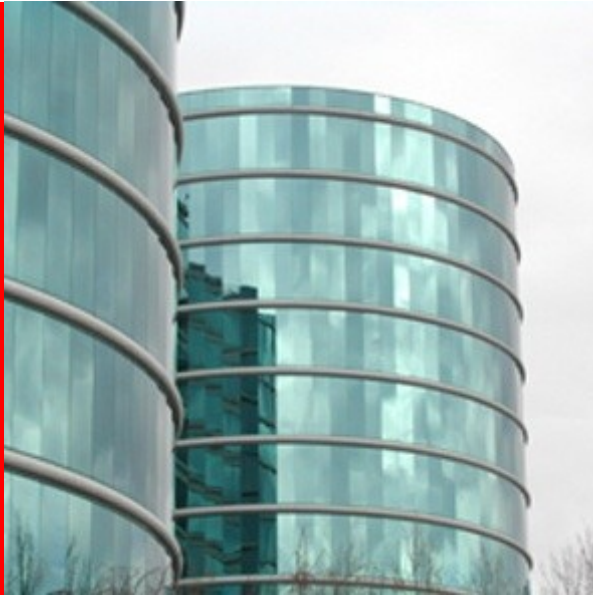


ORACLE®



ORACLE[®]

MySQL: Replication

Keith Larson

keith.larson@oracle.com

MySQL Community Manager

sqlhjalp.blogspot.com

sqlhjalp.com/pdf/2012_Scale_Replication.pdf



Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions.

The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.



Who am I and who are you?

Keith Larson

keith.larson@oracle.com

MySQL Community Manager

<http://sqlhjalp.blogspot.com/>

Started with MySQL during the dot.com days.

Primary real world work was with a MySQL InnoDB replicated chain environment that easily held over 4 billion rows of user data.

Numerous other sites developed on LAMP stack over the last 13 years.

Also a movie buff :)

Who are you?

DBAs?

Developers?

Already have replicated databases?

Have up to date slaves?

Cluster Users or Cluster curious?



Session Agenda

- MySQL Replication Overview
- Replication Configuration
- Examples of a real world set up
- MySQL 5.6 Replication Features
- Monitoring – MySQL Enterprise Monitor



ORACLE®

Who Uses Replication ?



"In my opinion, MySQL is the only database we would ever trust to power the Zappos.com website."



"As a leader in our field, we are committed to providing the best service to our users, and a web experience that meets members expectations and that starts with IT"



"craigslist infrastructure could not have handled the exponential growth in traffic without MySQL."



"They have a master server for all writes and slave servers for most Reads. The secret truth they claim behind configuring the master and slave machines is to make sure the slave machines are faster than the masters"

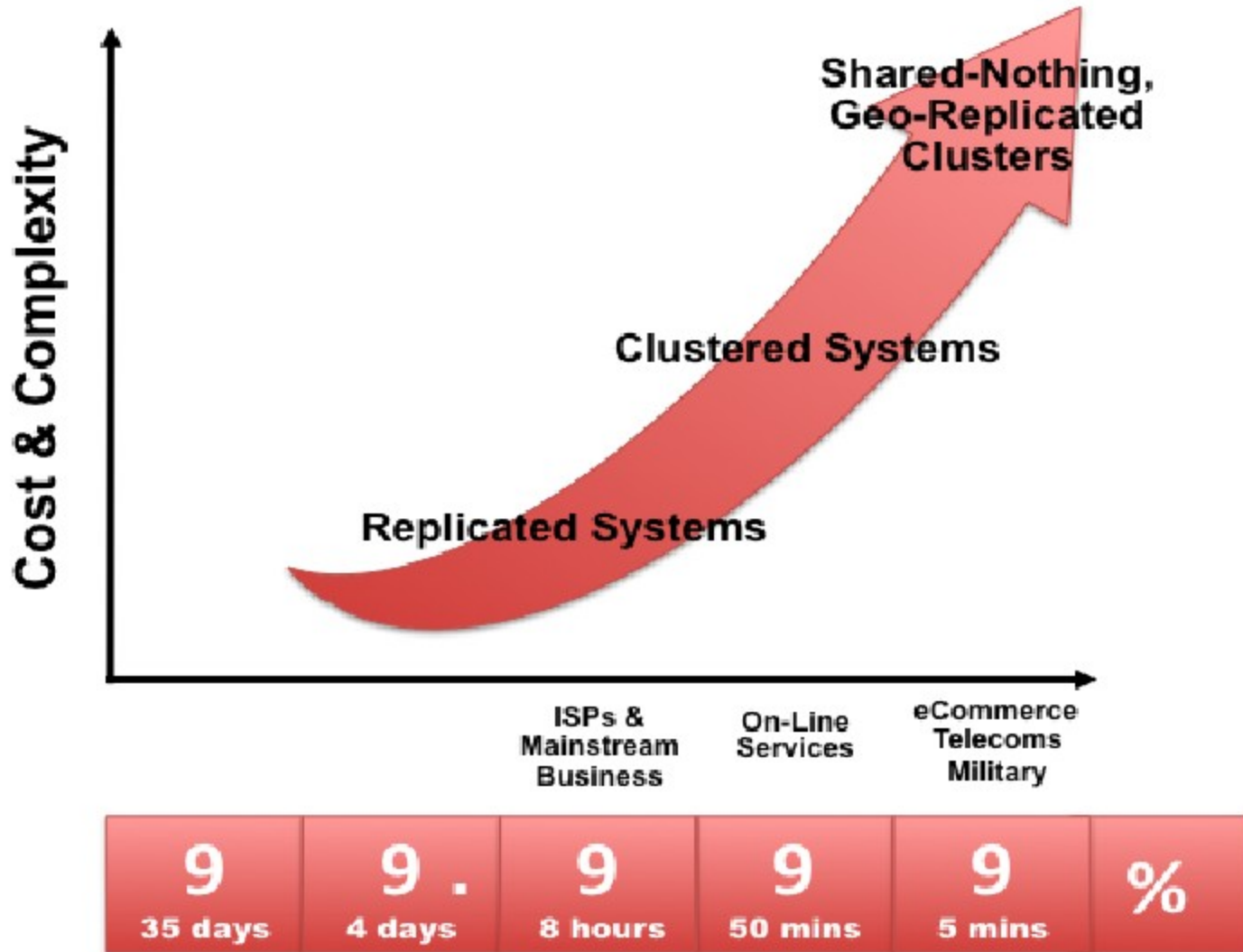


"On any given day we can sell close to 300,000 tickets on the Web site using MySQL as the database to search for events. It is amazing."



"We are one of the largest MySQL web sites in production"

Who Uses Replication ?



ORACLE®

MySQL Replication Overview



Duplicates database from a
“master” to a “slave”

Redundant copies of the data
provide foundation for High
Availability

Scale out by distributing queries
across the replication cluster



ORACLE®

MySQL Replication Overview



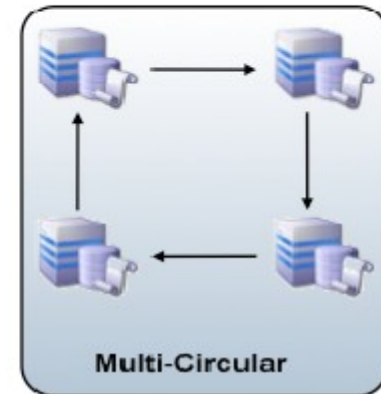
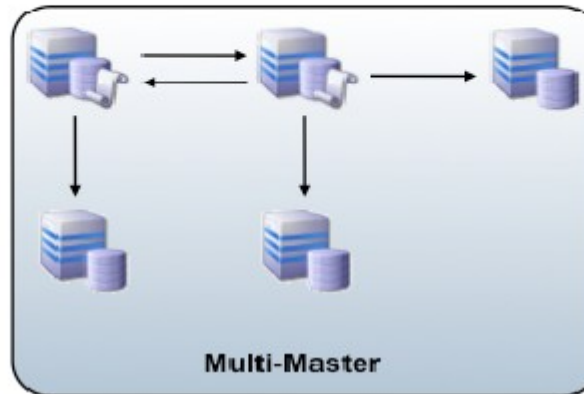
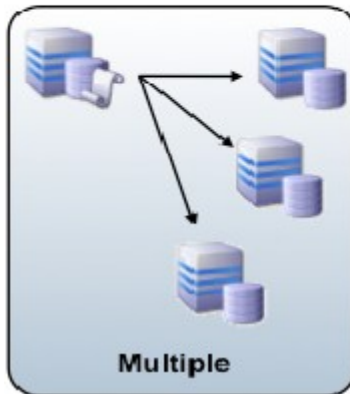
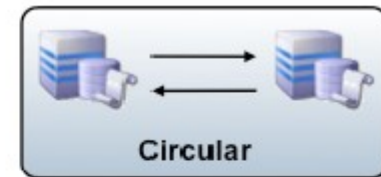
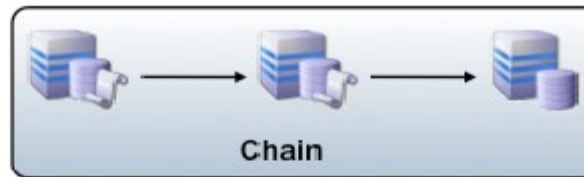
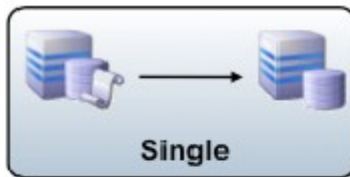
- Native in MySQL
- Replicate within and across data centers
- Failover is either scripted or provided by additional middleware
- Supports Asynchronous (standard) and Semi-Synchronous replication
- Each slave adds minimal load on master



