Automated MySQL failover with MHA: getting started & moving past its quirks

Colin Charles, Team MariaDB, SkySQL Ab
colin@mariadb.org | byte@bytebot.net
http://mariadb.org/
http://bytebot.net/blog/ | @bytebot on Twitter

Open Source Data Centre Conference, Berlin, Germany - 10 April 2014

whoami

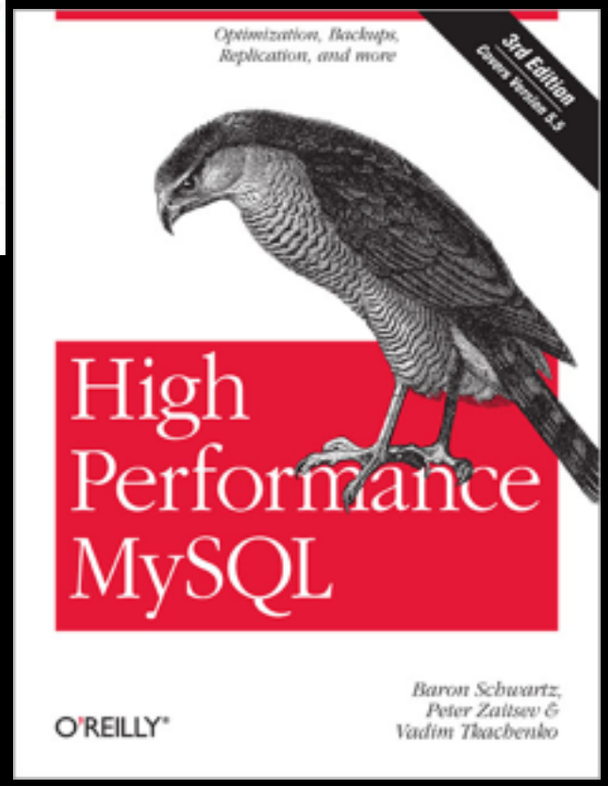
- Work on MariaDB at SkySQL Ab
  - Merged with Monty Program Ab, makers of MariaDB
- Formerly MySQL AB (exit: Sun Microsystems)
- Past lives include Fedora Project (FESCO), OpenOffice.org
- MHA experience
  - since November 2011 (MHA 0.52)
  - NRE work to make it run in a Solaris 10 environment... with no Internet access!
  - Continued deployment advice + work for data centre use
  - Much thanks to SkySQL for the experience

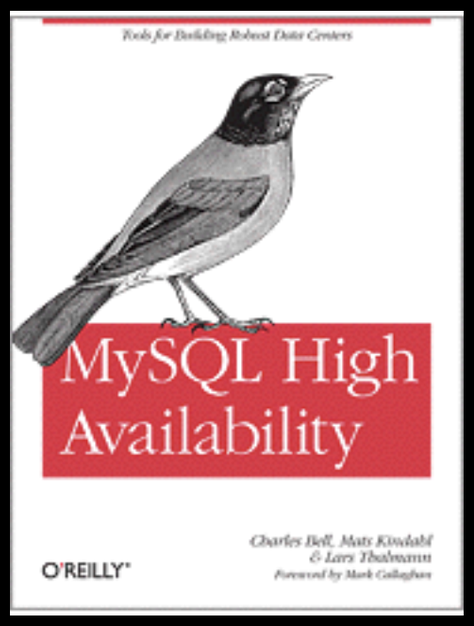
Aims

- Why MHA?
- What does MHA do?
- How does MHA do it?
- Running MHA, VIP failover, integration, etc
- Who uses MHA?
- Is fully automated failover a good idea?

Why this talk

The other tool, which is rather new, is Yoshinori Matsunobu's **MHA** toolkit (http://code.google.com/p/mysql-master-ha/). It is similar to MMM in that it is a set of scripts to build a pseudo-cluster with some of the same general techniques, but it is not a complete replacement; it doesn't attempt to do as many things, and it relies on Pacemaker to move virtual IP addresses. One major difference is that it has a test suite, which should prevent some of the problems MMM has encountered. Other than this, we don't have a strong opinion on the toolkit yet. We haven't talked with anyone other than Yoshinori who is using it in production, and we haven't used it ourselves.





