Preparing for the Big Oops!
Disaster Recovery Sites for MySQL

Robert Hodges, CEO, Continuent
MySQL Conference 2011
Topics

/ Introductions
/ A Motivating Story
/ Master / Slave Disaster Recovery
  • MySQL Replication
  • Tungsten Replicator
/ Multi-Master and Disaster Recovery
/ Lessons Learned
About Continuent

Continuent is the leading provider of data replication and clustering for open source relational databases

Our Products:
- Tungsten Replicator - High-performance, MySQL replication
- Tungsten Enterprise - Commercial replication and data management solution for MySQL and PostgreSQL

Our Services:
- Consulting on Tungsten plus replication and clustering in general
- Subscriptions for commercial products
A Motivating Story (in Case You Need It)
This Story Starts Out Well

/ There was once a SaaS application for time entry
/ It lived in a co-lo in Sacramento
/ Consultants entered hours and billed their customers
/ And life was good
But There Was a Small Problem…

/ A backhoe cut the ISP power connection
/ The backup power failed
/ And darkness descended upon the site…
It Ended Badly for Everyone

/ The consultants could not enter hours
/ The accountants could not generate invoices
/ The companies could not bill customers
/ And there was wailing and gnashing of teeth
The Moral of This Story

Perhaps we should think about disaster recovery before the disaster
Widely Varying Assumptions

- How much data do you have?
- What is your transaction load?
- How far apart are your sites?
- What is your budget for extra hardware?
- Can you tolerate data loss? How much?
- How much unplanned downtime can you tolerate?
- Do you have maintenance windows?
- Do you run in the cloud or in a co-lo?
Basic DR With MySQL Master/Slave Replication
MySQL Replication Explained in One Slide

Master

I/O Thread downloads binlog

Slave

I/O thread writes relay logs; updates master.info

SQL Thread reads and applies relay logs; updates relay-log.info

File: mysql-bin.003531
Pos: 3333245

Master_Log_File: mysql-bin.003531
Read_Master_Log_Pos: 3333245

Relay_Log_File: relay-mysql-bin.005922
Exec_Master_Log_Pos: 33332189

© Continuent 2011
MySQL [Very] Basic DR Setup

Main Site

[mysqld]
ssl
slave_compressed_protocol

my.cnf

saasco.com (TTL=300s)

DR Site

[mysqld]
ssl
slave_compressed_protocol

my.cnf

SSL encryption / ssh tunnel compression
Thoughts on DR Site Capacity

- Your DR site must bear your full application load
- The best way to ensure that is to test it
- You can trade off DR site redundancy vs. cost
- Less reasonable trade: using different DR environment to save money
- Example: Run production on local hardware, DR site in Amazon
MySQL Run Book: Failover

1. Stop main site DBMS
2. Stop main site apps
3. Stop slave; Make DR site writable.
4. Start DR site apps
5. Shift DNS

saasco.com (TTL=300s)
Fencing MySQL Servers

/ Make MySQL instance non-writable
  mysql> set global read_only=1;

/ Remove accounts/passwords/privileges
  mysql> update mysql.user set Password='goaway' where User='saasuser';
  mysql> flush privileges;

/ Stop the DBMS
  # service mysqld stop
MySQL Run Book: Assessing Damage

Main site master may have “lost” transactions

1. Run SHOW SLAVE on DR master to find main site master position
   // (Master_Log_File && Exec_Master_Log_Pos)

2. Use mysqlbinlog on main site master to read binlog(s)

3. Any master transaction past the slave position is lost

// Can apply directly, inform users to resubmit, or just ignore
MySQL Run Book: Failback

1. Provision main site DBMS from backup; start slave
2. Put DR in maint mode
3. START SLAVE; Make DR read only
4. Start main site apps
5. Shift DNS

saasco.com (TTL=300s)
MySQL Hot vs. Cold Backup

**mysqld**

- **DBMS Data Files**

  - **Backup**
    - **HOT/COLD**: mysqldump
    - **HOT**: XtraBackup
    - **HOT**: InnoDB Hot Backup

  - **Backup**
    - **HOT**: FS snapshot with InnoDB
    - **COLD**: File system copy
Zen of Backup and Restore

/ Disasters are a bad time to learn on the job about backups and provisioning

/ Example: Provision slave from master SAN snapshot (assumption: using InnoDB)

1. Snapshot file system under data dir
2. Copy/mirror to slave
3. Restart slave mysqld
4. CHANGE MASTER TO using coordinates from mysqld .err file
Master Site Availability

/ Site failover is bad and should be avoided if possible
/ All site components should be redundant
Master Site Redundancy via Local Slave

MySQL slave replication position defined by master binlog file and offset

Failover difficult without working master
Master Site Redundancy via File System

- **Good**: Master site failover possible; allows local maintenance
- **Bad**: Master site slave promotion very slow for large datasets; non-uniform architecture

MySQL master → MySQL slave (Main Site)
MySQL slave → MySQL slave (DR Site)

DRBD or shared disk
Monitoring

/ Component failures will undo the best DR strategy
/ Monitor everything (Nagios & Cacti)
/ Don’t just look for failures
/ Make sure critical tasks succeed
/ Example: Check backup results very carefully
Still Not Out of the Woods…