ORACLE®

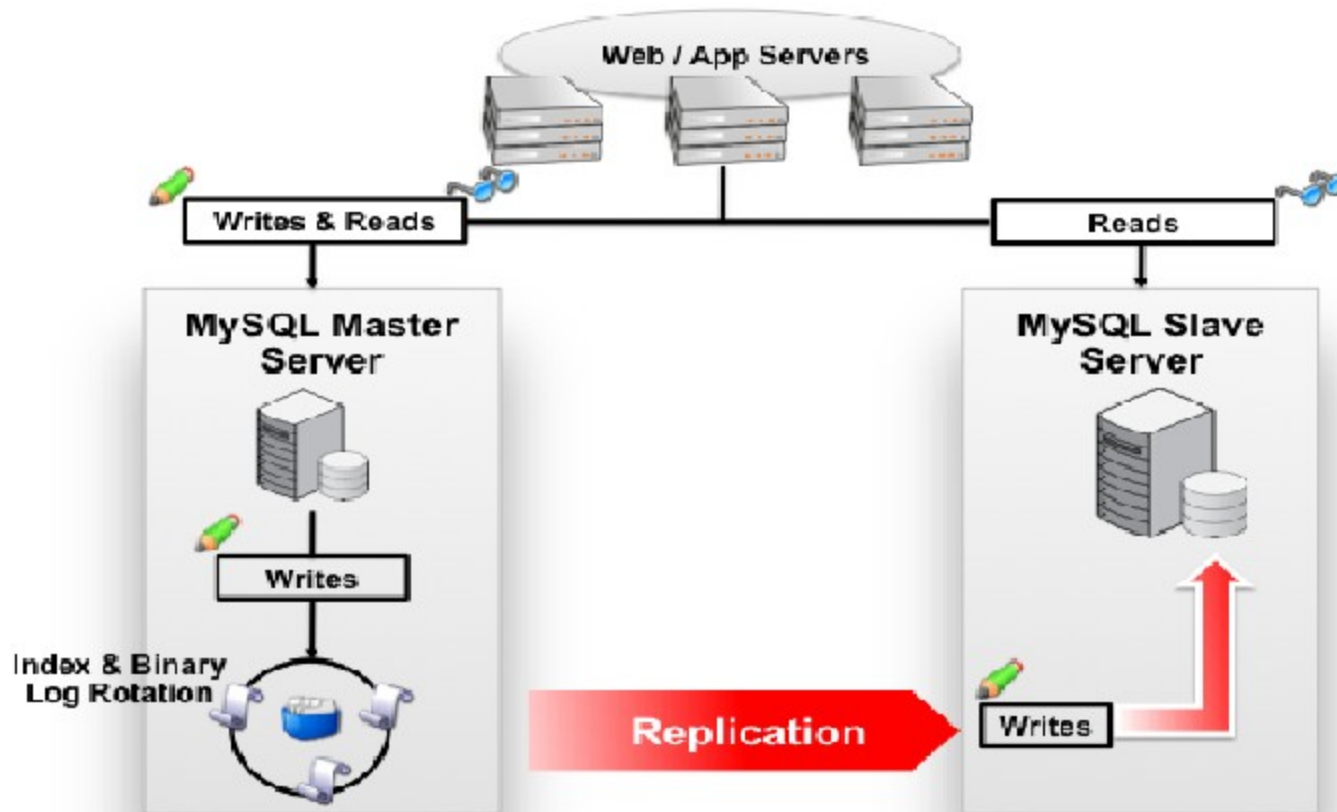
MySQL Replication Overview

Replication Topologies



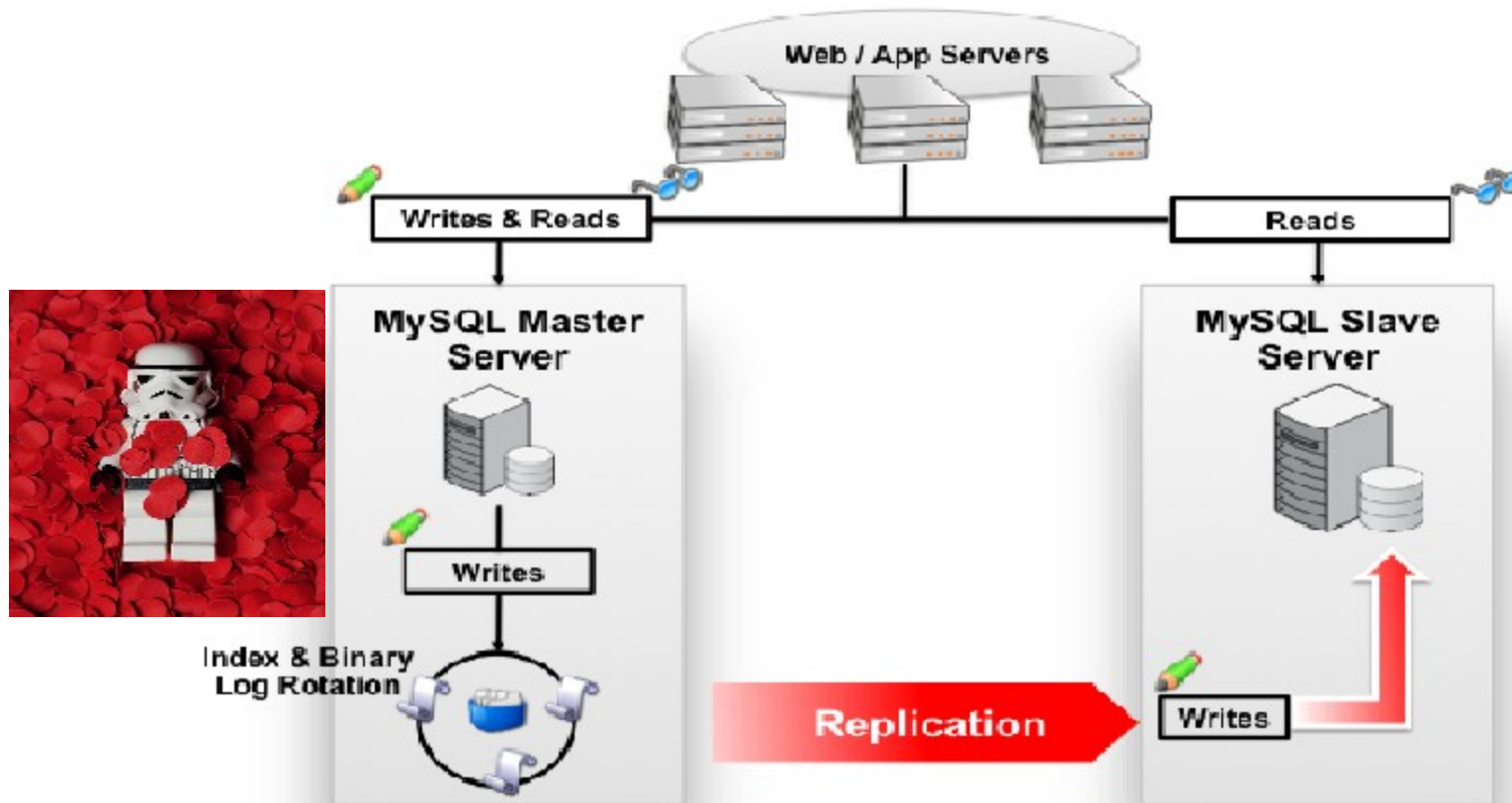
ORACLE®

MySQL Replication Overview



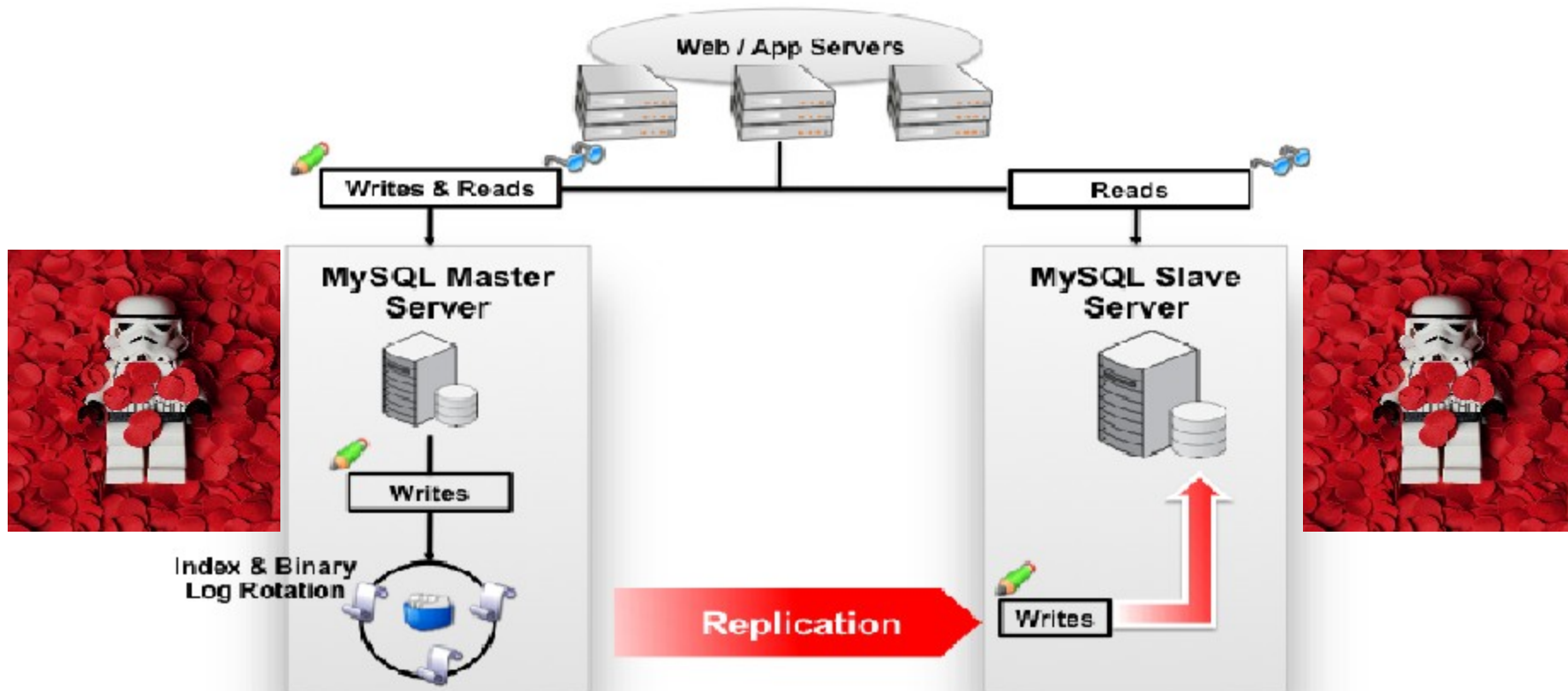
ORACLE®

MySQL Replication Overview



ORACLE®

MySQL Replication Overview



ORACLE®

MySQL Replication Overview

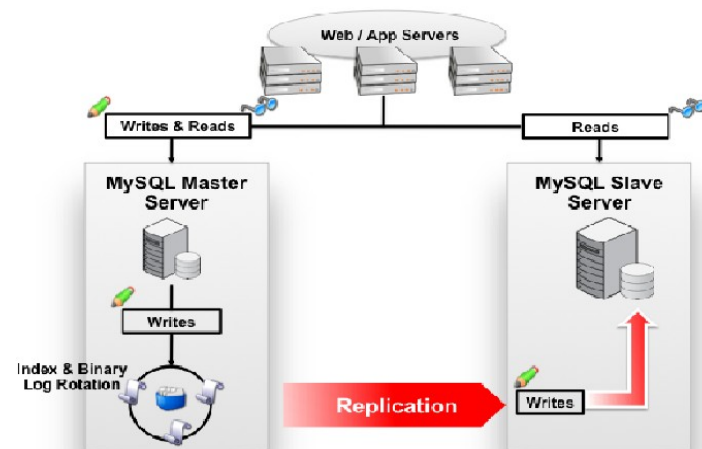
- Used for Scalability and HA
- Asynchronous as standard
- Semi-Synchronous support added in MySQL 5.5
- Each slave adds minimal load on master

Replication Threads

- Binlog dump thread
- Slave I/O thread
- Slave SQL thread

Replication Files

- relay log
- master info log
- relay log info log



ORACLE®

MySQL Replication Overview

- Used for Scalability and HA
- Asynchronous as standard
- Semi-Synchronous support added in MySQL 5.5
- Each slave adds minimal load on master

Replication Threads

- **Binlog dump thread**
- Slave I/O thread
- Slave SQL thread

Replication Files

- relay log
- master info log
- relay log info log

The master creates a thread to send the binary log contents to a slave when the slave connects. This thread can be identified in the output of SHOW PROCESSLIST on the master as the Binlog Dump thread.

• The binlog dump thread acquires a lock on the master's binary log for reading each event that is to be sent to the slave. As soon as the event has been read, the lock is released, even before the event is sent to the slave.

```
***** 2. row *****
```

```
Id: 27
```

```
User: replication
```

```
Host: 192.168.0.11:47129
```

```
db: NULL
```

```
Command: Binlog Dump
```

```
Time: 499
```

```
State: Master has sent all binlog to slave; waiting for binlog to  
be updated
```

```
Info: NULL
```



MySQL Replication Overview

- Used for Scalability and HA
- Asynchronous as standard
- Semi-Synchronous support added in MySQL 5.5
- Each slave adds minimal load on master

Replication Threads

- Binlog dump thread
- **Slave I/O thread**
- Slave SQL thread

When a **START SLAVE** statement is issued on a slave server, the slave creates an I/O thread, which connects to the master and asks it to send the updates recorded in its binary logs.

The slave I/O thread reads the updates that the master's Binlog Dump thread sends and copies them to local files that comprise the slave's relay log.

Replication Files

- relay log
- master info log
- relay log info log

The state of this thread is shown as **Slave_IO_running** in the output of **SHOW SLAVE STATUS** or as **Slave_running** in the output of **SHOW STATUS**.

```
Slave_IO_Running: Yes  
| Slave_running | ON |
```



MySQL Replication Overview

- Used for Scalability and HA
- Asynchronous as standard
- Semi-Synchronous support added in MySQL 5.5
- Each slave adds minimal load on master

Replication Threads

- Binlog dump thread
- Slave I/O thread
- **Slave SQL thread**

The slave creates an SQL thread to read the relay log that is written by the slave I/O thread and execute the events contained therein.

Replication Files

- relay log
- master info log
- relay log info log

Slave_SQL_Running: Yes



MySQL Replication Overview

- Used for Scalability and HA
- Asynchronous as standard
- Semi-Synchronous support added in MySQL 5.5
- Each slave adds minimal load on master

Replication Threads

- Binlog dump thread
- Slave I/O thread
- Slave SQL thread

Replication Files

- **relay log**
- master info log
- relay log info log

