Slave

← Initiates

Asynchronous

Master

Slave

← Initiates

← Requests a packet

Asynchronous

Master

Slave

← Initiates

← Requests a packet

Sends the packet →

Asynchronous

Master

Slave

← Initiates

← Requests a packet

Sends the packet →

... ?

Did Slave Recieve Data?

- Network error
- Authorization

Tool #1: SHOW SLAVE STATUS

- IO thread Configuration
- SQL thread Configuration
- IO thread State
- SQL thread State
- Errors
  - Only last one
  - All are in the error log

```
               Slave_IO_State: Waiting for master to send event
                  Master_Host: 127.0.0.1
                  Master_User: root
                  Master_Port: 13000
                Connect_Retry: 60
              Master_Log_File: master-bin.000002
          Read_Master_Log_Pos: 63810611
               Relay_Log_File: slave-relay-bin-master@002d1.000004
                Relay_Log_Pos: 1156
        Relay_Master_Log_File: master-bin.000001
             Slave_IO_Running: Yes
            Slave_SQL_Running: No
              Replicate_Do_DB:
          Replicate_Ignore_DB:
           Replicate_Do_Table:
       Replicate_Ignore_Table:
      Replicate_Wild_Do_Table:
  Replicate_Wild_Ignore_Table:
                   Last_Errno: 1032
                   Last_Error: Could not execute Update_rows event on...
                 Skip_Counter: 0
          Exec_Master_Log_Pos: 989
              Relay_Log_Space: 63814652
              Until_Condition: None
               Until_Log_File:
                Until_Log_Pos: 0
           Master_SSL_Allowed: No | Master_SSL_CA_File:
```

```
           Master_SSL_CA_Path:
              Master_SSL_Cert:
            Master_SSL_Cipher:
               Master_SSL_Key:
        Seconds_Behind_Master: NULL
Master_SSL_Verify_Server_Cert: No
                Last_IO_Errno: 0
                Last_IO_Error:
               Last_SQL_Errno: 1032
               Last_SQL_Error: Could not execute Update_rows event on...
  Replicate_Ignore_Server_Ids:
             Master_Server_Id: 1
                  Master_UUID: d08c509e-6857-11e6-8872-30b5c2208a0f
             Master_Info_File: mysql.slave_master_info
                    SQL_Delay: 0
          SQL_Remaining_Delay: NULL
      Slave_SQL_Running_State:
           Master_Retry_Count: 10
                  Master_Bind:
      Last_IO_Error_Timestamp:
     Last_SQL_Error_Timestamp: 160823 15:11:21
               Master_SSL_Crl:
           Master_SSL_Crlpath:
           Retrieved_Gtid_Set:
            Executed_Gtid_Set:
                Auto_Position: 0
         Replicate_Rewrite_DB:
                 Channel_Name: master-1 |  Master_TLS_Version:
```

Network Errors

```
        Slave_IO_Running: Connecting
       Slave_SQL_Running: Yes
       ...
           Last_IO_Errno: 1045
           Last_IO_Error: error connecting to master 'root@127.0.0.1:13000' -
          Last_SQL_Errno: 0
          Last_SQL_Error:
          ...
 Slave_SQL_Running_State: Slave has read all relay log; waiting for more updates
      Master_Retry_Count: 86400
             Master_Bind:
 Last_IO_Error_Timestamp: 160824 03:18:36
Last_SQL_Error_Timestamp:
```

#2: connection_status in Performance Schema

```
mysql> select * from performance_schema.replication_connection_status\G
*************************** 1. row ***************************
             CHANNEL_NAME:
               GROUP_NAME:
              SOURCE_UUID:
                THREAD_ID: NULL
            SERVICE_STATE: CONNECTING
COUNT_RECEIVED_HEARTBEATS: 0
 LAST_HEARTBEAT_TIMESTAMP: 0000-00-00 00:00:00
 RECEIVED_TRANSACTION_SET:
        LAST_ERROR_NUMBER: 1045
       LAST_ERROR_MESSAGE: error connecting to master ’root@127.0.0.1:13000’ -
                           retry-time: 60  retries: 4
     LAST_ERROR_TIMESTAMP: 2016-08-24 03:21:36
1 row in set (0,01 sec)
```

