High Availability Solutions for MySQL

Lenz Grimmer

<lenz@grimmer.com>

2008-08-29

DrupalCon 2008, Szeged, Hungary
Agenda

- High Availability in General
- MySQL Replication
- MySQL Cluster
- DRBD
- Links/Tools
Why High Availability Matters

- Downtime is expensive
- You miss $$$
- Your Boss complains
- New Site visitors won't come back
What Is HA Clustering?

- One service goes down → others take over its work
- IP address takeover, service takeover
- **Not designed** for high-performance
- **Not designed** for high throughput (load balancing)
Split-Brain

- Communications failures can lead to separated partitions of the cluster
- If those partitions each try and take control of the cluster, then it's called a split-brain condition
- If this happens, then bad things will happen
  [http://linux-ha.org/BadThingsWillHappen](http://linux-ha.org/BadThingsWillHappen)
- Use Fencing or Moderation/Arbitration to avoid it
Eliminating the SPOF

- Identify what will fail
  - Disks
- Find out what can fail
  - Network cables
  - OOM
  - Power supplies
Rules of High Availability

- Prepare for failure
- Keep it simple, stupid (KISS)
- Complexity is the enemy of reliability
- Test your setup frequently
MySQL Replication

- One-way, statement-based
- One Master, many Slaves
- Asynchronous – Slaves can lag
- Master maintains binary logs & index
- Easy to set up
- Built into MySQL
- Replication is single-threaded
Replication Topologies

- **Master > Slave**
- **Master > Slave > Slaves**
- **Master < > Master (Multi-Master)**
- **Master > Slaves**
- **Masters > Slave (Multi-Source)**
- **Ring (Multi-Master)**
Replication & HA

- Combined with Heartbeat
- Virtual IP takeover
- Slave gets promoted to Master
- Side benefits: load balancing & backup
- Tricky to fail back
- No automatic conflict resolution
- Proper failover needs to be scripted
Master-Master Replication

- Useful for easier failover
- Not suitable for load-balancing
  - Writes still end up on both machines
  - Neither machine has the authoritative data
- Don't write to both masters!
- Use Sharding or Partitioning instead (e.g. MySQL Proxy)
MySQL Cluster

- Shared nothing
- Automatic partitioning
- Distributed Fragments
- Synchronous replication
- Fast automatic fail-over of data nodes
- Automatic resynchronization
- Transparent to Application
- Supports Transactions
MySQL Cluster

- In-memory tables
- Not suitable for all query patterns
- Not suitable for large datasets
- Latency matters
- Can be combined with MySQL Replication
DRBD

- Distributed Replicated Block Device
- “Raid-1 over network”
- Synchronous block replication
- Automatic resync on recover
- Application-agnostic
- Can mask local I/O errors
- Active/passive configuration
DRBD & Heartbeat

- Heartbeat mounts file system on failover (passive node becomes active)
- Data only accessible on the active node
- (LVM snapshots can work around this)
- Increased I/O Latency
- Failover is “cold” (fsck, log recovery, buffers/caches)
<table>
<thead>
<tr>
<th>Availability</th>
<th>Requirements</th>
<th>MySQL Replication</th>
<th>MySQL Replication &amp; Heartbeat</th>
<th>MySQL, Heartbeat &amp; DRBD</th>
<th>MySQL Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automated IP Failover</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Automated DB Failover</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Typical Failover time</td>
<td>Varies</td>
<td>Varies</td>
<td>&lt; 30s</td>
<td>&lt; 3s</td>
<td></td>
</tr>
<tr>
<td>Auto resync of data</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Geographic redundancy</td>
<td>Yes</td>
<td>Yes</td>
<td>MySQL Replication</td>
<td>MySQL Replication</td>
<td></td>
</tr>
<tr>
<td>Scalability</td>
<td>Built-in load balancing</td>
<td>MySQL Replication</td>
<td>MySQL Replication</td>
<td>MySQL Replication</td>
<td>Yes</td>
</tr>
<tr>
<td>Read-intensive</td>
<td>Yes</td>
<td>Yes</td>
<td>MySQL Replication</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Write-intensive</td>
<td>No</td>
<td>No</td>
<td>Possible</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>#Nodes/Cluster</td>
<td>Master/Slave(s)</td>
<td>Master/Slave(s)</td>
<td>Active/Passive</td>
<td>255</td>
<td></td>
</tr>
</tbody>
</table>
Related tools / Links

- Linux Heartbeat
  [http://linux-ha.org/](http://linux-ha.org/)
- DRBD
  [http://drbd.org/](http://drbd.org/)
- Linux Cluster Information Center
  [http://www.lcic.org/ha.html](http://www.lcic.org/ha.html)
- Red Hat Cluster Suite
- Sun Open High Availability Cluster
  [http://opensolaris.org/os/project/ha-mysql/](http://opensolaris.org/os/project/ha-mysql/)

Tools/Links

- MySQL Multi-Master Replication Manager
  http://code.google.com/p/mysql-master-master/

- Maatkit
  http://maatkit.sourceforge.net/

- Mon – scheduler and alert management
  http://www.kernel.org/software/software/mon/

- Continuent Tungsten Replicator
  https://community.continuent.com/community/continuent-tungsten-replicator
Q & A

Questions, Comments?
Thank you!
Lenz Grimmer <lenz@grimmer.com>