The relay log consists of the events read from the binary log of the master and written by the slave I/O thread. Events in the relay log are executed on the slave as part of the SQL thread. caption text here.



MySQL Replication Overview

- Used for Scalability and HA
- Asynchronous as standard
- Semi-Synchronous support added in MySQL 5.5
- Each slave adds minimal load on master

Replication Threads

- Binlog dump thread
- Slave I/O thread
- Slave SQL thread

Replication Files

- relay log
- **master info log**
- relay log info log

The master info log contains status and current configuration information for the slave's connection to the master. This log holds information on the master host name, login credentials, and coordinates indicating how far the slave has read from the master's binary log.



MySQL Replication Overview

- Used for Scalability and HA
- Asynchronous as standard
- Semi-Synchronous support added in MySQL 5.5
- Each slave adds minimal load on master

Replication Threads

- Binlog dump thread
- Slave I/O thread
- Slave SQL thread

Replication Files

- relay log
- **master info log**
- relay log info log

The master info log contains status and current configuration information for the slave's connection to the master. This log holds information on the master host name, login credentials, and coordinates indicating how far the slave has read from the master's binary log.



MySQL Replication Overview

- Used for Scalability and HA
- Asynchronous as standard
- Semi-Synchronous support added in MySQL 5.5
- Each slave adds minimal load on master

Replication Threads

- Binlog dump thread
- Slave I/O thread
- Slave SQL thread

Replication Files

- relay log
- master info log
- **relay log info log**

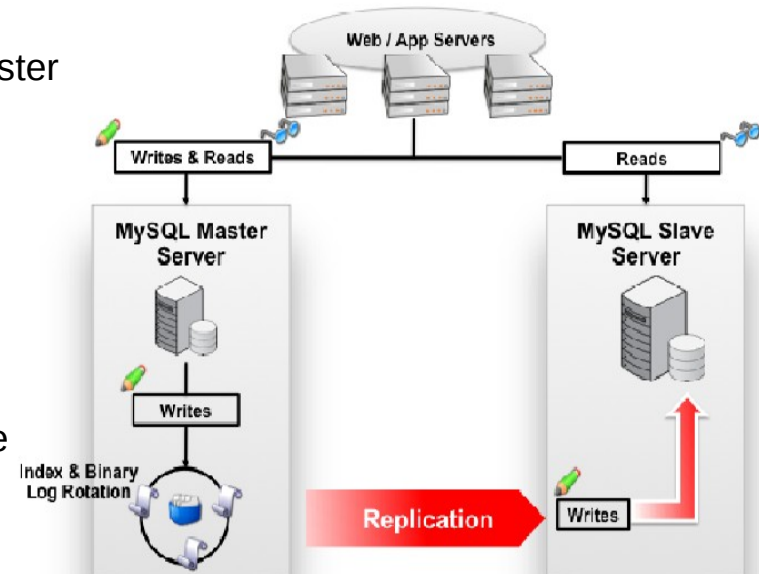
The relay log info log holds status information about the execution point within the slave's relay log



MySQL Replication Overview

Replication Formats

- Statement-based replication (SBR)
 - Advantages
 - Less data written to log files
 - taking and restoring from backups are faster
 - Disadvantages
 - Some functions or queries are nondeterministic and hard to replicate
- Row-based replication (RBR)
 - Advantages
 - safest form of replication
 - Fewer row locks are required on the slave
 - Disadvantages
 - generate more data that must be logged
 - Blob values take longer to replicate
- Mixed-format
 - statement-based logging is used by default
 - automatically switches to row-based logging in particular cases



<http://dev.mysql.com/doc/refman/5.5/en/replication-formats.html>

ORACLE®

MySQL Replication Overview

Asynchronous is standard

- events are occurring independently
- The master writes events to its binary log but does not know whether or when a slave has retrieved and processed them
- if the master crashes, transactions that it has committed might not have been transmitted to any slave

Semi-Synchronous support added in MySQL 5.5

- Original patch: Mark Callaghan and Wei Li, Google
Adoptions: Zhenxing He, Sun Microsystems
- acknowledges receipt of a transaction's events only after the events have been written to its relay log and flushed to disk
- best for close servers communicating over fast networks
- timeout occurs without any slave having acknowledged the transaction, the master reverts to asynchronous replication. When at least one semisynchronous slave catches up, the master returns to semisynchronous replication.