#3: Error Log File

```
2016-08-24T00:18:36.077384Z 3 [ERROR] Slave I/O for channel ’’: error connecting to
master ’root@127.0.0.1:13000’ - retry-time: 60  retries: 1, Error_code: 1045
2016-08-24T00:19:36.299011Z 3 [ERROR] Slave I/O for channel ’’: error connecting to
master ’root@127.0.0.1:13000’ - retry-time: 60  retries: 2, Error_code: 1045
2016-08-24T00:20:36.485315Z 3 [ERROR] Slave I/O for channel ’’: error connecting to
master ’root@127.0.0.1:13000’ - retry-time: 60  retries: 3, Error_code: 1045
2016-08-24T00:21:36.677915Z 3 [ERROR] Slave I/O for channel ’’: error connecting to
master ’root@127.0.0.1:13000’ - retry-time: 60  retries: 4, Error_code: 1045
2016-08-24T00:22:36.872066Z 3 [ERROR] Slave I/O for channel ’’: error connecting to
master ’root@127.0.0.1:13000’ - retry-time: 60  retries: 5, Error_code: 1045
```

#4: perror

```
$ perror 1045
MySQL error code 1045 (ER_ACCESS_DENIED_ERROR): Access denied for user ’%-.48s’@’%-.64s’
(using password: %s)
```

#5: MySQL Command Line Client

- On the slave

```
$ mysql -h127.0.0.1 -P13000 -uslave_user -pslave_password
Warning: Using a password on the command line interface can be insecure.
ERROR 1045 (28000): Access denied for user ’slave_user’@’localhost’ (using password: YES
```

#5: MySQL Command Line Client

- On the slave
- On the master

```
mysql> SHOW GRANTS;
+-----------------------------------------+
| Grants for slave_user@%                 |
+-----------------------------------------+
| GRANT SELECT ON *.* TO ’slave_user’@’%’ |
+-----------------------------------------+
1 row in set (0.00 sec)
```

#5: MySQL Command Line Client

- On the slave
- On the master
- Fix privileges on master

```
GRANT REPLICATION SLAVE
ON *.* TO 'slave_user'@'%'
```

#5: MySQL Command Line Client

- On the slave
- On the master
- Fix privileges on master
- Restart replication

Semisynchronous plugin

Master

Slave

← Initiates

Semisynchrous plugin

Master

Slave

← Initiates

← Requests a packet

Semisynchrous plugin

Master

Slave

← Initiates

← Requests a packet

Sends the packet →

Semisynchrous plugin

Master

Slave

← Initiates

← Requests a packet

Sends the packet →

Waits "Ack"

Semisynchrous plugin

Master	Slave
	← Initiates
	← Requests a packet
Sends the packet →	
Waits "Ack"	
	← Sends "Ack"

Semisynchrous: Troubleshooting

- Writes on master are slower

Semisynchrous: Troubleshooting

- Writes on master are slower
- How many ”Ack”-s master waits?

Semisynchrous: Troubleshooting

- Writes on master are slower
- How many ”Ack”-s master waits?
  - Before 5.7: from single slave

Semisynchrous: Troubleshooting

- Writes on master are slower
- How many ”Ack”-s master waits?
  - Before 5.7: from single slave
  - Now in MySQL: `rpl_semi_sync_master_wait_for_slave_count`

Semisynchrous: Troubleshooting

- Writes on master are slower
- How many ”Ack”-s master waits?
  - Before 5.7: from single slave
  - Now in MySQL:
    `rpl_semi_sync_master_wait_for_slave_count`
  - Won’t wait others

Semisynchrous: Troubleshooting

- Writes on master are slower
- How many ”Ack”-s master waits?
- What happens in case of timeout?

Semisynchrous: Troubleshooting

- Writes on master are slower
- How many ”Ack”-s master waits?
- What happens in case of timeout?
  - Replication becomes asynchronous

Semisynchrous: Troubleshooting

- Writes on master are slower
- How many ”Ack”-s master waits?
- What happens in case of timeout?
- What does ”Ack” mean?

