MySQL Replikation für Einsteiger

Oli Sennhauser
FromDual GmbH
Uster / Schweiz

Schlüsselworte:

MySQL, Replikation, Performance, Hochverfügbarkeit, Scale-Out, Master, Slave

Einleitung

Die Replikation ist eines der wichtigsten Features von MySQL. Sei es aus Performance- oder Hochverfügbarkeitsgründen, um die MySQL Replikation kommt man früher oder später nicht darum herum.

Was ist Replikation?

Unter Replikation versteht man das Weiterreichen von Daten von einer Datenbank, üblicher als Master bezeichnet, auf eine oder mehrere andere Datenbanken, üblicherweise als Slave bezeichnet. Die weiter zu reichenden Daten stammen normalerweise von einer Applikation, welche diese Daten in der Master-Datenbank ablegt.
Unter Daten verstehen wir entweder DML-Statements (Update, Insert, Delete, etc.) oder binäre Events welche die Daten(-änderungen) beinhalten.

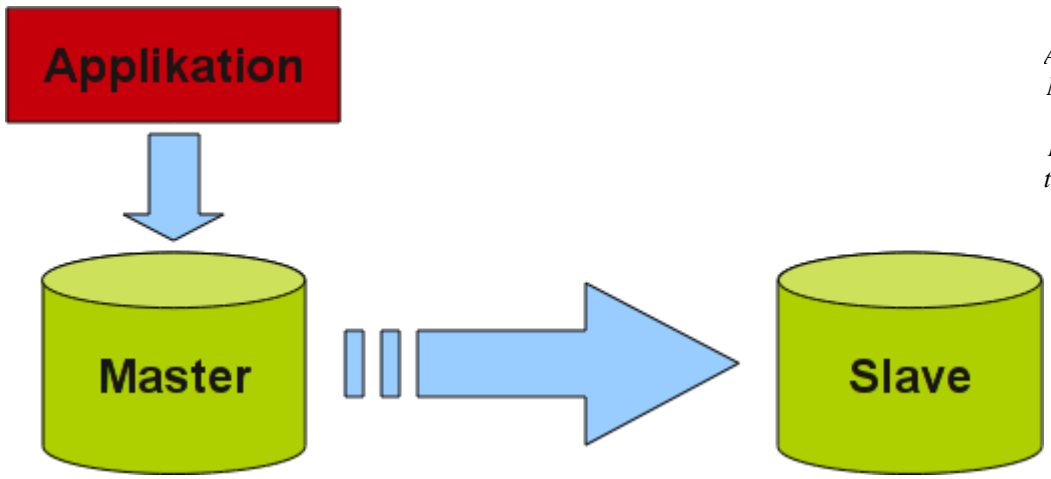

Abb. 1: Master/ Slave Replikation

Die MySQL Replikation

Durch das einschalten des sogenannten Binary Logs wird der Master dazu veranlasst, seine Statements in die sogenannten Binary Log Files zu schreiben.
Der Slave verbindet sich mittels des IO_threads auf den Master und fordert die entsprechenden Informationen an, welche er gerne als nächstes haben möchte. Sobald der Slave diese Informationen

erhalten hat, legt er sie in sein sogenanntes Relay Log ab, welches als Zwischenpuffer der Binary Logs auf dem Slave dient.
Ein Zweiter Thread, der sogenannten SQL_thread liest nun dieses Relay Log aus und appliziert die darin enthaltenen Informationen auf den Slave.
Dieser ganze Replikationsprozess erfolgt asynchron.

