Managing MySQL Replication

MySQL User Conference

Jeremy Zawodny
Yahoo!

April 10th, 2003
San Jose, California

http://jeremy.zawodny.com/mysql/
About Me

• Engineer in Y! Search (prev. Y! Finance)
• MySQL user for over 5 years
• Active in MySQL community
• Write about LAMP for Linux Magazine
• MySQL advocacy & support at Yahoo!

Home: Jeremy@Zawodny.com
Work: jzawodn@yahoo-inc.com
http://jeremy.zawodny.com/mysql/
MySQL at Yahoo!

- Roughly 200-400 servers world-wide
- FreeBSD and Linux
- Commodity hardware
- Replaces home-grown “database” systems
- Replaces Oracle in a few cases
- Typical install uses between 1-20GB
- Used both “live” and in batch processing
- Replication and load-balancing
Talk Goals

• Why to use replication
• Learn how replication works in 3.23 and 4.0
• Understand how to configure it
• Know what can go wrong
• Learn about helpful tools
• Decide which topologies make sense
• Look at load-balancing
• Discuss what’s missing
• Finish on-time (or before!)
Outline

• Goals
• Required Knowledge
• Quick Survey
• Replication Basics
• Common Problems
• Tools
• Topologies
• What’s missing
• Questions and Answers
Required Knowledge

• Basic MySQL administration
• Query types
• Networking concepts
Quick Survey

• What version of MySQL are you using?
• Are you using replication?
  – How many slaves?
  – More than one master?
  – Did it work on the first try?
• Which operating systems?
• Familiarity with other RDBMS servers?
• Role? DBA? Developer? SysAdmin?
• MySQL dedicated or shared servers?
Replication is

- Streaming of queries from a master to one or more slaves
- Very light-weight on the master
- Serial(ized)
- Relatively fast
- Easy to configure
- Easy to break
Replication is NOT

• The best solution to every problem
• Real-time
• Guaranteed
• Synchronous
• Perfect
• Rocket science
Why Replication?

- Hot spare
- Make backups easier
- Geography: put data closer to users
- Test Environment
- Insulate the main server from (ab)users
- Scaling: load-balancing
- Just to be cool! 😊
Replication Basics

• Master / Slave replication
• Master records all write queries in the binary log
• Slaves read the binary log from the master and run the queries locally
• A master can have many slaves
• A slave can have only one master
• A server can be both a master and a slave
  – A relay or “local master”
Replication Basics contd.

- Masters are mostly unaware of their slaves
  - They don’t know what the slave’s state is
  - In 4.0, there’s a new privilege called REPLICATION SLAVE

- Masters and Slaves can selectively filter queries
  - Database level
  - Table level

- Replication is asynchronous
Replication in 3.23.xx

• Slave connects to master
• Reads a query
• Executes the query
• Compares results to make sure it’s consistent
• Reads the next query and repeat

• Problems
  – Slow queries
  – High latency networks
Replication in 4.0

- Slave operation is multi-threaded
- Relay thread
  - Connects to master and copies queries to local spool (relay log) without delay
- SQL thread
  - Acts like 3.23.xx replication, but uses the local relay log rather than connecting to the master
- Faster than 3.23.xx
- Less chance of lost queries
- Transactions replicate properly (?)
Replication Setup

• Configure replication account on the master
• Enable binary log on the master
  – my.cnf file
• Snapshot master and reset log
• Install snapshot on the slave
• Setup replication options on slave
  – my.cnf file
• Restart the slave
• Check the error log
• http://www.mysql.com/doc/R/e/Replication.html
Snapshot Options

• Take master off-line
  – Use tar/zip/etc to grab all the data
  – Include InnoDB/BDB data files/logs
  – Very fast, no contention problems

• Keep master on-line
  – Use mysqlsnapshot to grab the data
  – Tables will be read-locked
  – Relatively fast
Snapshot Options Contd.

- Keep master on-line
  - Use `LOAD DATA FROM MASTER`
  - The security tables don’t replicate (bug!)
  - Slowest, because indexes are re-built
  - Best for small data sizes
- Remember to reset the master’s binlog
  - `RESET MASTER` command
- `mysqldump`
  - Use for small data
Snapshot Future

• Someday snapshots won’t be necessary
• MySQL will handle it automatically
• Point a new slave at the master and let it go
Replication Settings

- **Master**
  - `server-id = #`
  - `log-bin = filename`
  - `log-bin-index = filename`
  - `log-slave-updates`
  - `binlog-do-db = dbname`
  - `binlog-ignore-db = dbname`
  - `set-variable = max_binlog_size = size`
Replication Settings

• Slave
  – (all master options apply)
  - `server-id = #`
  - `master-host = hostname`
  - `master-user = username`
  - `master-password = password`
  - `master-port = #`
  - `master-info-file = filename`
  - `master-connect-retry`
Replication Settings

• Slave (continued)
  - `replicate-do-table = dbname.table`
  - `replicate-ignore-table = dbname.table`
  - `replicate-wild-do-table = dbname.table`
  - `replicate-wild-ignore-table = dbname.table`
  - `replicate-ignore-db = dbname`
  - `replicate-do-db = dbname`
  - `log-slave-updates`
  - `replicate-rewrite-db = old_db->new_db`
  - `slave-skip-errors = error_list`
  - `skip-slave-start`
  - `slave_net_timeout = #`

