MySQL Replication Troubleshooting

Sveta Smirnova
Principal Technical Services Engineer
August, 25, 2016
Table of Contents

• MySQL Replication Essentials
• Troubleshooting tools
• Master issues
• Slave IO thread issues
• Slave SQL thread issues
• Multi-threaded slave
• Multi-master
MySQL Replication Essentials
Asynchronous

- Slave initiates replication
- It requests a packet from the master
- Master sends the packet
- Master does not check if slave received the packet

From Troubleshooting perspective

- Master does not know if slave is up to date
- Slave can have different table structures, engines
- Any write on slave can break replication
- Data inconsistency may happen and stay unnoticed
Asynchronous

- From Troubleshooting perspective
  - Master does not know if slave is up to date
  - Slave can have different table structures, engines
  - Any write on slave can break replication
  - Data inconsistency may happen and stay unnoticed
Semisynchronous

- After slave receives packet it sends “Ack”
- Master waits for only one “Ack”
Semisynchronous

- From Troubleshooting perspective
  - If there are more than one slave master will wait "Ack" only from one of them
  - Master receives notification only after event is written into relay log
  - Master does not know if event applied by SQL thread successfully or not
  - If timeout occurs without any slave acknowledged event replication fails back to asynchronous
  - Writes on master are slower than for asynchronous replication
Logical

- Master maintains binary (update) log files
- Write happens two times: into engine files and into binary log file
Logical

- From Troubleshooting perspective
  - Extra IO activity: any event written on disk twice
  - You can write on slave
Compare: physical

- Does not exist in MySQL!
  - There are two known closed-source implementation
- Master writes only to engine files
- Engine log files are replicated to slave
Compare: physical

- From Troubleshooting perspective
  - Less IO: an event written on disk only once
  - You cannot write on slave
  - Any inconsistency between master and slave leads to unrecoverable replication failure
Two types of threads

- IO thread
  - Reads events from the master
  - Stores received events in the relay log file
- SQL thread
  - Reads events from the relay log
  - Executes them

From Troubleshooting perspective
Two kinds of issues: connection-related and SQL-related
Have to be diagnosed and fixed differently
Two types of threads

- From Troubleshooting perspective
  - Two kinds of issues: connection-related and SQL-related
  - Have to be diagnosed and fixed differently
Multiple SQL threads

• Since 5.6 you can specify maximum number of parallel SQL threads
Multiple SQL threads

- From Troubleshooting perspective
  - Still single IO thread
  - Performance is better than if one SQL thread used, but can be lower than master’s
  - Error during one thread execution stops all of them
Multiple Masters

- Since 5.7 you can replicate from multiple masters
Multiple Masters

- From Troubleshooting perspective
  - Multiple sets of relay logs
  - Multiple IO threads
  - Multiple SQL threads
  - slave_parallel_workers applies for every channel
  - Error during thread execution stops replication from the affected master only
  - If masters update objects with same name conflict can happen
Position-based replication

- When start replication you need to specify
  - Master’s binary log file name
  - Position in the log file
Position-based replication

- From Troubleshooting perspective
  - Easy to skip events by moving position forward
  - Easy to move position backward: duplicate transactions
  - No check if transaction was already applied
GTID-based replication

- Each transaction receives its own unique id: GTID
- You indicate AUTO_POSITION=1 when point slave to master
- No need to care about binary log file names and positions
- Easy failover
GTID-based replication

- From Troubleshooting perspective
  - Guarantee what no transaction is applied two times
  - Not easy to skip multiple transactions
    - Be careful with expire_logs_days!
    - Use mysqlslavetrx
Binary log format

- Statement-based (SBR)
  - Statements written as they are received from the client
  - Not-safe statements can lead to data inconsistency
    - INSERT IGNORE
    - LIMIT without ORDER BY
    - Non-deterministic functions
    - ...

- Row-based (RBR)
- Mixed
Binary log format

- Statement-based (SBR)
- Row-based (RBR)
  - Can log more data
    - IO
    - Network transfer cost
    - binlog_row_image
  - Data consistency issues and slowdowns may happen for tables without PRIMARY KEY
- Mixed
  - Combines advantages of both formats
Replication essentials: summary

- Always available, needs to be setup
- Asynchronous master-slave

**Master**
- Keeps all updates in the binary log file
  - Two formats: ROW and STATEMENT

**Slave**
- IO thread reads updates from the master and stores them in the relay log file
- SQL thread executes updates
  - Multiple SQL threads since 5.6
  - Multiple masters since 5.7