Semisynchrous: Troubleshooting

- Writes on master are slower
- How many ”Ack”-s master waits?
- What happens in case of timeout?
- What does ”Ack” mean?
  - Event written into relay log

Semisynchrous: Troubleshooting

- Writes on master are slower
- How many ”Ack”-s master waits?
- What happens in case of timeout?
- What does ”Ack” mean?
  - Event written into relay log
  - It is unknown if event applied

Two Kinds of Slave Threads

IO thread

Reads from the master

SQL thread

Two Kinds of Slave Threads

IO thread

Reads from the master

Stores in the relay log

SQL thread

Two Kinds of Slave Threads

IO thread

Reads from the master

Stores in the relay log

SQL thread

← Reads from relay log

Two Kinds of Slave Threads

IO thread

Reads from the master

Stores in the relay log

SQL thread

← Reads from relay log

Executes

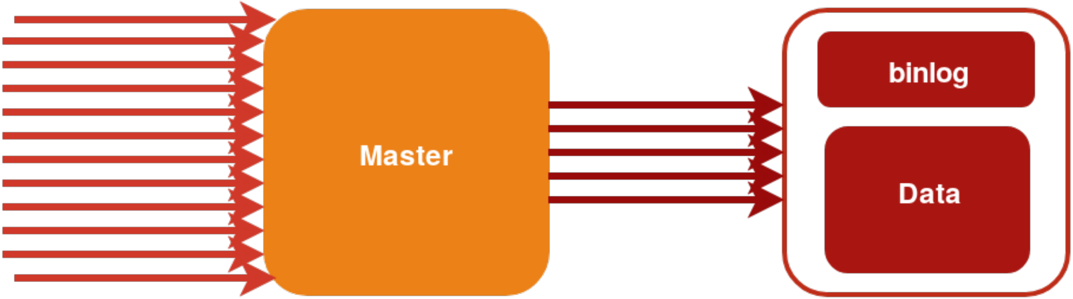
Single SQL Thread

- Easier for troubleshooting

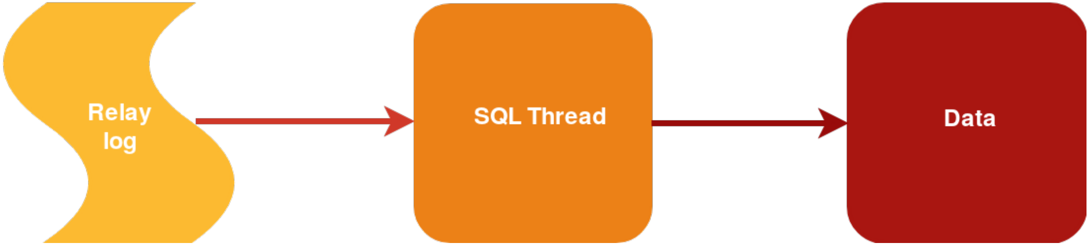
Single SQL Thread

- Easier for troubleshooting
- Slower than master
    - High parallel load

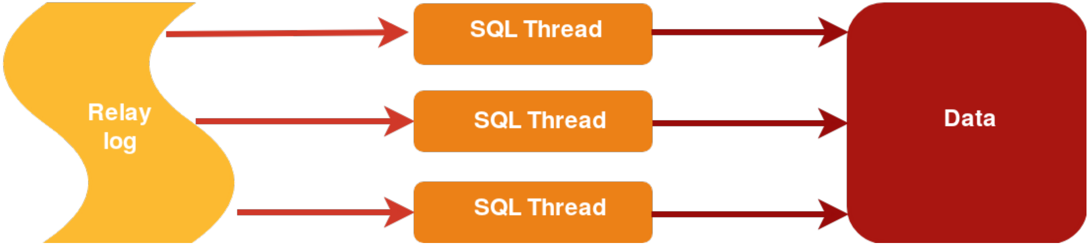
Writes on the Master



Writes on the Slave: Single SQL Thread



Multiple SQL Threads: 5.6+



Performance Tuning

- MySQL: `--slave_parallel_workers`
- MySQL: `--slave_parallel_type=DATABASE | LOGICAL_CLOCK`
- MySQL 8.0.1+:

```
SET @@GLOBAL.binlog_transaction_dependency_tracking =
WRITESET | WRITESET_SESSION | COMMIT_ORDER;
```