<http://dev.mysql.com/doc/refman/5.5/en/replication-semisync.html>

ORACLE

MySQL Steps in Replication

Master Setup:

```
# vi /etc/my.cnf
[mysqld]
server-id=1
log-bin = /var/lib/mysql/yoda-bin
```

Start and Log into MySQL

```
master_yoda>show master status\G
```

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

```
File: yoda-bin.000001
```

```
Position: 114
```

```
Binlog_Do_DB:
```

```
Binlog_Ignore_DB:
```

```
1 row in set (0.00 sec)
```

```
# mysqldump -p --all-databases --master-data=2 > /tmp/replication_example.sql
```

THIS LOCKS THE DATABASE!



<http://dev.mysql.com/doc/refman/5.5/en/replication-howto-masterbaseconfig.html>

ORACLE®

MySQL Steps in Replication

On Master :

```
mysql_yoda>CREATE USER 'replication'@'192.168.0.%' IDENTIFIED BY 'slavepass';  
mysql_yoda>GRANT REPLICATION SLAVE ON *.* TO 'replication'@'%';  
mysql_yoda>flush privileges;
```

Adjust all firewall rules if required for MySQL Port. (3306)



<http://dev.mysql.com/doc/refman/5.5/en/replication-howto-masterbaseconfig.html>

ORACLE®

MySQL Steps in Replication

Slave Setup:

```
vi /etc/my.cnf
[mysqld]
Server-id=2
relay-log=/var/lib/mysql/luke-relay-bin
```

```
# optional below
log-bin = /var/lib/mysql/luke-bin
```

```
luke>show master status\G
```

```
***** 1. row *****
      File: luke-bin.000003
      Position: 114
      Binlog_Do_DB:
      Binlog_Ignore_DB:
      1 row in set (0.00 sec)
```

LOAD DATA:

```
# mysql --user=root -p < /tmp/replication_example.sql
```



ORACLE®

MySQL Steps in Replication

On Slave :

```
mysql_luke> CHANGE MASTER TO  
  MASTER_HOST='yoda',  
  MASTER_USER='replication',  
  MASTER_PASSWORD='slavepass',  
  MASTER_PORT=3306,  
  MASTER_LOG_FILE='yoda-bin.000002',  
  MASTER_LOG_POS=83415,  
  MASTER_CONNECT_RETRY=10;
```

We gathered this info from the mysqldump file via the
“ **--master-data=2** ” flag.

```
-- CHANGE MASTER TO  
MASTER_LOG_FILE='yoda-bin.000002',  
MASTER_LOG_POS=83415;
```

```
Mysql > start slave;  
Query OK, 0 rows affected (0.00 sec)
```



ORACLE®

MySQL Steps in Replication

On Slave :

```
mysql_luke> show slave status\G
```

```
mysql_luke> show slave status\G
```

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

```
Slave_IO_State: Waiting for master to send event
```

```
Master_Host: yoda
```

```
Master_User: replication
```

```
Master_Port: 3306
```

```
Connect_Retry: 10
```

```
Master_Log_File: yoda-bin.000003
```

```
Read_Master_Log_Pos: 323
```

```
Relay_Log_File: luke-relay-bin.000004
```

```
Relay_Log_Pos: 475
```

```
Relay_Master_Log_File: yoda-bin.000003
```

```
Slave_IO_Running: Yes
```

```
Slave_SQL_Running: Yes
```

```
Master_Server_Id: 1
```

```
Master_UUID: 75d407df-2be4-11e1-9668-b4be9bce39b0
```

```
Seconds_Behind_Master: 0
```

```
SQL_Remaining_Delay: NULL
```



ORACLE®

MySQL Steps in Replication

Setting up replication

```
mysqlreplicate \ --master=root@master.example.com \  
--slave=root@slave.example.com \ --rpl-user=repl:xyzzz
```

<http://wb.mysql.com/utilities/man/mysqlreplicate.html>

Require master and slave to be off-line!



<https://github.com/greyrl/generaltools/blob/master/mysqlreplicate.py>



ORACLE®

MySQL Steps in Replication

- DB Luke processes all of DB Yoda's data via the relay log.
- If Vadar adjusts a table within DB Luke and then DB Luke tries to apply duplicate data you will **crash** replication.



MySQL Steps in Replication

Master to Master

```
mysql> SET GLOBAL auto_increment_offset=2;
mysql> SET GLOBAL auto_increment_increment=2;
```

```
mysql_yoda>show global variables like '%auto_increment%';
```

Variable_name	Value
auto_increment_increment	2
auto_increment_offset	1

```
mysql_luke>show global variables like '%auto_increment%';
```

Variable_name	Value
auto_increment_increment	2
auto_increment_offset	2

```
mysql_luke>SHOW MASTER STATUS\G
```

```
***** 1. row *****
File: luke-bin.000005
Position: 295
```

```
mysql_yoda>CHANGE MASTER TO
MASTER_HOST='luke',
MASTER_USER='replication2',
MASTER_PASSWORD='slavepass',
MASTER_PORT=3306,
MASTER_LOG_FILE='luke-bin.000005',
MASTER_LOG_POS=295,
MASTER_CONNECT_RETRY=10;
mysql_yoda> start slave;
```



- Slave Tables for Replication Information
- Replication Event Checksums
- Multi-Threaded Slave
- Time Delayed Replication
- Optimized Row Based Replication
- Informational Log Events
- Remote Backup of Binary logs
- Global Transaction IDs
- Golden Gate Replication



<http://dev.mysql.com/doc/refman/5.6/en/replication-features.html>

<http://dev.mysql.com/downloads/mysql/#downloads> Under Development Releases

Slave Tables for Replication Information



Problem: Slave thread commits and then updates slave data.

what if it crashes in-between?

- recovery issues - where to restart replication from?
- file corruption vulnerability
- administration hassle

Solution: store data in (transactional) tables:

- slave's data is updated transactionally
- engine agnostic
- crash-safeness (requires transactional tables) frees the DBA to manually recover transactions after a crash



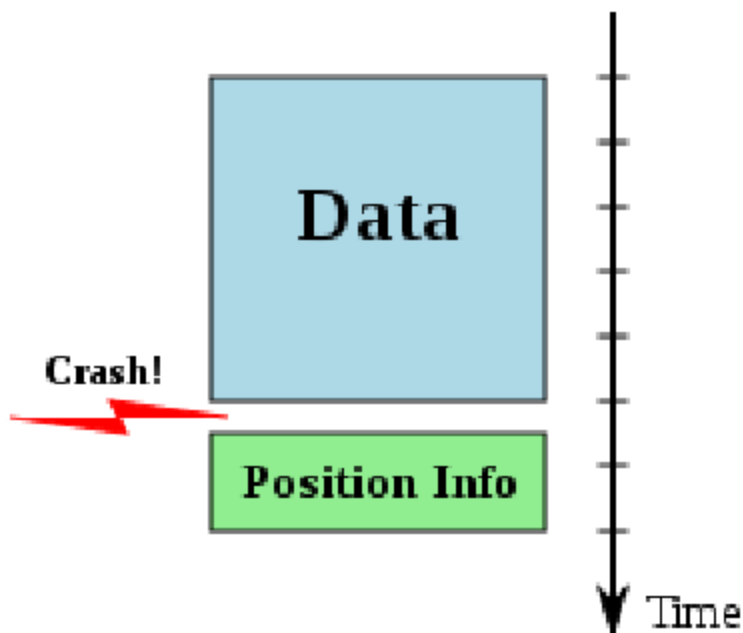
MySQL 5.6 Replication Features

Slave Tables for Replication Information

MySQL <5.6

Transaction Data: in tables

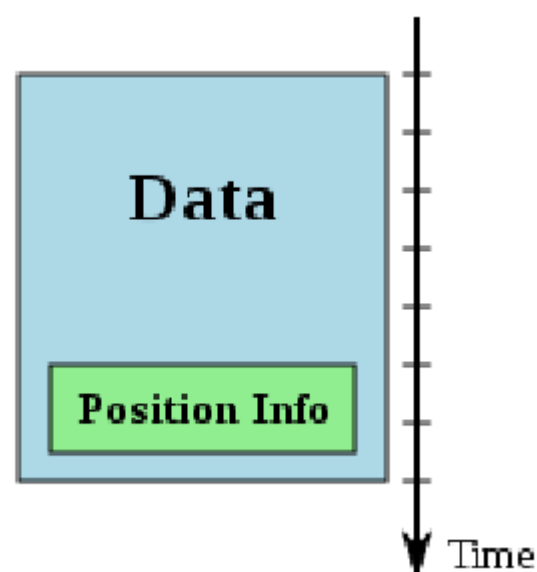
Replication Info: in files



MySQL 5.6

Transaction Data: in tables

Replication Info: in tables



ORACLE

Slave Tables for Replication Information

- System tables:

- slave_master_info (mysql.slave_master_info)

- master-info-repository=TABLE

- slave_relay_log_info (mysql.slave_relay_log_info)

- relay-log-info-repository=TABLE

```
mysql_slave> stop slave;
```

```
mysql_slave> SET GLOBAL master_info_repository = 'TABLE';
```

```
mysql_slave> SET GLOBAL relay_log_info_repository = 'TABLE';
```

```
mysql_slave> start slave;
```