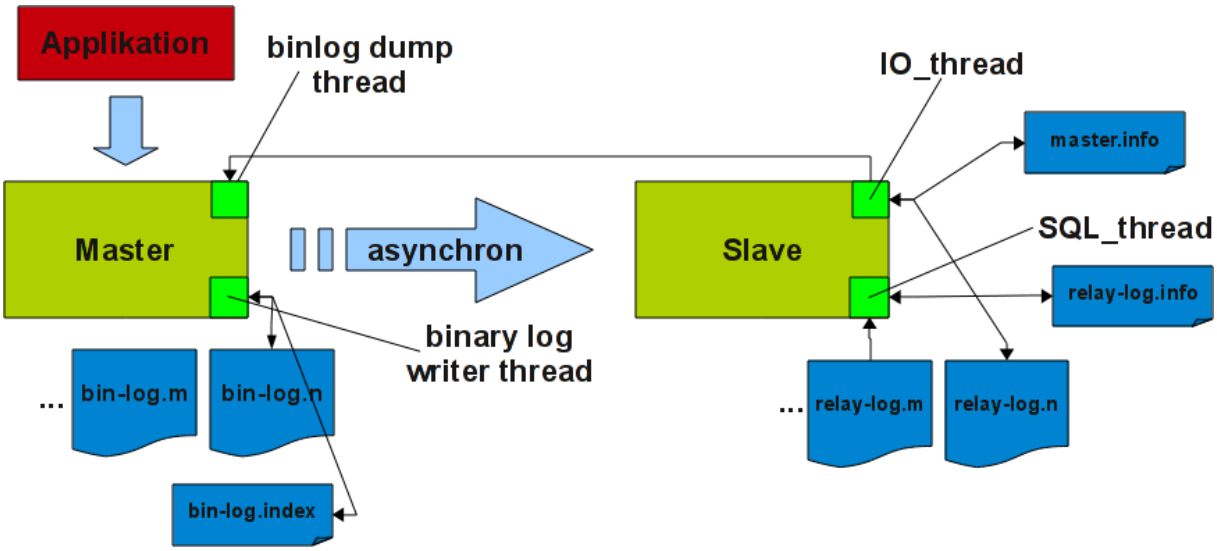

Abb. 2: MySQL Replikation

Vorbereiten des Masters

Um eine MySQL Datenbank zum Master zu machen sind 2 Schritte erforderlich. In der MySQL Konfigurationsdatei muss das Binary Logging mittels der Variable log-bin eingeschaltet werden. Zusätzlich muss in einem Replikationsverbund jede Komponente eine eindeutige ID vergeben werden, welches mittels der Variable server-id erfolgt.
Um diese Einstellungen zu aktivieren muss die Datenbank neu gestartet werden.
Anschliessend muss auf dem Master der entsprechende User angelegt werden, mit welchem der oder die Slaves auf den Master zugreifen sollen.

```
CREATE USER 'replication'@'192.168.1.%'
IDENTIFIED BY 'secret';

GRANT REPLICATION SLAVE ON *.*
TO 'replication'@'192.168.1.%';
```

Um einen Slave aufsetzen zu können ist es zudem notwendig einen initialen konsistenten Dump des Masters zu ziehen, welcher zusätzlich die dazugehörenden Information des Binary Log Files sowie der aktuellen Position, welche dem Dump entspricht, beinhaltet. Diese geschieht mit dem Befehl:

```
mysqldump --all-databases --single-transaction \
          --master-data > full_dump.sql
```

Aufsetzen des Slaves

Um einen neuen Slave aufzusetzen empfiehlt es sich, zuerst auf einem separaten Rechner eine neue MySQL Instanz zu installieren und zu starten. Diese Instanz muss eine eignen server-id aufweisen, da sonst die MySQL Replikation nicht funktionieren wird,
Anschliessend muss man dem zukünftigen Slave mitteilen, wo sein Master zu finden ist und welcher User für die Replikation verwendet werden soll.

```
CHANGE MASTER TO master_host='192.168.1.42'
, master_port=3306, master_user='replication'
, master_password='secret';
```

Dann kann der auf dem Master gezogene initiale Dump auf dem Slave eingespielt werden. Nach einem finalen überprüfen aller Einstellungen kann der Slave mit dem Befehl START SLAVE gestartet werden.
Der Slave verbindet sich nun automatisch mit dem Master und holt sich die ihm noch fehlenden Informationen ab.

Probleme beim Aufsetzen und Betrieb

Sofern man der Anleitung in der MySQL Dokumentation folgt funktioniert die MySQL Replikation gut. Die häufigsten Ursachen warum die MySQL Replikation nicht oder fehlerhaft funktioniert liegt darin, dass die Anleitung in der MySQL Dokumentation nicht korrekt befolgt wurde.
Ein häufig gemachter Fehler ist zum Beispiel, dass kein konsistentes Backup der Daten auf dem Master gemacht wurde. Die geschieht dann, wenn der Parameter --single-transaction nicht verwendet wird oder wenn jedes Schema einzeln auf dem Slave eingespielt wird.
Dann kommt es z.B. zu Primary Key Verletzungen.

Probleme, welche sich beim Betrieb ergeben sind oft dadurch begründet, dass die Master/Slave Replikation nicht überwacht wird. Dabei ist zu überwachen, dass der Slave nicht allzu oft und allzu weit hinter dem Master her hinkt. Ist dies der Fall, gilt zu überprüfen, was hierfür die Ursache ist (zu schwache Hardware, nicht optimale Abfragen, etc.).
Wenn auf dem Slave Daten manipuliert werden oder die Replikation unsauber aufgesetzt wurde kann es bei der Replikation zu Fehlern führen, welche diese unterbrechen. Auch dies sollte überwacht werden.
Auf dem Master sollte man von Zeit zu Zeit die Binary Logs weg löschen, wenn diese nicht mehr gebraucht werden um ein voll laufen der Disks zu vermeiden. Dies geschieht mit dem Befehl: PURGE BINARY LOGS.
Bei Fehlmanipulation oder mit Statement Based Replikation (SBR) kann es auch zu Dateninkonsitstenzen zwischen Master und Slave kommen. Diese sollten behoben oder der ggf. der Slave neu aufgesetzt werden, falls sie auftreten.