Performance Tuning

- MariaDB: `--slave_parallel_threads`
- MariaDB: `--slave_parallel_max_queued`
- MariaDB: `--slave_domain_parallel_threads`
- MariaDB: `--slave_parallel_mode=optimistic | conservative | aggressive | minimal | none`

#6: Error of One Thread Stops All

```
mysql> select WORKER_ID, SERVICE_STATE, LAST_SEEN_TRANSACTION, LAST_ERROR_NUMBER,
    -> LAST_ERROR_MESSAGE from performance_schema.replication_applier_status_by_worker\G
*************************** 1. row ***************************
            WORKER_ID: 1
        SERVICE_STATE: OFF
LAST_SEEN_TRANSACTION: d318bc17-66dc-11e6-a471-30b5c2208a0f:4988
    LAST_ERROR_NUMBER: 0
   LAST_ERROR_MESSAGE:
*************************** 2. row ***************************
            WORKER_ID: 3
        SERVICE_STATE: OFF
LAST_SEEN_TRANSACTION: d318bc17-66dc-11e6-a471-30b5c2208a0f:4986
    LAST_ERROR_NUMBER: 1032
   LAST_ERROR_MESSAGE: Worker 2 failed executing transaction...
```

#6: Error of One Thread Stops All

```
MariaDB [test]> select id, command, time, state from information_schema.processlist
             -> where user=’system user’;
+----+---------+------+------------------------------------------------------------------+
| id | command | time | state                                                            |
+----+---------+------+------------------------------------------------------------------+
| 25 | Connect | 4738 | Waiting for master to send event                                 |
| 24 | Connect | 5096 | Slave has read all relay log; waiting for the slave I/O thread t |
| 23 | Connect |    0 | Waiting for work from SQL thread                                 |
| 22 | Connect |    0 | Unlocking tables                                                 |
| 21 | Connect |    0 | Update_rows_log_event::ha_update_row(-1)                         |
| 20 | Connect |    0 | Waiting for prior transaction to start commit before starting ne |
| 19 | Connect |    0 | Update_rows_log_event::ha_update_row(-1)                         |
| 18 | Connect |    0 | Update_rows_log_event::ha_update_row(-1)                         |
| 17 | Connect |    0 | Update_rows_log_event::find_row(-1)
...
```

Which Kind of Errors?

- Different data
  - Slave cannot apply event from relay log

Which Kind of Errors?

- Different data
  - Slave cannot apply event from relay log
- Different errors on master and slave
  - Triggers
  - Transactional and non-transactional tables in the same transaction

Different Data on Master and Slave

- Did table change outside of the replication?
  - How?
  - Can it cause conflict with changes on the master?

Different Data on Master and Slave

- Did table change outside of the replication?
- Are table structures identical?
  - Percona Toolkit
    - `pt-table-checksum, pt-table-sync`
  - MySQL Utilities
    - `mysqlrplsync, mysqldbcompare, mysqldiff`

Different Data on Master and Slave

- Did table change outside of the replication?
- Are table structures identical?
- Are changes in the correct order?
  - `mysqlbinlog`
  - Application logic on the master

Logical

Master

Recieves a change

Storage Engine

Logical

Master

Recieves a change

Sends to SE →

Storage Engine

Logical

Master

Recieves a change

Sends to SE →

Storage Engine

Writes into table

Logical

Master

Recieves a change

Sends to SE →

Storage Engine

Writes into table

← Returns control

Logical

Master

Recieves a change

Sends to SE →

Storage Engine

Writes into table

← Returns control

Writes into binary log

Logical

Master	Storage Engine
Recieves a change	
Sends to SE →	
	Writes into table
	← Returns control
Writes into binary log	
Synchronizes →	← Synchronizes