- High Performance MySQL, 3rd Edition
- Published: March 16 2012

But first... MySQL replication

- Single master, multiple slave architecture
  - When master is unavailable, writes stop being accepted
- Promoting a new master is not that easy
  - New master needs to wait to apply all relay log events
  - Slaves need to be consistent
  - MySQL clients need to reconnect to new master
  - All slaves need to start replication from proper binlog position

Where did MHA come from?

- DeNA won 2011 MySQL Community Contributor of the Year (April 2011)
- MHA came in about 3Q/2011
- Written by Yoshinori Matsunobu, Oracle ACE Director

What is MHA?

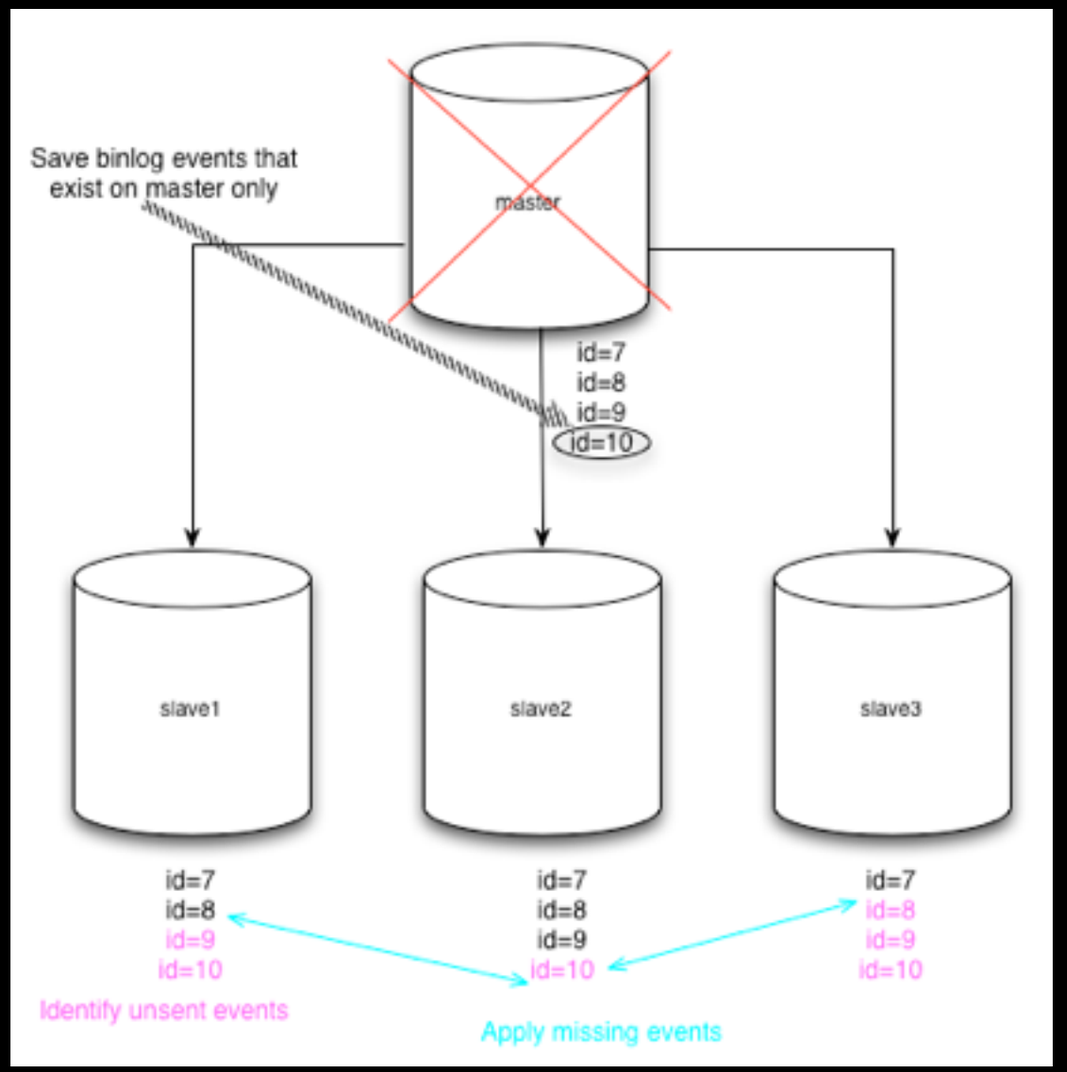
- MHA for MySQL: Master High Availability Manager tools for MySQL
- Goal: automating master failover & slave promotion with minimal downtime
- Set of Perl scripts
- http://code.google.com/p/mysql-master-ha/