- GTIDs since 5.6
Troubleshooting tools
Main troubleshooting tools

- Error log file
- Slave
  - `SHOW SLAVE STATUS`
  - Tables in Performance Schema
  - Tables in mysql database
- Master
  - `SHOW MASTER STATUS`
  - `SHOW BINLOG EVENTS`
- `mysqlbinlog`
- `mysql command line client`
2016-08-23T12:11:21.867440Z 4 [ERROR] Slave SQL for channel 'master-1': Could not execute Update_rows event on table m2.t1; Can’t find record in 't1', Error_code: 1032; handler error HA_ERR_END_OF_FILE; the event’s master log master-bin.000001, end_log_pos 1213, Error_code: 1032
2016-08-23T12:11:21.867484Z 4 [ERROR] Error running query, slave SQL thread aborted. Fix the problem, and restart the slave SQL thread with "SLAVE START". We stopped at log 'master-bin.000001' position 989
SHOW SLAVE STATUS

mysql> show slave status \G
***** 1. row **********
  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:
SHOW SLAVE STATUS

Replicate_Wild_Ignore_Table:

Last_Errno: 1032
Last_Error: Could not execute Update_rows event on table m2.t1; Can’t find record in ’t1’, Error_code: 1032; handler error HA_ERR_END_OF_FILE;
the event’s master log master-bin.000001, end_log_pos 1213

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
SHOW SLAVE STATUS

Last_IO_Errno: 0
Last_IO_Error:
Last_SQL_Errno: 1032
Last_SQL_Error: Could not execute Update_rows event on table m2.t1; Can’t find record in 't1', Error_code: 1032; handler error HA_ERR_END_OF_FILE; the event’s master log master-bin.000001, end_log_pos 1213

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
SHOW SLAVE STATUS

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:
mysql> select * from replication_applier_status join
    -> replication_applier_status_by_worker using(channel_name);

*************************** 1. row ***************************
    CHANNEL_NAME: master-1
    SERVICE_STATE: OFF
    REMAINING_DELAY: NULL
    COUNT_TRANSACTIONS_RETRIES: 0
    WORKER_ID: 0
    THREAD_ID: NULL
    SERVICE_STATE: OFF
    LAST_SEEN_TRANSACTION: ANONYMOUS
    LAST_ERROR_NUMBER: 1032
    LAST_ERROR_MESSAGE: Could not execute Update_rows event on table m2.t1;
    Can’t find record in ’t1’, Error_code: 1032; handler error HA_ERR_END_OF FILE;
    the event’s master log master-bin.000001, end_log_pos 1213
    LAST_ERROR_TIMESTAMP: 2016-08-23 15:11:21

...
mysql database

mysql> select * from slave_relay_log_info\G
*************************** 1. row ***************************
 Number_of_lines: 7
 Relay_log_name: ./slave-relay-bin-master@002d1.000004
 Relay_log_pos: 1156
 Master_log_name: master-bin.000001
 Master_log_pos: 989
 Sql_delay: 0
 Number_of_workers: 0
    Id: 1
 Channel_name: master-1
SHOW MASTER STATUS

mysql> show master status\G
*************************** 1. row ***************************
File: master-bin.000005
Position: 154
Binlog_Do_DB:
Binlog_Ignore_DB:
Executed_Gtid_Set:
1 row in set (0,00 sec)
SHOW BINLOG EVENTS

mysql> show binlog events in 'master-bin.000001' from 989;

+-------------------+------+----------------+-----------+-------------+--------------------------------+
| Log_name | Pos | Event_type     | Server_id | End_log_pos | Info                                      |
+-------------------+------+----------------+-----------+-------------+--------------------------------+
| master-bin.000001 | 989 | Anonymous_Gtid | 1         | 1054        | SET @@SESSION.GTID_NEXT= ... |
| master-bin.000001 | 1054| Query          | 1         | 1124        | BEGIN                                    |
| master-bin.000001 | 1124| Table_map      | 1         | 1167        | table_id: 109 (m2.t1)                   |
| master-bin.000001 | 1167| Update_rows    | 1         | 1213        | table_id: 109 flags: STMTE_END_F     |
| master-bin.000001 | 1213| Xid            | 1         | 1244        | COMMIT /* xid=64 */                     |
+-------------------+------+----------------+-----------+-------------+--------------------------------+

5 rows in set (0,00 sec)
$ mysqlbinlog -v var/mysqld.1/data/master-bin.000001 --start-position=989 --stop-position=1213
/*!50530 SET @@SESSION.PSEUDO_SLAVE_MODE=1*/;
/*!50003 SET @OLD_COMPLETION_TYPE=@@COMPLETION_TYPE,COMPLETION_TYPE=0*/;
DELIMITER */*