Master Performance

- More writes
  - RBR: `--binlog_row_image`

Master Performance

- More writes
  - RBR: `--binlog_row_image`
  - `--binlog_cache_size`
    - Watch `Binlog_cache_disk_use`

Master Performance

- More writes
  - RBR: `--binlog_row_image`
  - `--binlog_cache_size`
    - Watch `Binlog_cache_disk_use`
  - `--binlog_stmt_cache_size`
    - Watch `Binlog_stmt_cache_disk_use`

Master Performance

- More writes
- Synchronization
  - `--binlog_sync`
  - Do not disable!
  - You may set it greater than 1

Master Behavior

- Binary log lifetime
  - `--expire_log_days`

Master Behavior

- Binary log lifetime
- Synchronization
  - SBR is not safe with `READ COMMITTED` and `READ UNCOMMITTED`

Master Behavior

- Binary log lifetime
- Synchronization
- Order of records in the binary log
  - Non-deterministic events and SBR

Statement-Based Binary Log Format

Client

Binary log

Statement-Based Binary Log Format

Client

INSERT INTO ... →

Binary log

Statement-Based Binary Log Format

Client

INSERT INTO ... →

Binary log

SET TIMESTAMP...

Statement-Based Binary Log Format

Client

INSERT INTO ... →

Binary log

SET TIMESTAMP...

SET sql_mode...

Statement-Based Binary Log Format

Client

INSERT INTO ... →

Binary log

SET TIMESTAMP...

SET sql_mode...

INSERT INTO ...

SBR: Strong Sides

- Exists since very first versions

SBR: Strong Sides

- Exists since very first versions
- Table definitions on master and slave can significantly vary

SBR: Strong Sides

- Exists since very first versions
- Table definitions on master and slave can significantly vary
- Usually less writes
  - There are exceptions!

SBR: Strong Sides

- Exists since very first versions
- Table definitions on master and slave can significantly vary
- Usually less writes
- Human readable

SBR: Strong Sides

- Exists since very first versions
- Table definitions on master and slave can significantly vary
- Usually less writes
- Human readable
- Easy to troubleshoot

#7: SHOW BINLOG EVENTS

```
mysql> SHOW BINLOG EVENTS IN ’mysql-bin.000316’ FROM 422;
+------------------+-----+------------+------------+-------------+---------------------------------+
| Log_name         | Pos | Event_type | Server_id  | End_log_pos | Info                            |
+------------------+-----+------------+------------+-------------+---------------------------------+
| mysql-bin.000316 | 422 | Query      | 1456667904 |         509 | BEGIN                           |
| mysql-bin.000316 | 509 | Query      | 1456667904 |         609 | use ‘PgDay‘; update ai set f1=1 |
| mysql-bin.000316 | 609 | Xid        | 1456667904 |         640 | COMMIT /* xid=60328 */          |
+------------------+-----+------------+------------+-------------+---------------------------------+
3 rows in set (0,12 sec)
```

SBR: Weak Sides

- Not all queries are safe
  - Non-deterministic functions
  - MySQL extentions
  - Triggers
  - Mix with non-transactional tables
  - Temporary tables

SBR: Weak Sides

- Not all queries are safe
- Order of events matter!
  - Row-based locks

SBR: Weak Sides

- Not all queries are safe
- Order of events matter!
  - Row-based locks
  - Triggers
    - `SET GLOBAL slave_skip_counter` – No GTIDs!
    - Skip transaction – GTIDs
    - Synchronize tables!

Row-Based Binary Log Format

Client

Binary log

Row-Based Binary Log Format

Client

UPDATE … →

Binary log

Row-Based Binary Log Format

Client

UPDATE ... →

Binary log

SET TIMESTAMP...

Row-Based Binary Log Format

Client

UPDATE ... →

Binary log

SET TIMESTAMP...

SET sql_mode...

Row-Based Binary Log Format

Client

UPDATE ... →

Binary log

SET TIMESTAMP...

SET sql_mode...

Row before changes

Row-Based Binary Log Format

Client

UPDATE ... →

Binary log

SET TIMESTAMP...