/ Slave lag creates windows where failover may be delayed

/ Lag is especially problematic for:
  • mysqld restart
  • Catch-up after reprovisioning
  • I/O-bound workloads, e.g., large batches or complex updates

/ Unapplied transactions create possibility of multi-hour data loss
DR With Tungsten Replication
Tungsten Replicator Architecture

Master

MySQL

Binlogs

Tail binlog or login as client

Tungsten Replicator (replicator)

Transaction History Log

(Transactions + Metadata)

Transport via TCP/IP connection

Slave

MySQL

replicator.properties

Tungsten Replicator (replicator)

replicator.properties

Transport via TCP/IP connection

Apply using JDBC

© Continuent 2011
Tungsten Global ID and Epoch Number

<table>
<thead>
<tr>
<th>Replicator Log</th>
<th>Replicator Log</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seqn</td>
<td>Epoch</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
</tr>
</tbody>
</table>

Backup reloadable on any server

Backup seqno=3

Epoch numbers prevent inconsistent slaves from connecting

© Continuent 2011
DR Site Enabling with Global IDs

Main Cluster
- Master
- Slave
- Replicator
- Floating IP
- Backups

DR Cluster
- Slave (Relay)
- Replicator
- Backups

Seqno 35
Epoch 1

Seqno 25
Epoch 1

Seqno 28
Epoch 1

Seqno 21
Epoch 1

Seqno 35
Epoch 1

Seqno 25
Epoch 1
Reconfiguration After Failover

Old Master
Replicator Manager

New Master
Replicator

Floating IP

Main Cluster
Backups

Seqno 35
Epoch 1
(epoch number prevents reconnect as slave)

Seqno 29
Epoch 29

DR Cluster
Backups

Seqno 25
Epoch 1
Slave (Relay)
Replicator

Seqno 21
Epoch 1
Slave
Replicator
# Tungsten Multi-Site Run Book

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Difference vs. Native MySQL?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Failover</td>
<td>Major steps identical</td>
</tr>
<tr>
<td>Damage assessment</td>
<td>Major steps identical; new master promotion point readable from Tungsten message log</td>
</tr>
<tr>
<td>Failback</td>
<td>Major steps identical</td>
</tr>
<tr>
<td>Backup and Restore</td>
<td>Hot master backups require special handling; slave backups are simpler</td>
</tr>
</tbody>
</table>
Tag transactions with **shard ID** to show independence

- Shard Assignment
- Partition
- Parallel Queue
- Apply data from each channel independently on slave

Buffer channels in **parallel queue**

- Apply
- Apply
- Apply

MySQL Binlogs

Parallel apply reduces lag 2-3x for suitable workloads
It’s Still Not All Gravy…

/ Tungsten cures brittle slave promotion and reduces lag with parallel apply

But…

/ Tungsten is [a lot] newer than MySQL
/ Tungsten does not support SSL or compression between master and slave
/ App changes may be necessary for parallel apply

/ Both MySQL and Tungsten require regular testing and monitoring to work properly
Combining Multi-Master with Disaster Recovery
The Million Dollar Question

/ Will disaster recovery work?
/ Testing provides assurance but needs to be regular and thorough
/ Why not keep both sites active all the time?
Single Writable Master

Tests pathways on DR site; helps if app can split reads and writes

© Continuent 2011
Multi-Master with Partitioned Writes

MySQL replication requires DRBD/fixed disk

Need app discipline to prevent invalid updates
Full Multi-Master

PHP App
Apache
Mod_PHP

MySQL
master

MySQL
slave

Main
Site

Requires app rewrite to remove conflicts

Recovery from failures is hard

Reasonable if you did multi-master for other reasons

PHP App
Apache
Mod_PHP

MySQL
slave

MySQL
slave

DR
Site

© Continuent 2011
Multi-Master Is an Evolving Effort

- PHP App
  - Connector
  - MySQL master
    - Replicator
    - Manager
  - MySQL slave
    - Replicator
    - Manager

Tungsten Enterprise Cluster

Main Site

- MySQL master
  - Replicator
  - Manager
- MySQL slave
  - Replicator
  - Manager

Tungsten Enterprise Cluster

DR Site

- MySQL slave
  - Replicator
  - Manager
- MySQL slave
  - Replicator
  - Manager

© Continuent 2011
Summary

// MySQL native replication and Tungsten replicator offer powerful disaster recovery capabilities
1. Study your chosen solution carefully
2. Script everything you can
3. Put everything else in your Run Book
4. Ensure your DR site has enough capacity
5. Test the daylights out of it
6. Run as much as possible on both sites
7. Monitor everything
Parting Thoughts

It is rare to find a problem that you cannot make worse by bad corrective action.

Know the technology.

Don’t call audibles in a crisis.
Contact Information

HQ
560 S. Winchester Blvd., Suite 500
San Jose, CA 95128
Tel (866) 998-3642
Fax (408) 668-1009
e-mail:
robert.hodges@continuent.com

Continucent Web Site:
http://www.continuent.com

Tungsten Replicator 2.0:
http://code.google.com/p/tungsten-replicator

© Continuent 2011