Wann wird die MySQL Replikation gebraucht?

Die MySQL Replikation wird für verschiedene Szenarien eingesetzt. Oft wird sie verwendet um Hochverfügbarkeit zu erreichen. Fällt der Master aus, kann auf dem Slave weiter gearbeitet werden. Bei einem sehr hohen Leseaufkommen können mehrere Slaves verwendet werden, um diese Last zu

bewältigen (Read Scale-Out). Ein ähnliches Szenario ist dann gegeben, wenn wir den Slave fürs Backup verwenden wollen oder zu Reporting- und Datenanalyse Zwecke einsetzen.

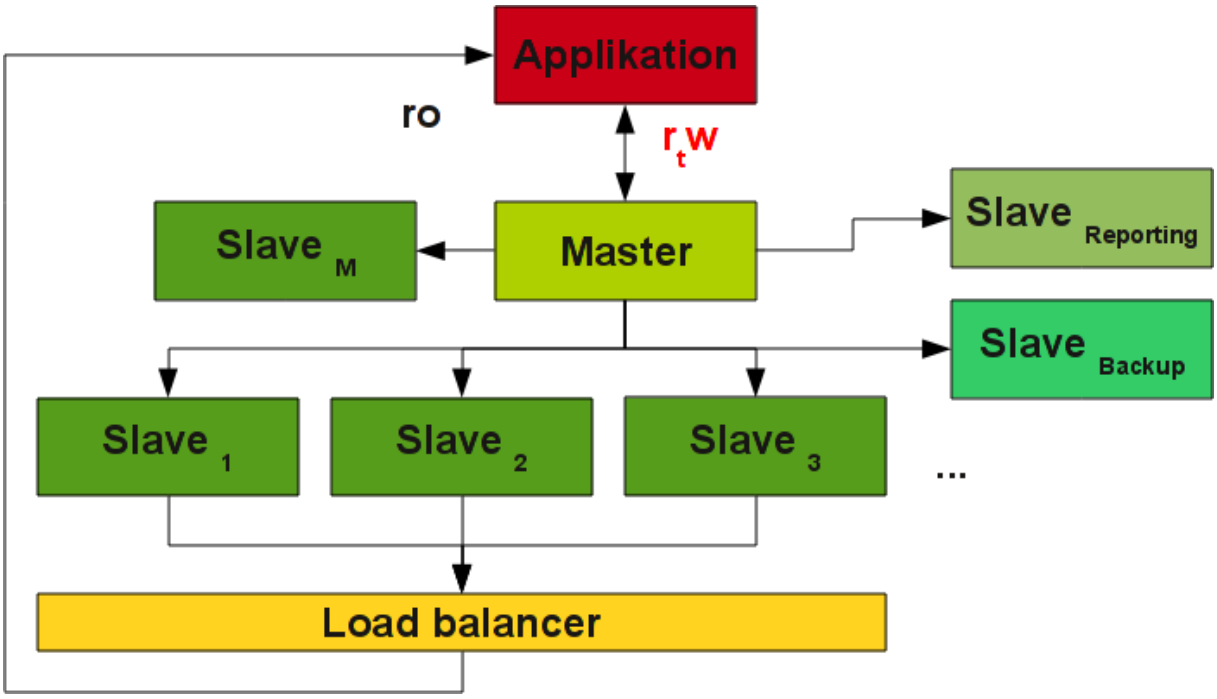

Abb. 2: MySQL Scale-Out

Neuerungen in der MySQL Replikation

Mit MySQL 5.1 ist neu die Row Based Replikation (RBR) hinzugekommen. Da die "alte" Statement Base Replikation (SBR) gewisse Nachteile betreffen Datensicherheit und Datenkonsistenz mit sich bringt ist es empfehlenswert auf das neue Format zu schwenken wenn hier höchste Anforderungen bestehen. Das Umschalten erfolgt einfach durch das ändern des Parameters binlog_format auf "row". RBR gilt als die sicherere Art der Replikation.