SET sql_mode...

Row before changes

Row with changes

RBR: Strong Sides

- Safe
  - You do not need to care about
    - Order of events
    - Triggers
    - Functions
    - Which queries you send to master

RBR: Weak Sidex

- Sensitive for table structures
- More writes
  - --binlog_row_image=FULL | MINIMAL | NOBLOB
- Harder to read

#8: mysqlbinlog

```
$ mysqlbinlog ./mysqld.1/data/master-bin.000001 --start-position=989 --stop-position=1213
...
# at 1167
#160822 14:15:11 server id 1  end_log_pos 1213 CRC32 0x1f346c6b
Update_rows: table id 109 flags: STMT_END_F

BINLOG '
v966VxMBAAAAKwAAAI8EAAAAAG0AAAAAAAEAAm0yAAJ0MQABAwABY2H0oQ==
v966Vx8BAAAALgAAAL0EAAAAAG0AAAAAAAEAAgAB///+BQAAAP4GAAAAa2w0Hw==
'/*!*/;
ROLLBACK /* added by mysqlbinlog */ /*!*/;
SET @@SESSION.GTID_NEXT= 'AUTOMATIC' /* added by mysqlbinlog */ /*!*/;
...
```

#8: mysqlbinlog

```
$ mysqlbinlog -v ./mysqld.1/data/master-bin.000001 --start-position=989 --stop-position=1213
...
# at 1167
#160822 14:15:11 server id 1  end_log_pos 1213 CRC32 0x1f346c6b
Update_rows: table id 109 flags: STMT_END_F

BINLOG '
v966VxMBAAAAKwAAAI8EAAAAAG0AAAAAAAEAAm0yAAJ0MQABAwABY2H0oQ==
v966Vx8BAAAALgAAAL0EAAAAAG0AAAAAAAEAAgAB///+BQAAAP4GAAAAa2w0Hw==
'/*!*/;
### UPDATE `m2`.`t1`
### WHERE
###   @1=5
### SET
###   @1=6
ROLLBACK /* added by mysqlbinlog */ /*!*/;
SET @@SESSION.GTID_NEXT= 'AUTOMATIC' /* added by mysqlbinlog */ /*!*/;
```

Position-Based

- You must specify
  - Name of the master's binary log file
  - Position

Position-Based

- You must specify
  - Name of the master’s binary log file
  - Position
- From the troubleshooting point of view
  - Event executes if on the current position

Position-Based

- You must specify
  - Name of the master’s binary log file
  - Position
- From the troubleshooting point of view
  - Event executes if on the current position
  - Easy to skip

Position-Based

- You must specify
  - Name of the master’s binary log file
  - Position
- From the troubleshooting point of view
  - Event executes if on the current position
  - Easy to skip
  - Easy to move position backward

Position-Based

- You must specify
  - Name of the master’s binary log file
  - Position
- From the troubleshooting point of view
  - Event executes if on the current position
  - Easy to skip
  - Easy to move position backward
  - No conflict resolution

Global Transaction Identifiers (GTID)

- Each transaction has unique number: GTID

Global Transaction Identifiers (GTID)

- Each transaction has unique number: GTID
- MySQL: `AUTO_POSITION=1`

Global Transaction Identifiers (GTID)

- Each transaction has unique number: GTID
- MySQL: `AUTO_POSITION=1`
- MariaDB: `master_use_gtid = { slave_pos | current_pos }`

Global Transaction Identifiers (GTID)

- Each transaction has unique number: GTID
- MySQL: `AUTO_POSITION=1`
- MariaDB: `master_use_gtid = { slave_pos | current_pos }`
- No need to specify binary log and position

Global Transaction Identifiers (GTID)

- Each transaction has unique number: GTID
- MySQL: `AUTO_POSITION=1`
- MariaDB: `master_use_gtid = { slave_pos | current_pos }`
- No need to specify binary log and position
- Hard to skip erroneous event