Make sure you add to my.cnf

- master-info-repository =TABLE

- relay-log-info-repository =TABLE

- Transactional tables enables transactional slave positions

- Automatic conversion between files and tables on startup

- Long time awaited feature



<http://dev.mysql.com/doc/refman/5.6/en/replication-options-binary-log.html>

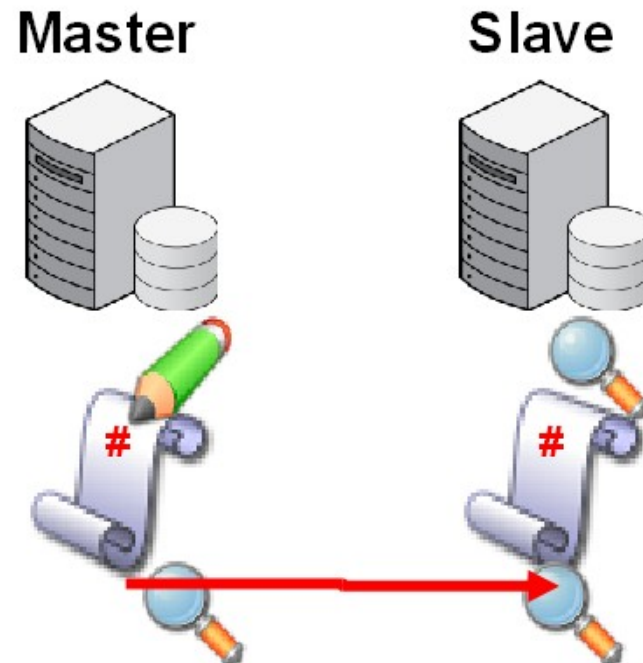
Slave Tables for Replication Information

```
mysql_slave> select * from slave_master_info \G
***** 1. row *****
      Master_id: 2
    Number_of_lines: 22
    Master_log_name: yoda-bin.000003
    Master_log_pos: 323
           Host: yoda
      User_name: replication
    User_password: slavepass
           Port: 3306
    Connect_retry: 10
    Enabled_ssl: 0
           Ssl_ca:
      Ssl_capath:
           Ssl_cert:
      Ssl_cipher:
           Ssl_key:
    Ssl_verify_server_cert: 0
           Heartbeat: 1800
           Bind:
    Ignored_server_ids: 0
           Uuid: 75d407df-2be4-11e1-9668-b4be9bce39b0
      Retry_count: 86400
           Ssl_crl:
      Ssl_crlpath:
1 row in set (0.00 sec)
```



Replication Event Checksums

- Detects corrupt replication events before they are applied
- Guards against bugs and disk or network corruptions
- CRC-32 checksum, more precisely ISO-3309 (supplied with zlib)
- New mysqld options:
 - binlog-checksum= NONE or CRC32
generated by the session thread and written to the binary log
 - SET GLOBAL binlog_checksum = 1;
 - master-verify-checksum= 0 or 1
Master validates checksum read from the binary log
 - SET GLOBAL master_verify_checksum = 1;
 - slave-sql-verify-checksum= 0 or 1
SQL thread should verify the checksum when reading it from the relay log on the slave
 - mysql> SET GLOBAL slave_sql_verify_checksum=1;



<http://mysqlmusings.blogspot.com/2011/04/replication-event-checksum.html>

Replication Event Checksums

Problem: events get corrupted while en route from master to slave.

Why and where it got corrupted?

Disk corruption, network, bugs in replication, faulty memory, cosmic ray, act of God?

Solution: include in each event its control checksum and verify it before:

sending it to the slave (master - dump thread)

storing it in the relay log (slave - IO thread)

applying it (slave - SQL thread)

several verification points: flexibility

```
mysql> show global variables like '%checksum%';
```

```
+-----+-----+
| Variable_name      | Value |
+-----+-----+
| binlog_checksum    | CRC32 |
| innodb_checksum_algorithm | innodb |
| innodb_checksums   | ON    |
| master_verify_checksum | ON    |
| slave_sql_verify_checksum | ON    |
+-----+-----+
```



Multi-Threaded Slave

Throughput of slave increased by allowing multiple slave threads:

- 0 - functionality disabled
- 0 to 1024

Exec_Master_Log_Posn in SHOW SLAVE STATUS represents a “low-water” mark, before which no uncommitted transactions remain.

Configure using:

- slave-parallel-workers=4

On a per-database basis

- can process successive transactions on a given database without waiting for updates on other databases to complete



http://dev.mysql.com/doc/refman/5.6/en/replication-options-slave.html#sysvar_slave_parallel_workers

MySQL 5.6 Replication Features

5.6.4 DMR

Multi-Threaded Slave

```
mysql_luke> show slave status\G
```

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

```
....
```

```
Exec_Master_Log_Pos: 114
```

```
mysql> show global variables like '%workers%';
```

Variable_name	Value
slave_parallel_workers	0

```
1 row in set (0.00 sec)
```

```
mysql> SET GLOBAL slave_parallel_workers=4;
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> show global variables like '%workers%';
```

Variable_name	Value
slave_parallel_workers	4

```
1 row in set (0.00 sec)
```



http://dev.mysql.com/doc/refman/5.6/en/replication-options-slave.html#sysvar_slave_parallel_workers

ORACLE

Time Delayed Replication

Problem: Make replication slave to lag a specified amount of time behind the master to:

- To protect against user mistakes on the master.
- To test how the system behaves when there is a lag.
- To inspect what the database looked like long ago, without having to reload a backup.

Solution: The slave waits until a given number of seconds elapses before applying the changes:

- Delays configured per slave: flexible deployment;
- Are implemented in the SQL thread layer.
- Rolling Database Backups with Relayed Replication



Time Delayed Replication

User interface:

- **CHANGE MASTER TO MASTER_DELAY = <NUM_SECONDS>;**
 - mysql> stop slave;
 - mysql> CHANGE MASTER TO MASTER_DELAY=86400; start slave;
- **SHOW SLAVE STATUS:**
 - SQL_Delay: 86400
 - SQL_Remaining_Delay: 86395
 - Slave_SQL_Running_State: Waiting until MASTER_DELAY seconds after master executed event
- **RESET SLAVE** clears the configured delay;

Rolling forward delayed slaves until bad event:

```
START SLAVE [SQL_THREAD] UNTIL MASTER_LOG_FILE = 'log_name',  
MASTER_LOG_POS = log_pos
```



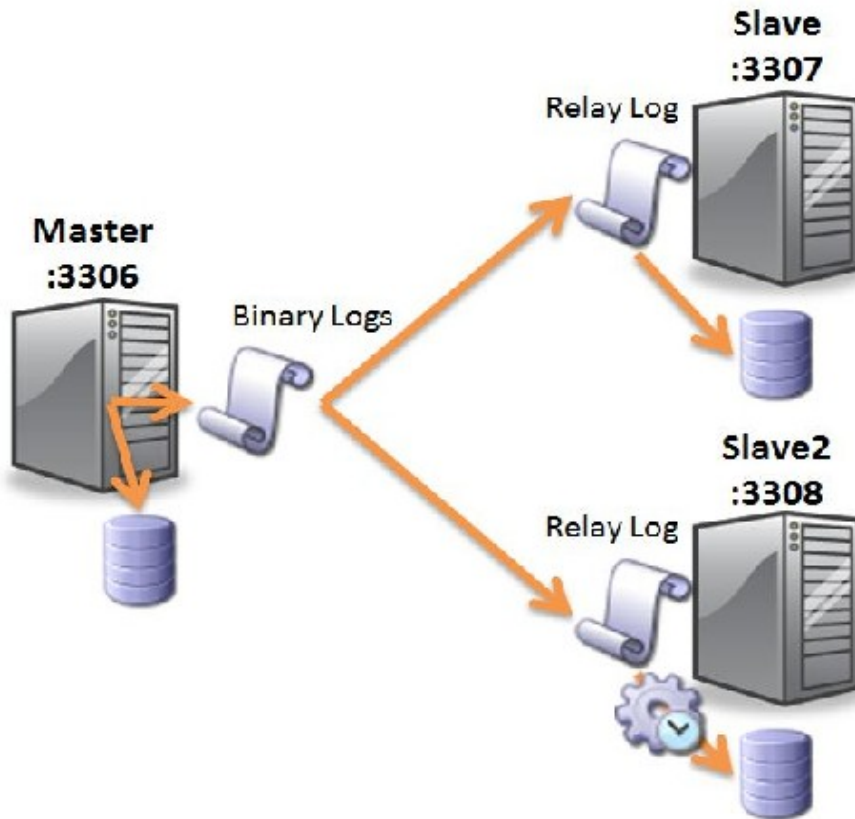
<http://dev.mysql.com/doc/refman/5.6/en/replication-delayed.html>

<http://dev.mysql.com/doc/refman/5.6/en/start-slave.html>

MySQL 5.6 Replication Features

5.6.4 DMR

Time Delayed Replication



```
slave2> CHANGE MASTER TO  
-> MASTER_HOST = 'localhost',  
-> MASTER_PORT = 3306,  
-> MASTER_USER = 'repl_user',  
-> MASTER_PASSWORD = 'pw',  
-> MASTER_DELAY = 86400;
```

```
slave2> START SLAVE;
```

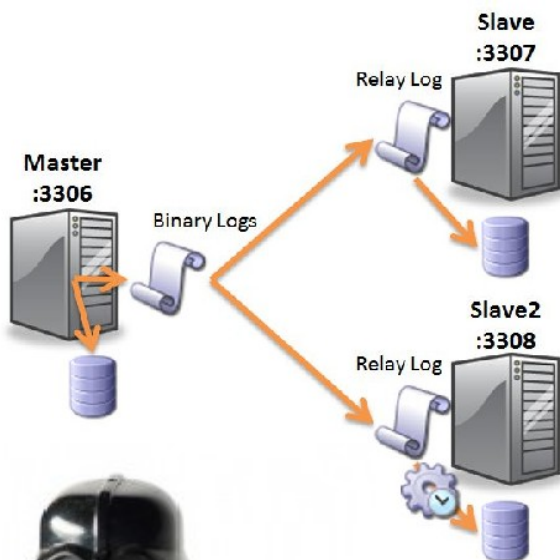


ORACLE

MySQL 5.6 Replication Features

5.6.4 DMR

Time Delayed Replication



```
mysql_luke> STOP SLAVE;
```

```
mysql_luke> SHOW RELAYLOG EVENTS FROM 2337\G
***** 4. row *****
```

Log_name: luke-relay-bin.000005

Pos: 2674

Event_type: Query

Server_id: 1

End_log_pos: 2623

Info: **drop database towns**

```
mysql_luke> START SLAVE UNTIL
```

```
-> MASTER_LOG_FILE='luke-relay-bin.000005',
```

```
-> MASTER_LOG_POS=2674;
```



ORACLE

Optimized Row Based Replication

In MySQL row-based replication (RBR), each row change event contains two images, a “before” image whose columns are matched against when searching for the row to be updated, and an “after” image containing the changes.