# at 4
#160822 14:01:44 server id 1 end_log_pos 123 CRC32 0xc6ab0154 Start: binlog v 4,
server v 5.7.13-debug-log created 160822 14:01:44 at startup
ROLLBACK/!*!*/;
BINLOG 'mNu6Vw8BAAAAdwAAAAHsAAAAAAQANS43LjEzLWR1YnVnLWxvZwAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAACAAAAAAAAAAAAAAAAAAACQ7pXEzgNAAGAEgAEBAQEElgAAxwAEggAAAAICAgCAAAAACgoKKioAEjQAAVQBoQY=

# at 989
#160822 14:15:11 server id 1 end_log_pos 1054 CRC32 0xe388e040 Anonymous_GTID last_committed=4
sequence_number=5
SET @@SESSION.GTID_NEXT= 'ANONYMOUS'/*!*/;
SET TIMESTAMP=1471864511/*!*/;
SET @@session.pseudo_thread_id=9/*!*/;
SET @@session.foreign_key_checks=1, @@session.sql_auto_is_null=0, @@session.unique_checks=1,@@session.autocommit=1/*!*/;
SET @@session.sql_mode=1436549152/*!*/;
SET @@session.auto_increment_increment=1, @@session.auto_increment_offset=1/*!*/;
/*!*/\C utf8 *//*!*/;
SET @@session.character_set_client=33,@@session.collation_connection=33,@@session.collation_server=8/*!*/;
SET @@session.lc_time_names=0/*!*/;
SET @@session.collation_database=DEFAULT/*!*/;
BEGIN
/*!*/;
# at 1124
#160822 14:15:11 server id 1 end_log_pos 1167 CRC32 0xa1ce6163
Table_map: 'm2'.'t1' mapped to number 109
mysqlbinlog

# 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 'v966VxMBAAAKwAAAI8EAAAAAG0AAAAAAAEEAAm0yAAJOMQABAwABY2H0oQ== v966Vx8BAAAAALgAAAL0EAAAAAG0AAAAAAAEEAAgAB///+BQAAAP4GAAAAa2w0Hw== */!**/;
### UPDATE 'm2'.'t1'
### WHERE
### @1=5
### SET
### @1=6
ROLLBACK /* added by mysqlbinlog */ /*!**/;
SET @@SESSION.GTID_NEXT= 'AUTOMATIC' /* added by mysqlbinlog */ /*!**/;
DELIMITER ;
# End of log file
/*!50003 SET COMPLETION_TYPE=@OLD_COMPLETION_TYPE*/;
/*!50530 SET @@SESSION.PSEUDO_SLAVE_MODE=0*/;
Tables in Performance Schema

- No need to parse SHOW output
- Configuration
- IO thread
- SQL thread
Tables in Performance Schema

- No need to parse SHOW output
- Configuration
  - replication_connection_configuration
  - replication_applier_configuration
- IO thread
- SQL thread
Tables in Performance Schema

- No need to parse SHOW output
- Configuration
- IO thread
  - replication_connection_status
- SQL thread
Tables in Performance Schema

- No need to parse SHOW output
- Configuration
- IO thread
- SQL thread
  - replication_applier_status
  - replication_applier_status_by_coordinator
  - replication_applier_status_by_worker - MTS only
mysql> select * from replication_connection_configuration
    -> join replication_applier_configuration using(channel_name)\G
*************************** 1. row ***************************
CHANNEL_NAME:
    HOST: 127.0.0.1
    PORT: 13000
    USER: root
NETWORK_INTERFACE:
    AUTO_POSITION: 1
    SSL_ALLOWED: NO
    SSL_CA_FILE:
...
Performance Schema: Configuration

...  
CONNECTION_RETRY_INTERVAL: 60  
CONNECTION_RETRY_COUNT: 10  
HEARTBEAT_INTERVAL: 60.000  
CHANNEL_NAME:  
DESIRED_DELAY: 0  
1 row in set (0.00 sec)
### Performance Schema: State of IO Thread

```sql
mysql> select * from replication_connection_status\G
```

```
*************************** 1. row ***************************
CHANNEL_NAME:          
GROUP_NAME:             
SOURCE_UUID: d0753e78-14ec-11e5-b3fb-28b2bd7442fd
THREAD_ID: 21
SERVICE_STATE: ON
COUNT_RECEIVED_HEARTBEATS: 17
LAST_HEARTBEAT_TIMESTAMP: 2015-06-17 15:49:08
RECEIVED_TRANSACTION_SET:
    LAST_ERROR_NUMBER: 0
    LAST_ERROR_MESSAGE:  
    LAST_ERROR_TIMESTAMP: 0000-00-00 00:00:00
1 row in set (0.00 sec)
```
Performance Schema: State of SQL Thread