#9: mysqlslavetrx

```
sveta@thinkie> mysqlslavetrx --gtid-set=fb776095-8474-11e5-ad41-30b5c2208a0f:3 \
               --slaves=root:@127.0.0.1:13001
WARNING: Using a password on the command line interface can be insecure.
#
# GTID set to be skipped for each server:
# - 127.0.0.1@13001: fb776095-8474-11e5-ad41-30b5c2208a0f:3
#
# Injecting empty transactions for ’127.0.0.1:13001’...
#
#...done.
#
```

Complicated Setups

Complicated Setups

- Same specifics and methods

Complicated Setups

- Same specifics and methods
- Multiply complexity on number of channels

Complicated Setups

- Same specifics and methods
- Multiply complexity on number of channels
- Control writes
  - What
  - On which server

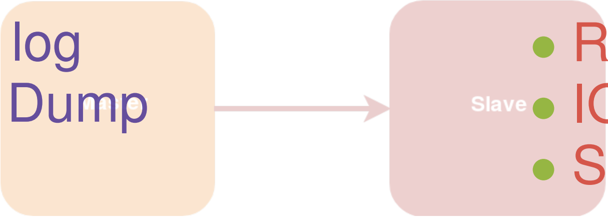
Master and Slave

Master

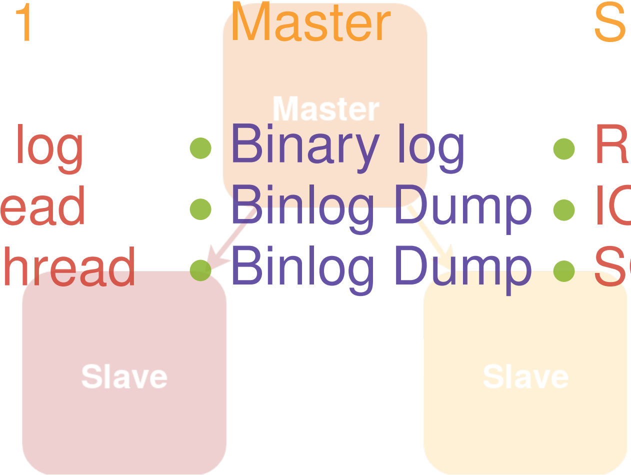
- Binary log
- Binlog Dump

Slave

- Relay log
- IO thread
- SQL thread



Master and Two Slaves



Slave 1

- Relay log
- IO thread
- SQL thread

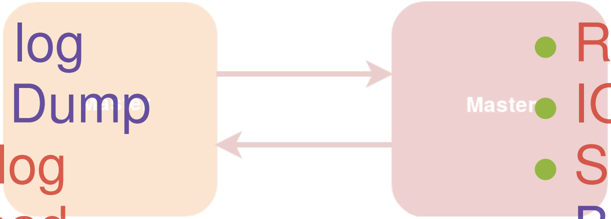
Master

- Binary log
- Binlog Dump
- Binlog Dump

Slave 2

- Relay log
- IO thread
- SQL thread

Circular



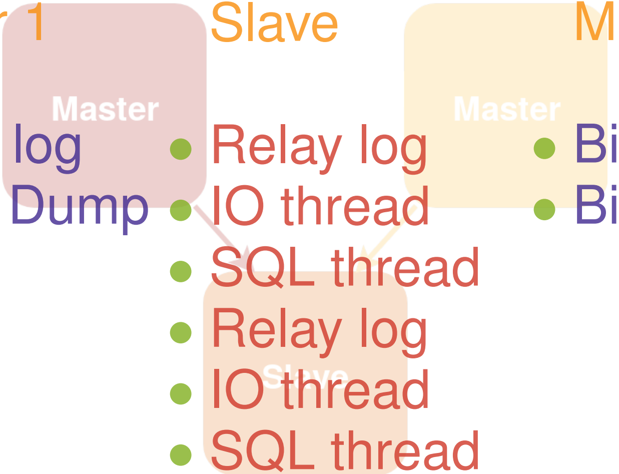
Master 1

- Binary log
- Binlog Dump
- Relay log
- IO thread
- SQL thread

Master 2

- Relay log
- IO thread
- SQL thread
- Binary log
- Binlog Dump

Multiple Masters (Multi-channel): 5.7+



Master 1

- Binary log
- Binlog Dump

Slave

- Relay log
- IO thread
- SQL thread
- Relay log
- IO thread
- SQL thread

Master 2

- Binary log
- Binlog Dump

Multi-Master: Troubleshooting

- Multiple sets of relay logs
- Multiple IO threads
- Multiple SQL threads
- MySQL: `--slave_parallel_workers` for each