- can often save disk, memory, and network usage by logging only those columns which are actually required.
- Default is full : Log all columns in both the before image and the after image.
- New option: `binlog-row-image= minimal`
- no effect when the binary logging format is STATEMENT. When `binlog_format` is MIXED, the setting for `binlog_row_image` is applied to changes that are logged using row-based format, but this setting no effect on changes logged as statements.

```
mysql> show global variables like '%binlog_row_image%';  
mysql> SET GLOBAL binlog_row_image=minimal;
```



<http://d2-systems.blogspot.com/2011/04/mysql-562-dm-optimized-row-based.html>

http://dev.mysql.com/doc/refman/5.6/en/replication-options-binary-log.html#sysvar_binlog_row_image

ORACLE

Optimized Row Based Replication

Problem: all columns are logged for a row, even if only some are requested or changed (this is valid for Before and After Images - BI/AI):

network bandwidth waste, increased memory footprint, disk space overuse...

Solution: Make the server to dynamically choose which columns to log for DELETE, UPDATE and INSERT row events:

Minimal - PK or UK for BI and changed columns for AI

Noblob - no blobs columns when not needed

Full - all columns always



Informational Log Events

Problem: no way to send informational events down the replication stream.

Solution: Create a class of events that carry information from master to slave(s):

- Use case: log the query that originated several rows events up-front as an informational event;
- Feature often requested for debugging.



http://dev.mysql.com/doc/refman/5.6/en/replication-options-binary-log.html#option_mysql_binlog-rows-query-log-events

ORACLE

Informational Log Events

Enhances auditing and debugging when using Row-Based Replication by writing the original query to the binary log, which is then replicated with its associated row-based event to the slave.

write informational log events such as row query log events into its binary log. `sysvar_binlog_rows_query_log_events` must be disabled during logging.

Logs the query that originated the subsequent rows changes.

Shows up in `mysqlbinlog` and `SHOW SLAVE STATUS` output.

New variable:

`--binlog-rows-query-log-events= ON|OFF` (default: OFF)

```
mysql> SET GLOBAL binlog_rows_query_log_events=ON;
```

```
mysql> show global variables like '%binlog_rows_query_log_events%';
```

```
+-----+-----+
| Variable_name      | Value |
+-----+-----+
| binlog_rows_query_log_events | ON   |
+-----+-----+
```



Remote Backup of Binary logs

Problem: There is no way to create real-time backups of the master's binary logs.

Solution: Make use of mysqlbinlog facilities that retrieve and dump remote MySQL log contents as SQL statements to make it output in raw format:

- DBAs don't need to do remote logins to retrieve master's binlogs or setup an intermediate slave.



Remote Backup of Binary logs

Make use of mysqlbinlog facilities that retrieve and dump remote MySQL log contents. Writes to a local file with the same name as the original.

DBAs don't need to do remote logins to retrieve master's binlogs or setup an intermediate slave.

Relevant new options for mysqlbinlog:

- raw: dump in raw format
- stop-never: waits for new data upon reaching the end of the log
- stop-never-slave-server-id: id that mysqlbinlog will use to emulate itself as a slave.

Raw format use cases:

- make a static backup
- backing up a set of log files and stopping when the end of the last file is reached
- continuous (“live”) backup

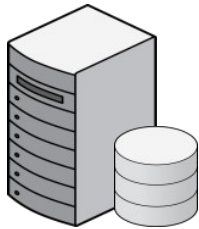
```
$> mysqlbinlog --read-from-remote-server --raw -h secret_server -P 3306 -u root mysql-bin.000001
```



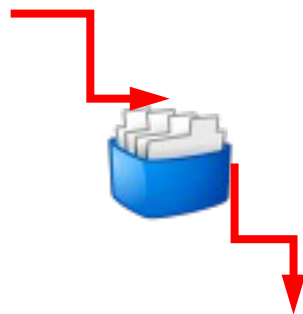
<http://dev.mysql.com/doc/refman/5.6/en/mysqlbinlog-backup.html>

MySQL 5.6 Replication Features

Global Transaction IDs



GTID=123456



{my-bin.000101,873}



- A logical identifier associated with each transaction instead of a physical one (filename + offset) it pin-points a set of changes resulting from the execution of a transaction.
- Each transaction identified by logical id rather than physical (file + offset)
 - Same for all servers
 - Contained in binary log
 - Index to map global ID to local physical position
 - Probably not in the initial version
 - Doesn't change during failover
 - Simpler slave promotion & maintenance of complex replication topologies