• Single master

```sql
mysql> select * from replication_applier_status join replication_applier_status_by_coordinator using(channel_name)\G
```

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

CHANNEL_NAME:            
SERVICE_STATE: ON        
REMAINING_DELAY: NULL    
COUNT_TRANSACTIONS_RETRIES: 0
THREAD_ID: 22            
SERVICE_STATE: ON        
LAST_ERROR_NUMBER: 0     
LAST_ERROR_MESSAGE:      
LAST_ERROR_TIMESTAMP: 0000-00-00 00:00:00

1 row in set (0.00 sec)
```

• Multiple masters

27 www.percona.com
Performance Schema: State of SQL Thread

- Single master
- Multiple masters

```sql
mysql> select * from replication_applier_status join replication_applier_status_by_worker using(channel_name)\G
```

```
*************************** 1. row ***************************
CHANNEL_NAME: master-1
SERVICE_STATE: OFF
REMAINING_DELAY: NULL
COUNT_TRANSACTIONS_RETRIES: 0
    WORKER_ID: 0
    THREAD_ID: NULL
SERVICE_STATE: OFF
LAST_SEEN_TRANSACTION: ANONYMOUS
    LAST_ERROR_NUMBER: 1032
    LAST_ERROR_MESSAGE: Could not execute Update_rows event on table m2.t1; Can’t find record in ’t1’, Error_code: 1032; handler error HA_ERR_END_OF_FILE; the event’s master log master-bin.000001, end_log_pos 1213
LAST_ERROR_TIMESTAMP: 2016-08-23 15:48:20
```
Performance Schema: State of SQL Thread

- Single master
- Multiple masters

<table>
<thead>
<tr>
<th>CHANNEL_NAME: master-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>SERVICE_STATE: ON</td>
</tr>
<tr>
<td>REMAINING_DELAY: NULL</td>
</tr>
<tr>
<td>COUNT_TRANSACTIONS_RETRIES: 0</td>
</tr>
<tr>
<td>WORKER_ID: 0</td>
</tr>
<tr>
<td>THREAD_ID: 42</td>
</tr>
<tr>
<td>SERVICE_STATE: ON</td>
</tr>
<tr>
<td>LAST_SEEN_TRANSACTION: ANONYMOUS</td>
</tr>
<tr>
<td>LAST_ERROR_NUMBER: 0</td>
</tr>
<tr>
<td>LAST_ERROR_MESSAGE:</td>
</tr>
<tr>
<td>LAST_ERROR_TIMESTAMP: 0000-00-00 00:00:00</td>
</tr>
</tbody>
</table>

2 rows in set (0,00 sec)
Troubleshooting Toolkits

- **Percona Toolkit**
  - *pt-table-checksum*
    - Checks master and slave for consistency
  - *pt-table-sync*
    - Fixes consistency issues
  - *pt-slave-find*
    - Displays replication topology
Troubleshooting Toolkits

- MySQL Utilities
  - `mysqlrplcheck`
    - Checks replication pre-requisites
  - `mysqlrplshow`
    - Displays replication topology
  - `mysqlrplsync`
    - Checks master and slave for consistency
  - `mysqlslavetrx`
    - Skips 1-N transactions
Troubleshooting Toolkits

- MySQL Utilities
  - `mysqladbcompare`
    - Compares two databases for changes
  - `mysqldiff`
    - Compares object definitions
  - `mysqlserverinfo`
    - Displays basic server options, such as port and datadir
    - Replication-oriented
Master issues
Performance

- More IO
  - `binlog_row_image = FULL | MINIMAL | NOBLOB`
  - `binlog_cache_size`
    - Watch `Binlog.cache_disk_use`
  - `binlog_stmt_cache_size`
    - Watch `Binlog_stmt_cache_disk_use`

- Synchronization
  - `binlog_sync`
  - Be aware, do not change!
Behavior

- Binary log lifetime
  - `expire_log_days`
- Correlation with transactions
  - SBR does not work with READ COMMITTED and READ UNCOMMITTED
- Order of events in the binary log
  - Non-deterministic events with SBR
Slave IO thread issues
Communication issues

SHOW SLAVE STATUS

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' - retry-time: 60 retries: 1

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:
Communication issues

- **SHOW SLAVE STATUS**
- **P_S.replication_connection_status**