Why MHA?

- Automating monitoring of your replication topology for master failover
- Scheduled online master switching to a different host for online maintenance
  - Switch back after OPTIMIZE/ALTER table, software or hardware upgrade
  - Schema changes without stopping services
    - pt-online-schema-change, oak-online-alter-table, Facebook OSC
- Interactive/non-interactive master failover (just for failover, with detection of master failure + VIP takeover to Pacemaker)

Why is master failover hard?

- When master fails, no more writes till failover complete
- MySQL replication is asynchronous (MHA works with async + semi-sync replication)
- slave2 is latest, slave1+3 have missing events, MHA does:
- copy id=10 from master if possible
- apply all missing events



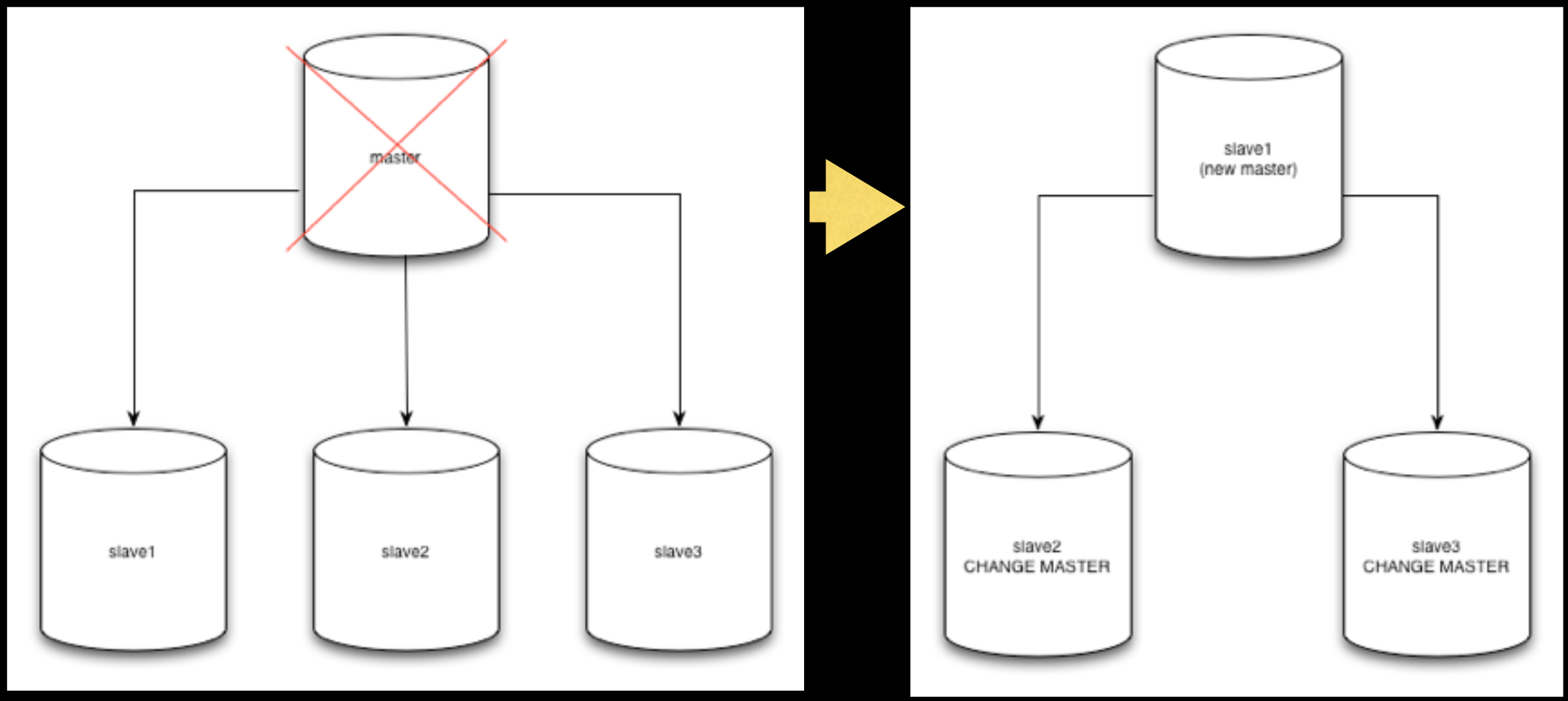
Semi-sync replication

- If master crashes (or is shutdown :P), MHA cannot save binlogs, latest data is lost
- Minimises risk of binlog event existing only on crashed master
- Guarantees at least one slave receives binlog events at commit
- http://code.google.com/p/mysql-master-ha/wiki/UseCases#Using_together_with_Semi-Synchronous_Replication

MHA: Typical scenario

- Monitor replication topology
- If failure detected on master, immediately switch to a candidate master or the most current slave to become new master
  - MHA must fail to connect to master server three times
- CHANGE MASTER for all slaves to new master
- Print (stderr)/email report, stop monitoring

So really, what does MHA do?