• SSL options on the slave
  - Not yet used (4.1 or 5.0)
Replication Commands

- SLAVE START
  - SLAVE START SQL_THREAD
  - SLAVE START RELAY_THREAD

- SLAVE STOP
  - SLAVE STOP SQL_THREAD
  - SLAVE STOP RELAY_THREAD

- 3.23
  - SLAVE START
  - SLAVE STOP
Replication Commands

- RESET MASTER (FLUSH MASTER)
- RESET SLAVE (FLUSH SLAVE)
- LOAD TABLE ... FROM MASTER
- LOAD DATA FROM MASTER
- CHANGE MASTER TO ...
- SHOW MASTER STATUS
- SHOW SLAVE STATUS
- SHOW SLAVE HOSTS
- PURGE MASTER LOGS TO 'logname'
Common Problems

• Auto-increment fields
• User variables
• Improper sync of a new slave
• Cross database queries and slave filtering
• Binary (or relay) logs use all your disk space!
• Version mismatch
• LOAD TABLE ... FROM MASTER
  – Not for InnoDB
Troubleshooting

• When the slave stops/dies/etc.
  – Check the MySQL error log
  – Run `SHOW SLAVE STATUS`
  – Notice the query that is stuck
  – Maybe skip the current query and move on?
    • `SET GLOBAL SQL_SLAVE_SKIP_COUNTER = 1`
  – Did all the slaves have the same problem, or just this one?
### Slave Status

```
mysql> SHOW SLAVE STATUS \G
*************************** 1. row ***************************
    Master_Host: mysql.yahoo.com
    Master_User: foo
    Master_Port: 3306
    Connect_retry: 15
    Master_Log_File: binary-log.145
    Read_Master_Log_Pos: 908423089
    Relay_Log_File: relay-log.145
    Relay_Log_Pos: 127617325
    Relay_Master_Log_File: binary-log.145
    Slave_IO_Running: Yes
    Slave_SQL_Running: Yes
    Replicate_do_db:
    Replicate_ignore_db:
    Skip_counter: 0
    Exec_master_log_pos: 908423089
    Relay_log_space: 127636933
1 row in set (0.01 sec)
```
Performance Problems

• Replicate only data you *need* replicated
  – It sounds like common sense
  – Too easy to fall into the “replicate everything” trap

• Consider using a relay server to save bandwidth

• Upgrade from 3.23 to 4.0 to decrease latency

• Put logs on a separate disk
Tools

- **mysqlbinlog**
  - Converts binary log to normal SQL
  - Works on master or slave or relay logs

- **mysqlsnapshot**
  - Creates snapshot for setting up slaves
  - Master stays on-line
  - Handles the tricky parts for you
  - [http://jeremy.zawodny.com/mysql/](http://jeremy.zawodny.com/mysql/)
Monitoring

• Heartbeat for performance
  – Create a table to hold timestamps
  – Add records on the master
  – Read them on the slave
  – Compare and compute the delay

• Errors
  – Watch error log on slave
  – Or run a `SHOW SLAVE STATUS`
Monitoring

- Watch slave status values
- Compare “read” and “exec” positions
- Byte counts are tricky, just like timestamps
Topologies

• Replication is very flexible!
• Master / Slave(s)
• Dual-Master
• Multi-Master “Ring”
• Master / Relay / Slave
• Mix and Match
Master/Slave Replication

Write queries, read queries

Server
Master

Binary log

Read queries only

Server
Slave

Copyright 2003, Jeremy Zawodny
Dual-Master Replication

Write queries, read queries

Server
Master

Write queries, read queries

Server
Master

Binary logs
Relay

Master

High latency or low bandwidth link

Slave

Slave

Copyright 2003, Jeremy Zawodny
Health Checks

• How can the load-balancer know which slaves are healthy?
  – Number of connections (or free slots)
  – Performance
  – Not behind on replication

• Often, the decision is application specific

• Load-balancers aren’t that smart

• Build your own checks and let the load-balancer check your checks

Copyright 2003, Jeremy Zawodny
Connection Algorithms

• How does the load-balancer decide who gets the next connection?
  – Best response time
  – Least connections
  – Round robin
  – Random
  – Hash (sticky)
  – Custom?

• All connections are not equal
• What happens under load?
Dedicated vs. Shared

• Shared works well for
  – Small workloads
  – Small amounts of data
  – Simplicity

• Dedicated MySQL servers
  – Specialized hardware not wasted
  – Make tuning easier
  – More room to scale
Good Practices

• Server version diversity
• Platform diversity, if you can afford it
• Daily backup/snapshot
• Never give out real server addresses
• Be careful with selective replication
  – Make sure developers understand it
• Code with replication in mind from day #1
• Monitor, monitor, and monitor
• Server is not the same as “machine”
What’s MySQL Missing

• This is part bug list, part wish list, and a glimpse into the future
• Multiple Replication Logs
  – Decrease network traffic for partial slaves
  – Filter similar to apache with virtual hosts
• Replication Relay or Proxy
• Safe auto-increment handling
• Relay Log Repair
• Multiple binlog with filters
• Replication Relay

Copyright 2003, Jeremy Zawodny
More Missing Stuff

• Efficient LOAD DATA FROM MASTER
• User variable replication
• Management of server-id values
• Mechanism to send specific commands to slaves
  – OPTIMIZE TABLE
  – SET GLOBAL key_buffer_size ...
• Safe auto-increment handling in multi-master setups
Even More Missing Stuff

- SSL for replication
  - Can use SSH or stunnel today
- Absolute log offsets (or a global query counter, sequence number) to make master failover easier
For More Info...

• MySQL mailing lists
  – Visit lists.mysql.com

• Books
  – MySQL Manual
  – MySQL (Paul’s Book)
  – Managing & Using MySQL

• Web searching
Questions and Answers

That's it. I'm done and probably out of time!