```sql
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)
```

- Check slave error log
- Access error
- Try to connect using normal MySQL client using slave credentials
- SHOW GRANTS
- Fix privileges on master
- Restart slave
Communication issues

- SHOW SLAVE STATUS
- P_S.replication_connection_status
- Check slave error log

```
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
```
Communication issues

- SHOW SLAVE STATUS
- P_S.replication_connection_status
- Check slave error log
- Access error

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

- SHOW SLAVE STATUS
- P_S.replication_connection_status
- Check slave error log
- Access error
  - Try to connect using normal MySQL client using slave credentials
    $ mysql -h127.0.0.1 -P13000 -uroot -pbar
    Warning: Using a password on the command line interface can be insecure.
    ERROR 1045 (28000): Access denied for user ‘root’@’localhost’ (using password: YES)
Communication issues

- SHOW SLAVE STATUS
- P.S.replication_connection_status
- Check slave error log
- Access error
  - Try to connect using normal MySQL client using slave credentials
    - SHOW GRANTS
      
      ```
      mysql> SHOW GRANTS;
      +----------------------------------+
      | Grants for foo@%                |
      +----------------------------------+
      | GRANT SELECT ON *.* TO 'foo'@'%' |
      +----------------------------------+
      1 row in set (0.00 sec)
      ```
  - Fix privileges on master
  - Restart slave
Communication issues

- SHOW SLAVE STATUS
- P_S.replication_connection_status
- Check slave error log
- Access error
  - Try to connect using normal MySQL client using slave credentials
    - SHOW GRANTS
  - Fix privileges on master
  - Restart slave
Performance issues

- Same methods as for regular client-server network issues
- Check with MySQL command line client if needed
- See also Troubleshooting hardware resource usage
Slave SQL thread issues
SQL thread: typical issues

• Simple master-slave
  • Data is different on master and slave
    • Replication event can not be applied
  • Different errors on master and slave
  • Slave lags far behind the master
• Circular replication or other writes in addition to slave SQL thread
  • Data is different on master and slave
Data is different on master and slave

- Was the table modified besides the SQL thread?
  - How?
  - Can it affect content of the table in the wrong way?
- Are the table definitions same on master and slave?
  - Percona Toolkit
    - pt-table-checksum, pt-table-sync
  - MySQL Utilities
    - mysqlrplsync, mysqldbcompare, mysqldiff
- Maybe master events were applied in wrong order?
  - Use mysqlbinlog to find queries caused the issue
  - Check application, running on master
Events applied in different order

- Happens only with STATEMENT format
- Row lock issues
- Triggers
  - SET GLOBAL slave_skip_counter - No GTIDs!
  - Skip transaction - GTIDs
  - Synchronize tables!
- Different options
  - Start slave with master’s options, then check
  - Happens in older versions
Slave lags behind master

- Threads
  - Master runs in multiple update threads
  - Slave uses single
- Seconds_behind_master is growing - Not 100 % reliable!
- Tune slave performance
  - Multi-threaded slave
    - One thread per database in 5.6
    - Affected by concurrency issues
  - Indexes on slave only
    - For statement-based replication
Multi-threaded slave
Performance

- Single relay log
  - Speed still may be not same as on master in high concurrent environment
- slave_parallel_workers
- slave_parallel_type=DATABASE | LOGICAL_CLOCK
Wrong behavior

- Same as with single-threaded slave
- Error in one SQL thread stops them all

```sql
mysql> select WORKER_ID, SERVICE_STATE, LAST_SEEN_TRANSACTION, LAST_ERROR_NUMBER, 
    -> LAST_ERROR_MESSAGE from performance_schema.replication_applier_status_by_worker

*************************** 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...
```
Multi-master
Specifics

- Replication should be configured for each channel
- Mixing masters with GTID and without GTIDs is possible
- Same issues as with traditional replication
- Replication filters are same for all channels
Summary
Summary

- Master issues
  - Same as for regular server
  - Additional IO and safety checks
- Slave IO thread
  - Common network-related issues
  - Use mysql command line client to test
- Slave SQL thread
  - Common SQL-related issues
  - Common engine-specific issues
  - Less concurrency than on master
More information

- Basic Techniques - troubleshooting webinar
- Troubleshooting hardware resource usage
- Introduction into storage engine troubleshooting
- Percona Toolkit
- MySQL Utilities
- MySQL High Availability book
- MySQL Replication Team blog
Place for your questions

???
Thank you!

http://www.slideshare.net/SvetaSmirnova
https://twitter.com/svetsmirnova