Typical timeline

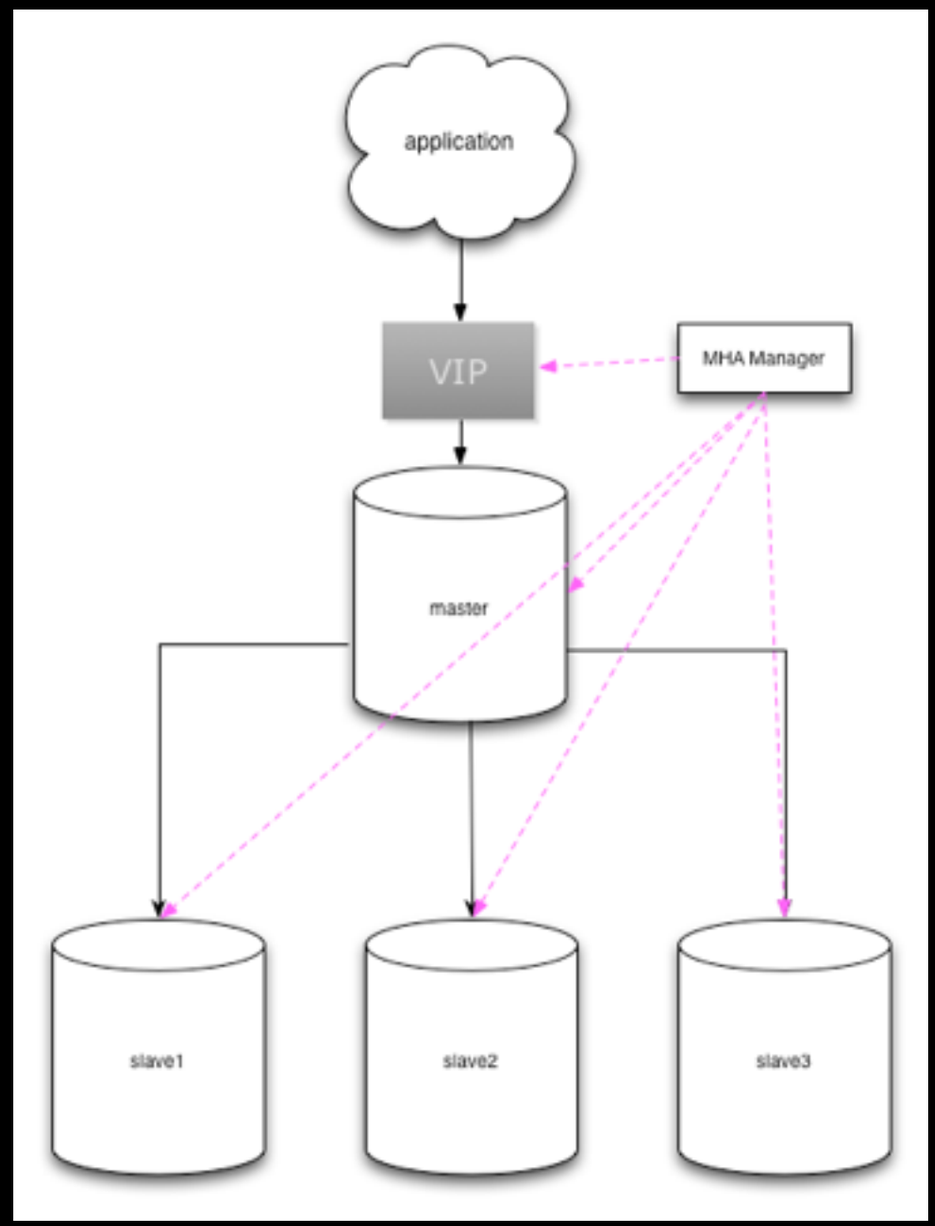
- Usually no more than 10-30 seconds
- 0-10s: Master failover detected in around 10 seconds
  - (optional) check connectivity via secondary network
- (optional) 10-20s: 10 seconds to power off master
- 10-20s: apply differential relay logs to new master
- Practice: 4s @ DeNA, usually less than 30s

How does MHA work?

- Save binlog events from crashed master
- Identify latest slave
- Apply differential relay log to other slaves
- Apply saved binlog events from master
- Promote a slave to new master
- Make other slaves replicate from new master

Getting Started

- MHA requires no changes to your application
- You are of course to write to a virtual IP (VIP) for your master
- MHA does not build replication environments for you - that's DIY



MHA Node

- Download `mha4mysql-node` & install this on all machines: master, slaves, monitor
- Packages (DEB, RPM) available
- Manually, make sure you have `DBD::mysql` & ensure it knows the path of your MySQL

What's in MHA node

- `save_binary_logs` - save & copy master's binlogs
- `apply_diff_relay_logs` - find differential relay log events & apply missing events
- `purge_relay_logs` - purge relay log files

MHA Manager server

- Monitor server doesn't have to be powerful at all, just remain up