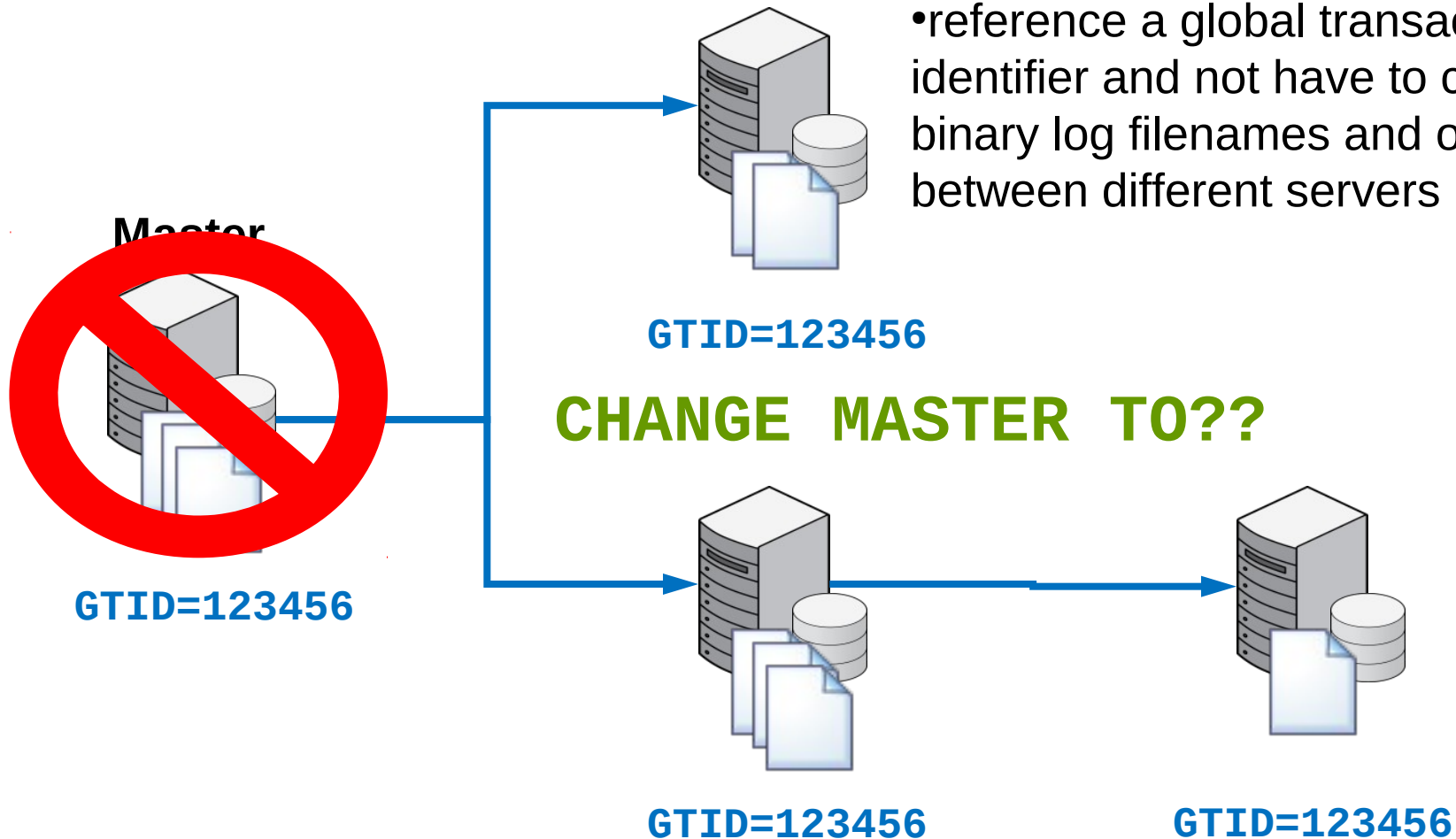


MySQL 5.6 Replication Features

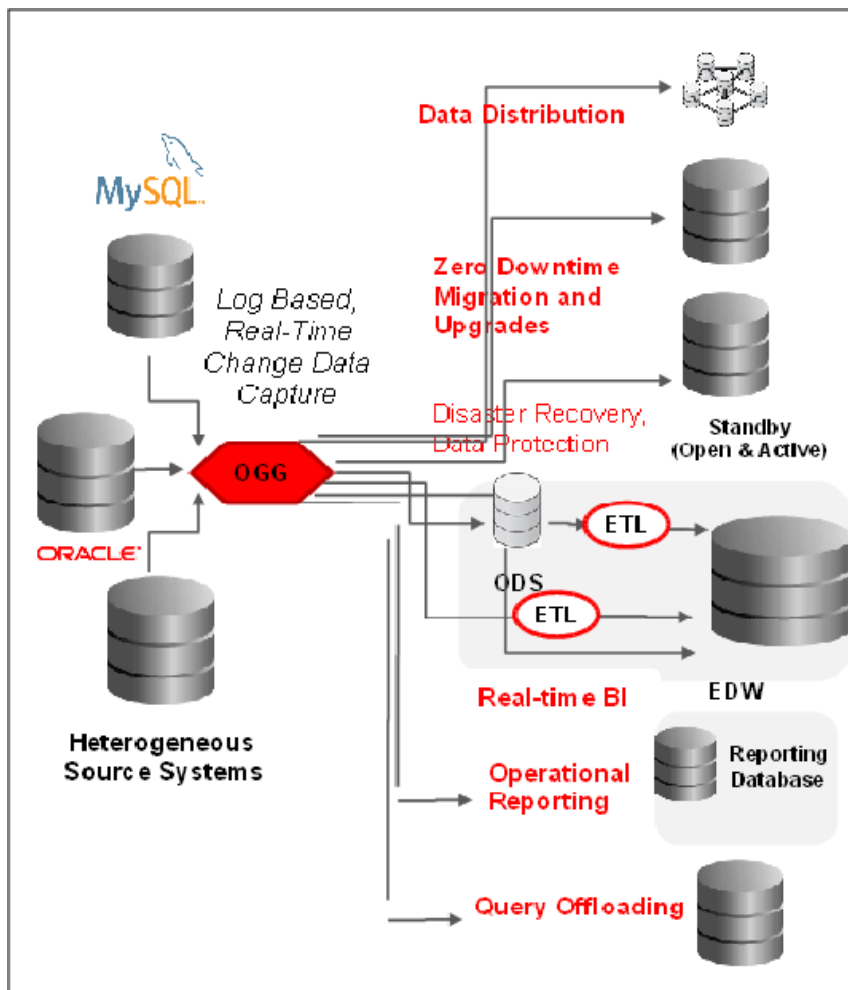
Global Transaction IDs

Slave(s) can start from same GTID

- reference a global transaction identifier and not have to convert binary log filenames and offsets between different servers



MySQL Replication



Oracle Integrations: Golden Gate

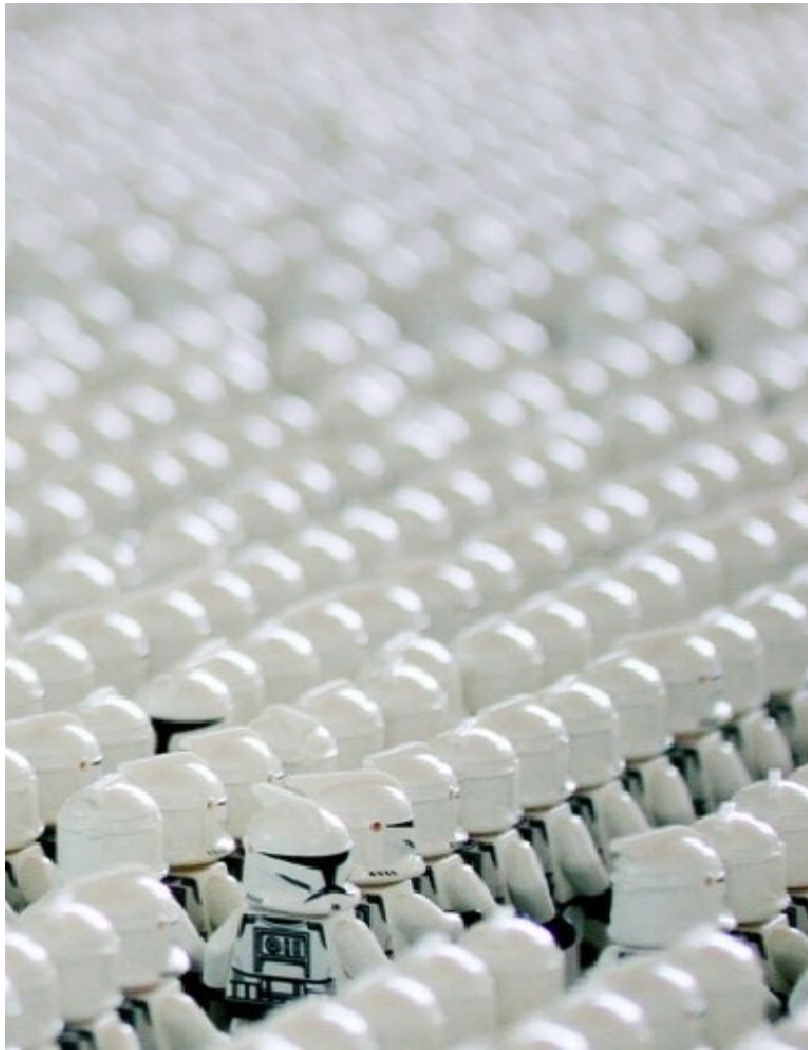
- Heterogeneous Replication between MySQL, Oracle
- MySQL specific optimizations
- Hybrid web, enterprise applications (Sabre Holdings)
- Offload, scale query activity to MySQL read-only slaves
- Real-time access to web-based analytics, reporting
- Migration path from/to MySQL from other databases with minimal downtime

ORACLE® **11g**
FUSION MIDDLEWARE
GOLDENGATE



ORACLE®

MySQL Enterprise Monitor



All the slaves paying attention?

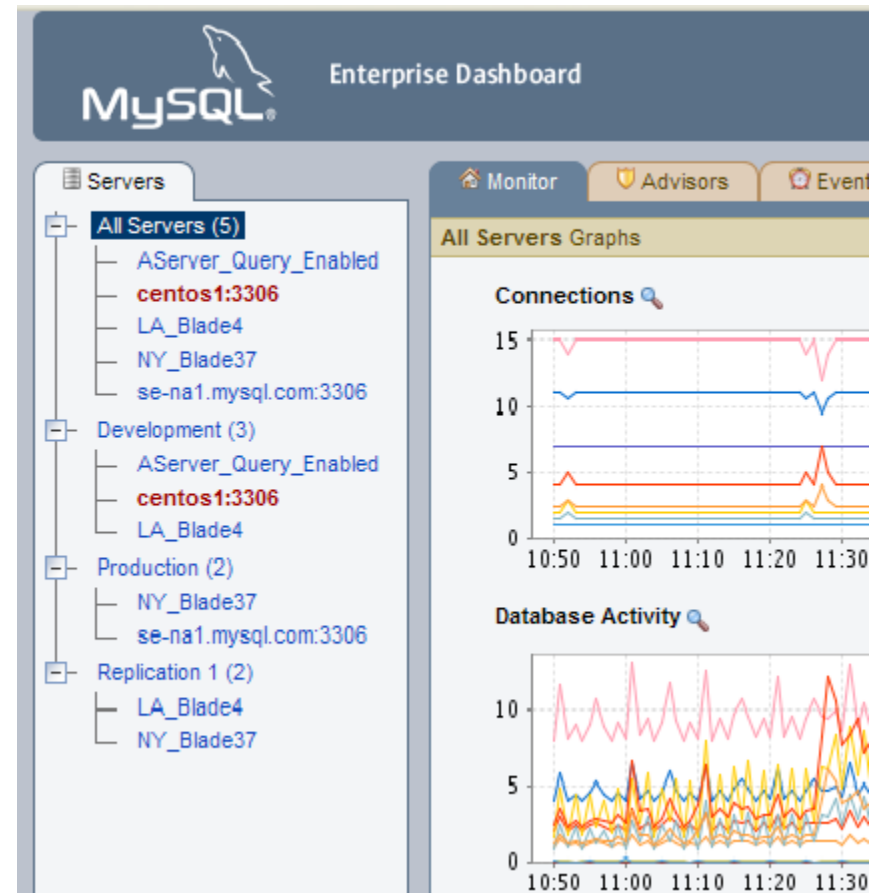
- Do you know when a slave is behind?
- What type of topology do you have?
- Alerted on Errors?
- Slow Queries?