Die Replikation bis und mit MySQL 5.1 erfolgt asynchron. Diese bedeutet, dass wir keine Garantie dafür haben, dass Daten, welche auf dem Master geschrieben wurden, auch auf dem Slave angekommen sind.
Um den Datenverlust im Falle eines Master-Crashes zu minimieren ist in MySQL 5.5 die semi-synchrone Replikation hinzugekommen. Dieses Feature kann in MySQL 5.5 zugeschaltet werden und es stellt sicher, dass mindestens 1 Slave die Daten vom Master auch wirklich erhalten hat. Somit haben wir die Gewähr, dass im Fall eines Crashes keine Daten verloren gehen und wir mit dem vollständigen Datenset auf dem Slave weiterarbeiten können.
Diesen Vorteil handeln wir uns mit einer geringfügig schlechteren Schreibperformance ein.

Ein Nachteil der in MySQL 5.5 hinzugekommen Row Based Replikation (RBR) ist, das unter Umständen die Datenmenge, welche vom Master auf den Slave übermittelt wird je nach Art der Abfragen signifikant anwächst.
Um diesem Verhalten entgegen zu wirken hat MySQL in Version 5.6 das neue Feature namens Row Image Control eingeführt. Mit dem Parameter binlog_row_image kann gesteuert werden ob eine Zeile

komplett oder ob nur die geänderten Spalten übermittelt werden sollen. Dadurch lässt sich die Datenmenge in einigen Szenarien signifikant reduzieren und auch die Slave-Performance verbessern.

Ebenfalls mit MySQL 5.6 neu hinzugekommen sind die Crash Safe Binary Logs. Es sollte jetzt also nicht mehr möglich sein, dass beim Crash eines Master irgendwelche Inkonsistenzen zwischen Transaktionslog und Binary Log auftreten. Das sollte auch die Replikation zum Slave noch robuster machen.
Um auch auf der Slave-Seite bei der Replikation mehr Robustheit zu erlangen können die beiden Dateien master.info und relay-log.info neu als Tabellen in der Datenbank abgelegt werden. Zur zeit werden diese noch als (nicht Crash-sichere) MyISAM Tabellen angelegt. Man kann diese aber anschliessend von Hand selber nach InnoDB umwandeln.

Weiter hinzugekommen ist das Remote Binary Log Shipping. Mit dem mysqlbinlog Utility können jetzt neu die Binary Logs vom Master auf eine andere Maschine kopiert werden, damit im Falle eines Disk-Ausfalles die Binary Logs noch vorhanden sind.

Delayed Replikation ist ebenfalls neu hinzugekommen. Früher musste man hierfür eine externes Skript aus der Maatkit-Toolbox bemühen. Neu kann man einen Slave mit dem CHANGE MASTER TO MASTER_DELAY Befehl hinterherhinken lassen.

Und endlich sind auch die Variablen zur Steuerung des Binary Logs sauber getrennt und implementiert mit log_bin_basename und log_bin, welche für das benennen der Binary Logs und für das Ein- und Ausschalten des Binary Log dienen.

Features welche zur Zeit in den MySQL Labs verfügbar sind, sind multi-threaded Slaves und Mutli-Source Replikation.

Varianten der Replikation

Es gibt noch zwei alternative Varianten zur MySQL Replikation. Zum einen gibt es mit Galera eine vollständig synchrone Replikation, welche ebenfalls die Verfügbarkeit und die Skalierbarkeit eines Datenbank Backends erhöht. Diese Replikation ist für die Applikation vollständig transparent und soll sehr gut skalieren.
Um diese Lösung zum Laufen zu bringen braucht es speziell gepatchte MySQL Binaries, welche mit Galera mitgeliefert werden.

Ein weiteres Produkt ist der Tungsten Replikation Cluster. Diese Lösung baut auf den MySQL Binary Logs auf und verspricht ebenfalls eine höhere Verfügbarkeit, mehr Performance und eine bessere Datenintegrität durch das Einführen einer global Transaction ID.
Zwei nette Features dieser Lösung sind die Möglichkeit von MySQL nach PostgreSQL und Oracle und umgekehrt zu replizieren und die Möglichkeit der Multi-Source Replikation.
Ein Nachteil dieser Lösung ist, dass Sie zusätzlich Java und Ruby erfordert und nicht nahtlos in dieMySQL Tools integriert ist.

Die Seitenzahl wird von uns eingefügt!

Bitte fügen Sie Ihre Kontaktadresse hinzu.

Kontaktadresse:

Name
FromDual GmbH
Rebenweg, 6
CH-8610 Uster

Telefon: +41 79-830-09-33
Fax: +41 43 55 68204
E-Mail oli.sennhauser@fromdual.com
Internet: www.fromdual.com