- This is a single-point-of-failure so monitor the manager server where MHA Manager gets installed
- If MHA Manager isn't running, your app still runs, but automated failover is now disabled

MHA Manager

- You must install `mha4mysql-node` then `mha4mysql-manager`
- Manager server has many Perl dependencies: `DBD::mysql`, `Config::Tiny`, `Log::Dispatch`, `Parallel::ForkManager`, `Time::HiRes`
- Package management fixes dependencies, else use CPAN

Configuring MHA

- Application configuration file: see `samples/conf/app1.cnf`
  - Place this in `/etc/MHA/app1.cnf`
- Global configuration file: see `/etc/MHA/masterha_default.cnf` (see `samples/conf/masterha_default.cnf`)

app1.cnf

```
[server default]
manager_workdir=/var/log/masterha/app1
manager_log=/var/log/masterha/app1/manager.log

[server1]
hostname=host1

[server2]
hostname=host2
candidate_master=1

[server3]
hostname=host3

[server4]
hostname=host4
no_master=1
```

no need to specify master as MHA auto-detects this

sets priority, but doesn't necessarily mean it gets promoted as a default (say its too far behind replication).
But maybe this is a more powerful box, or has a better setup

will never be the master. RAID0 instead of RAID1+0?
Slave is in another data centre?