ORACLE®

MySQL Enterprise Monitor

- Global view of MySQL environment
- Automated, rules-based monitoring and alerts (SMTP, SNMP enabled)
- Query capture, monitoring, analysis and tuning, correlated with Monitor graphs
- Visual monitoring of “hot” applications and servers
- Real-time Replication Monitor with auto-discovery of master-slave topologies
- Integrated with MySQL Support



ORACLE®

MySQL Enterprise Monitor

MySQL Replication Monitoring

MySQL Enterprise Monitor - Replication Monitoring

Servers	Type	Slave ID	Slave SQL	Seconds Behind	Binlog	Binlog Pos	Master Binlog	Master Binlog Pos	Last Error
Basic (2)	TRF	Running	Running						rename group
master: 0101	master				MysqldResource-bin.000002	1274			
slave 10100	slave	Running	Running	000:00			MysqldResource-bin.000002	1274	
Ringset (2)	RING	Running	Running						rename group
Yang: 0120	master/slave	Running	Running	000:00	MysqldResource-bin.000002	272	MysqldResource-bin.000002	443	
Yin 1012	master/slave	Running	Running	000:00	MysqldResource-bin.000002	446	MysqldResource-bin.000002	272	
RingSpoke (4)	MXED	Running	Running						rename group
ring: 10133	master/slave	Running	Running	000:00	MysqldResource-bin.000002	446	MysqldResource-bin.000002	272	
ring2 10132	master/slave	Running	Running	000:00	MysqldResource-bin.000002	272	MysqldResource-bin.000002	443	
ring3 10131	master/slave	Running	Running	000:00	MysqldResource-bin.000002	272	MysqldResource-bin.000002	272	
ring3 slave: 0180	slave	Running	Running	000:00			MysqldResource-bin.000002	272	
Tree 3 (5)	TREE	Running	Running						rename group
master: 0153	master				MysqldResource-bin.000002	272			
slave1: 0150	slave	Running	Running	000:00			MysqldResource-bin.000002	272	
slave2 master: 10152	master/slave	Running	Running	000:00	MysqldResource-bin.000002	272	MysqldResource-bin.000002	272	
slave2 slave 10151	slave	Running	Running	000:00			MysqldResource-bin.000002	272	
slave2: 0154	slave	Running	Running	000:00			MysqldResource-bin.000002	272	

MySQL Enterprise © 2009-2010 MySQL AB. All rights reserved. Enterprise Software | Update Service | Knowledge Base | Technical Support | About

Logged in as 'admin' (Oct 7, 2018 11:55 AM)
Monitoring 5 instances on 2 hosts (98 hosts remaining). Subscription is up-to-date. [More info...](#)



ORACLE

MySQL Replication Common Questions



Do u need two different mysql servers to be installed ?

- Yes - replication happens between MySQL servers

How do you know if replication is behind ?

- "SHOW SLAVE STATUS" on the slave and check the Seconds_Behind_Master value



ORACLE®

MySQL Replication Common Questions



Does semi-synchronous replication work on windows platform ?

- Yes, replication is not OS dependent.

Can you promote a slave to a master in the event that the master server has a hardware failure?

Can you also switch the master back to a slave at a later time?

- Absolutely, covered in a white paper...
- <http://www.mysql.com/why-mysql/white-papers/mysql-wp-replication.php>



ORACLE®

MySQL Replication Common Questions

If you are using circular replication and you have an `auto_increment` column, how can you make sure two inserts at the same time will not get the same id on two different servers?



- When using multi-master replication together with auto-increment columns, you should use the `auto_increment_offset` and `auto_increment_increment` parameters on each server to make sure that there are no duplicate values assigned. They should use the same `auto_increment_increment` value but different `auto_increment_offset` values to make sure they don't clash



MySQL User Groups

Asia

Mongolia
Mongolia User group

India

Bangalore
Delhi

Indonesia

MySQL Indonesia

Japan

MyNA(MySQL Nippon Association)

Malaysia

The Malaysian MySQL User Group (Kuala Lumpur)

Russian Federation

Moscow

Singapore

Singapore

Taiwan

Taiwan MySQL User Group

Europe

Finland

Helsinki

France

Le MUG.fr

Ireland

Dublin

Italia

Italia

Lithuania

Kaunas

Switzerland

Sauvons MySQL

United Kingdom

Birmingham
London and South East

Middle East

Israel

Israel MySQL User Group

United Arab Emirates

Dubai

Turkey

Turkey

North America

Canada

Calgary, AB
Montreal, QC
Toronto, ON

USA

Alabama
Huntsville

California

Los Angeles
San Francisco
Silicon Valley

Colorado

Denver LAMP Meetup
Colorado Users Group

Florida

Miami

Georgia

Atlanta

Illinois

Chicago
Massachusetts
Boston

Minnesota

Minneapolis

New Jersey

Northern New Jersey

New York

New York

North Carolina

Raleigh
Ohio
Cleveland

Pennsylvania

Philadelphia

Tennessee

Memphis

Texas

Austin
Dallas/Fort Worth
Houston
San Antonio

Washington

Seattle:

South America

Brasil

MySQL Brasil

Oceania

Australia

Sydney

Africa

South Africa

Pretoria



<https://wikis.oracle.com/display/mysql/List+of+MySQL+User+Groups>

MySQL Available Now

Development Releases:

- <http://dev.mysql.com/downloads/mysql/#downloads>

Evaluate the new features

- Questions & Feedback: forums.mysql.com/list.php?26
- Bugs: bugs.mysql.com/

Replication in MySQL 5.5

- mysql.com/why-mysql/white-papers/mysql-wp-replication.php

Replication 5.6 documentation

- dev.mysql.com/doc/refman/5.6/en/replication.html

Planet

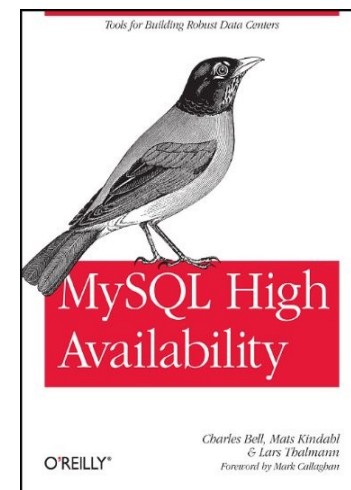
- Planet.mysql.com

MySQL High Availability

- <http://shop.oreilly.com/product/9780596807290.do>

Learn more about HA Solutions for MySQL

- mysql.com/why-mysql/white-papers/mysql_wp_ha_strategy_guide.php



ORACLE®

MySQL Embedded Online Forum

Tuesday, January 31, 2012 9:00AM PST - 12:00 PM PST

<http://bit.ly/w1bOwX>



Think You Know MySQL? Get the Facts.

MySQL is even better than you think. According to the experts, it's easier to use and it's more capable than most ISVs / OEMs realize, and by taking a few tried-and-tested steps before shipping your MySQL-embedded products, you can lower your product's costs and increase its performance.

Join our expert-created and delivered MySQL Embedded Online Forum to learn:

- Why MySQL is a great embedded database for startups as well as the largest software, hardware, and appliance vendors in the world, and how its features ensure costs remain low throughout an application's life cycle.
- MySQL installation options that require minimal or zero end user effort and how to easily build them into your application
- How to secure MySQL embedded in applications, appliances, and devices
- Tips to simplify your integration with MySQL using the most popular MySQL Connectors and guidance on selecting the settings that will maximize your application's performance
- Why MySQL delivers excellent performance and how to go beyond the default settings to optimize MySQL's queries, reporting, and search capabilities

Agenda:

It Just Works! The Beauty of MySQL as an Embedded Database

Building MySQL Embedded for Simple Installation and Security

Craig Sylvester , Principal MySQL Sales Engineer, Oracle

Getting the Most Out of MySQL Connectors for Better Performance and Streamlined Integration

Matt Lord, Senior Principal MySQL Support Engineer, Oracle

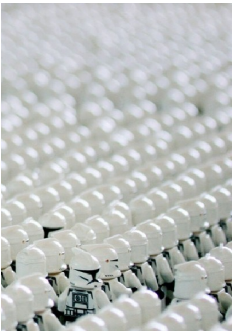
Better than Great: MySQL Embedded Performance and Reporting

Alexander Rubin, MySQL Technical Consulting Manager, Oracle



ORACLE®

Credits



<http://www.starwars.com/>

<http://starwars.lego.com/en-us/Default.aspx>

<http://thewondrous.com/wp-content/uploads/2010/04/Largest-display-of-Star-Wars-clone-troopers-built-with-interlocking-plastic-bricks.jpg>

<http://www.betabeat.com/2011/09/02/clone-wars-rise-of-the-fast-follower-startups/>

<http://www.minifigures.co.uk/lego/star-wars/master-yoda/>

<http://www.minifigures.co.uk/lego/star-wars/luke-skywalker-on-tatooine/>

<http://www.minifigures.co.uk/lego/star-wars/darth-vader-minifig/>

http://farm6.static.flickr.com/5024/5554971120_df447dd31c.jpg

<http://www.stormgrounds.com/wallpaper/Miscellaneous/Storm-Trooper-Legos>

http://www.jorymon.com/images/2009/june/stormtrooper_1.jpg

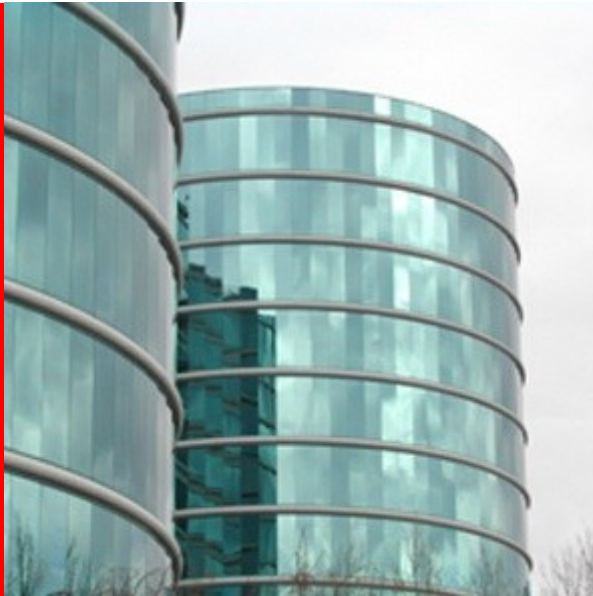
http://203.148.253.29/wp-content/photos/KIS_Levi/

<http://highdefinitionwallpapers.com/hd-wallpapers/funny-wallpapers/lego-storm-trooper-keyboard-funny-wallpaper/>

<http://www.stormgrounds.com/wallpaper/Entertainment/Srorm-Trooper>



ORACLE®



ORACLE®



Thanks for attending!

keith.larson@oracle.com

sqlhjalp.blogspot.com

sqlhjalp.com/pdf/2012_Scale_Replication.pdf



ORACLE®