Multi-Master: Troubleshooting

- Multiple sets of relay logs
- Multiple IO threads
- Multiple SQL threads
- MySQL: `--slave_parallel_workers` for each
- Independent channels

Multi-Master: Troubleshooting

- Multiple sets of relay logs
- Multiple IO threads
- Multiple SQL threads
- MySQL: `--slave_parallel_workers` for each
- Independent channels
- Error in one stops only one

Multi-Master: Troubleshooting

- Multiple sets of relay logs
- Multiple IO threads
- Multiple SQL threads
- MySQL: `--slave_parallel_workers` for each
- Independent channels
- Error in one stops only one
- No automatic conflict resolution

PERCONA LIVE EUROPE DUBLIN

Summary

Basic Tools

- Error log file

Basic Tools

- Error log file
- On the slave
  - SHOW SLAVE STATUS
  - MySQL: Tables in Performance Schema
  - System database mysql

Basic Tools

- Error log file
- On the slave
- On the master
  - SHOW MASTER STATUS
  - SHOW BINLOG EVENTS
  - mysqlbinlog

Basic Tools

- Error log file
- On the slave
- On the master
- Percona Toolkit

Basic Tools

- Error log file
- On the slave
- On the master
- Percona Toolkit
- MySQL Utilities

Replication Must Know

- Always available, requires setup
- Asynchronous
- Master
  - Keeps all changes in the binary log
    - Two formats: ROW and STATEMENT
- Slave
  - IO thread reads from the master into relay log
  - SQL thread executes updates
    - Multiple SQL threads in 5.6+
    - Multiple channels/sources (masters) in 5.7+
- GTID in 5.6+

Typical Issues

- Master
  - Same as for standalone server
  - More writes and consistency checks

Typical Issues

- Master
- Slave IO thread
  - Common network issues
  - mysql command line client for tests

Typical Issues

- Master
- Slave IO thread
- Slave SQL thread
  - Regular query-related issues
  - Regular storage engine issues
  - Less execution threads than on master

More Information

- Basic Techniques – troubleshooting webinar
- Troubleshooting hardware resource usage
- Introduction into storage engine troubleshoot...
- Percona Toolkit
- MySQL Utilities
- Book MySQL High Availability
- MySQL Replication Team blog

PERCONA LIVE EUROPE DUBLIN

Contact Information

http://www.slideshare.net/SvetaSmirnova

https://twitter.com/svetsmirnova

https://github.com/svetasmirnova

Support Team at Percona Live

- ... Case Study: .IE Continuous Restore ...
  - ■ Marcelo Altmann - Percona, Mick Begley - IE Domain Registry
  - ■ Tuesday 2:20PM-3:10PM @ Goldsmith 3
- A ... Walkthrough on pt-stalk
  - ■ Marcelo Altmann - Percona, Marcos Albe - Percona
  - ■ Wednesday 3:20PM-4:10PM @ Field Suite 2
- Percona XtraBackup Best Practices
  - ■ Marcelo Altmann - Percona
  - ■ Wednesday 4:30PM-4:55PM @ Sky Suite 2
- MySQL-MongoDB-Redis-Cassandra-HBase
  - ■ Marcos Albe - Percona
  - ■ Wednesday 4.55PM-5.20PM @ Sky Suite 2

PERCONA LIVE EUROPE DUBLIN

Thank you!

PERCONA SUPPORTS MORE
THAN JUST OUR SOFTWARE,
WE SUPPORT YOUR SOFTWARE.
PERCONA
Server for MySQL
MariaDB
MySQL
CHAMPIONING
OPEN SOURCE DATABASES
PERCONA