masterha_default.cnf

```
[server default]
user=root
password=rootpass
ssh_user=root
master_binlog_dir= /var/lib/mysql,/var/log/mysql
remote_workdir=/data/log/masterha
ping_interval=3
# secondary_check_script=masterha_secondary_check -s remote_host1 -s
remote_host2
# master_ip_failover_script= /script/masterha/master_ip_failover
# shutdown_script= /script/masterha/power_manager
# report_script= /script/masterha/send_report
# master_ip_online_change_script= /script/masterha/master_ip_online_change
```

check master activity from manager->remote_hostN->master (multiple hosts to ensure its not a network issue)

STONITH

MHA uses SSH

- MHA uses SSH actively; passphraseless login
  - In theory, only require Manager SSH to all nodes
  - However, remember `masterha_secondary_check`
  - `masterha_check_ssh --conf=/etc/MHA/app1.cnf`

Check replication

- `masterha_check_repl --conf=/etc/MHA/app1.cnf`
- If you don't see **MySQL Replication Health is OK**, MHA will fail
- Common errors? Master binlog in different position, read privileges on binary/relay log not granted, using multi-master replication w/o read-only=1 set (only 1 writable master allowed)

MHA Manager

- `masterha_manager --conf=/etc/MHA/app1.cnf`
- Logs are printed to `stderr` by default, set `manager_log`
- Recommended running with `nohup`, or `daemontools` (preferred in production)
- http://code.google.com/p/mysql-master-ha/wiki/Runnning_Background

So, the MHA Playbook

- Install MHA node, MHA manager
- `masterha_check_ssh --conf=/etc/app1.cnf`
- `masterha_check_repl --conf=/etc/app1.cnf`
- `masterha_manager --conf=/etc/app1.cnf`
- That's it!

master_ip_failover_script

- Pacemaker can monitor & takeover VIP if required
- Can use a catalog database
  - map between application name + writer/reader IPs
- Shared VIP is easy to implement with minimal changes to `master_ip_failover` itself (however, use shutdown_script to power off machine)

master_ip_online_change

- Similar to master_ip_failover script, but used for online maintenance
  - `masterha_master_switch --master_state=alive`
- MHA executes `FLUSH TABLES WITH READ LOCK` after the writing freeze

Test the failover

- `masterha_check_status --conf=/etc/MHA/app1.cnf`
- Kill MySQL (kill -9, shutdown server, kernel panic)
- MHA should go thru failover (stderr)
  - parse the log as well
- Upon completion, it stops running

masterha_master_switch

- Manual failover
  - `--master_state=dead`
- Scheduled online master switchover
- Great for upgrades to server, etc.
  - `masterha_master_switch --master_state=alive --conf=/etc/MHA/app1.cnf --new_master_host=host2`

Handling VIPs

```
my $vip = ‘192.168.0.1/24”;
my $interface = “0”;
my $ssh_start_vip = “sudo /sbin/ifconfig eth0:$key $vip”;
my $ssh_stop_vip = “sudo /sbin/ifconfig eth0:$key down”;
...
sub start_vip() {
`ssh $ssh_user\@$new_master_host \” $ssh_start_vip \”`; }
sub stop_vip() {
`ssh $ssh_user\@$orig_master_host \” $ssh_stop_vip \”`; }
```

Integration with other HA solutions

- Pacemaker
  - on RHEL6, you need some HA add-on, just use the CentOS packages
  - `/etc/ha.d/haresources` to configure VIP
    - `` `masterha_master_switch --master_state=dead --interactive=0 --wait_on_failover_error=0 --dead_master_host=host1 --new_master_host=host2` ``
- Corosync + Pacemaker works well

Solaris

- MHA tested to work on Linux and Solaris 10 & greater
- Use a .pkg MySQL
- Solaris 10 needs a compiler (SolarisStudio - register w/ Oracle to download)
- CPAN: manual builds of dependencies
- Solaris doesn't have md5sum (call md5)
- Solaris ssh isn't OpenSSH, missing some features like ConnectionTimeout

What about replication delay?

- By default, MHA checks to see if slave is behind master. By more than 100MB, it is never a candidate master
- If you have `candidate_master=1` set, consider setting `check_repl_delay=0`
- You can integrate it with `pt-heartbeat` from Percona Toolkit
  - http://www.percona.com/doc/percona-toolkit/2.1/pt-heartbeat.html

MHA deployment tips

- When testing, you really should install this as root
- SSH needs to work across all hosts
- If you don't want plaintext passwords in config files, use init_conf_load_script
- Each monitor can monitor multiple MHA pairs (hence app1, app2, etc.)
- You can have a standby master, make sure its read-only
- By default, master1->master2->slave3 doesn't work
  - MHA manages master1->master2 without issue
  - Use multi_tier_slave=1 option
- Make sure replication user exists on candidate master too!

Alternate solutions

- Heartbeat + DRBD
  - Costs $$$ -> passive master
  - innodb_flush_log_at_trx_commit=1, sync_binlog=1 is expensive in MySQL
    - http://www.mysqlperformanceblog.com/2008/06/02/how-much-overhead-drdb-could-cause/
- MariaDB 5.3/5.5 or Percona Server 5.5 or MySQL 5.6 mitigate this by having group commit in the binary log

Alternate solutions II

- MySQL NDB Cluster
- Galera Cluster
  - MariaDB Galera Cluster, Percona XtraDB Cluster
- Percona Replication Manager (PRM)
  - https://raw.github.com/y-trudeau/resource-agents-prm/master/heartbeat/mysql
- Tungsten Replicator

mysqlfailover

- mysqlfailover from mysql-utilities using GTID's in 5.6
- target topology: 1 master, n-slaves
- enable: log-slave-updates, report-host, report-port, master-info-table=TABLE
- modes: elect (choose candidate from list), auto (default), fail
- --discover-slaves-login for topology discovery
- monitoring node: SPoF
- Errant transactions prevent failover!
- Restart node? Rejoins replication topology, as a slave.

MariaDB 10

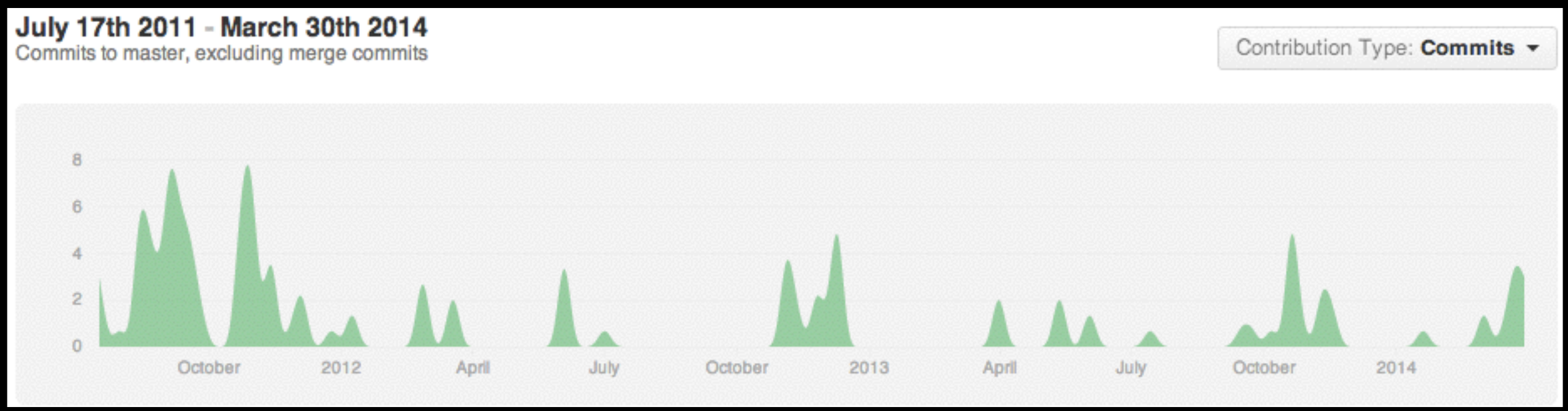
- New slave: SET GLOBAL GTID_SLAVE_POS = BINLOG_GTID_POS("masterbin.00024", 1600); CHANGE MASTER TO master_host="10.2.3.4", master_use_gtid=slave_pos; START SLAVE;
- use GTID: STOP SLAVE
  CHANGE MASTER TO
  master_use_gtid=current_pos; START SLAVE;
- Change master: STOP SLAVE
  CHANGE MASTER TO master_host="10.2.3.5";
  START SLAVE;

Where is MHA used

- DeNA
- Premaccess (Swiss HA hosting company)
- Ireland's national TV & radio service
- Jetair Belgium (MHA + MariaDB!)
- Samsung
- SK Group
- DAPA
- Facebook

MHA 0.56 is out now!

- MHA 0.56 released April 1 2014 (0.55: December 12 2012)
- http://code.google.com/p/mysql-master-ha/wiki/ReleaseNotes



MHA 0.56

- 5.6 GTID: GTID + auto position enabled? Failover with GTID SQL syntax not relay log failover
  - MariaDB 10 needs work
  - MySQL 5.6 support for checksum in binlog events + multi-threaded slaves
  - mysqlbinlog and mysql in custom locations (configurable clients)
  - binlog streaming server supported

MHA 0.56

- ping_type = INSERT (for master connectivity checks - assuming master isn't accepting writes)
- future
  - Ship MHA resource agent - http://www.skysql.com/downloads/mha-master-high-availability
  - MariaDB 10 gtid handling

Is fully automated failover a good idea?

- False alarms
  - Can cause short downtime, restarting all write connections
- Repeated failover
  - Problem not fixed? Master overloaded?
  - MHA ensures a failover doesn't happen within 8h, unless --last_failover_minute=n is set
- Data loss
  - id=103 is latest, relay logs are at id=101, loss
  - group commit means sync_binlog=1, innodb_flush_log_at_trx_commit=1 can be enabled! (just wait for master to recover)
- Split brain
  - sometimes poweroff takes a long time

Support

- SkySQL: www.skysql.com
- Percona: www.percona.com
- PalominoDB: www.palominodb.com
- AccelerationDB: www.accelerationdb.com

Automated tools to play with

- 4-host Vagrant setup for MySQL MHA
  - https://github.com/hholzgra/vagrant-mysql-mha
- Palomino Cluster Tool
  - https://github.com/time-palominodb/PalominoClusterTool
  - Ansible playbooks for MHA
- 4-host Vagrant setup for MHA
  - https://github.com/lefred/vagrant-mha

Video resources

- Yoshinori Matsunobu talking about High Availability & MHA at Oracle MySQL day
  - http://www.youtube.com/watch?v=CNCALAw3VpU
- Alex Alexander (AccelerationDB) talks about MHA, with an example of failover, and how it compares to Tungsten
  - http://www.youtube.com/watch?v=M9vVZ7jWTgw

References

- Design document
  - http://www.slideshare.net/matsunobu/automated-master-failover
- Configuration parameters
  - http://code.google.com/p/mysql-master-ha/wiki/Parameters
- JetAir MHA use case
  - http://www.percona.com/live/mysql-conference-2012/sessions/case-study-jetair-dramatically-increasing-uptime-mha
- MySQL binary log
  - http://dev.mysql.com/doc/internals/en/binary-log.html

Q&A

colin@mariadb.org | byte@bytebot.net
http://skysql.com/ | http://mariadb.org/
twitter: @bytebot | url: http://bytebot.net/blog/