

# Master 不停機並修改 Slave 參數

## Prerequisite

- 必須要有兩台以上的 Slave Server；若只有一台，要確定該主機沒有提供 prod 的任何服務。
- 若有 A、B 兩台 Slave Server，要先確定所有的 READ 服務都已經導到 A，這樣才可以對 B 進行以下動作；反之亦然。

## DM 環境

- 跳板機: 3.39.26.208
- (Master) HL-PROD-mysql-rp01: 10.23.1.180
- (Slave) HL-PROD-mysql-rp02: 10.23.1.181
- (Slave) HL-PROD-mysql-rp03: 10.23.1.182
- (BackUp) HL-PROD-mysql-rp04: 10.25.1.183

## 2022-07-22：DM 的 DB 設定調整之流程

1. 系統組通知 DM 的 RD：現在要進行系統調整（預計 14:00）
2. 系統組將所有 Slave 的服務導到 HL-PROD-mysql-rp02 (10.23.1.181)，切斷所有導向 HL-PROD-mysql-rp03 (10.23.1.182)
3. 第一次服務重啟
4. DBA 確認無任何服務導向 HL-PROD-mysql-rp03 (10.23.1.182)，特別是 10.23.2.140/10.23.2.141/10.25.2.142
5. 系統組通知 DM 的 RD：檢查服務是否有異常
6. DBA 進入 HL-PROD-mysql-rp03 (10.23.1.182)，暫停 MySQL Server，修改 my.cnf 並存檔，重新啟動 MySQL Server
  - 確認有無 Nano
  - 確認服務連線狀態(Server 端)
  - 確認服務連線狀態(MySQL 端)
  - 進入 MySQL(Slave) 暫停
  - 暫停 Slave Server
  - 新增參數到 /etc/my.cnf
  - 啟動 Slave Server
  - 進入 MySQL(Slave) 重新啟動
7. DBA 確認 HL-PROD-mysql-rp03 (10.23.1.182) 有與 Master 同步
8. DBA 通知系統組：HL-PROD-mysql-rp03 (10.23.1.182) 已修改完畢
9. 系統組將所有 Slave 的服務導到 HL-PROD-mysql-rp03 (10.23.1.182)，切斷所有導向 HL-PROD-mysql-rp02 (10.23.1.181)
10. 第二次系統重啟

11. DBA 確認無任何服務導向 HL-PROD-mysql-rp02 (10.23.1.181)，特別是 10.23.2.140/10.23.2.141/10.25.2.142
12. 系統組通知 DM 的 RD：檢查服務是否有異常
13. DBA 進入 HL-PROD-mysql-rp02 (10.23.1.181)，暫停 MySQL Server，修改 my.cnf 並存檔，重新啟動 MySQL Server
  - 確認有無 Nano
  - 確認服務連線狀態(Server 端)
  - 確認服務連線狀態(MySQL 端)
  - 進入 MySQL(Slave) 暫停
  - 暫停 Slave Server
  - 新增參數到 /etc/my.cnf
  - 啟動 Slave Server
  - 進入 MySQL(Slave) 重新啟動
14. DBA 確認 HL-PROD-mysql-rp02 (10.23.1.181) 有與 Master 同步
15. DBA 通知系統組：HL-PROD-mysql-rp02 (10.23.1.181) 已修改完畢
16. 系統組將 Slave 的服務導回原本對 HL-PROD-mysql-rp03 (10.23.1.182) 與 HL-PROD-mysql-rp02 (10.23.1.181) 的分配
17. 第三次系統重啟
18. DBA 確認 HL-PROD-mysql-rp03 (10.23.1.182) 與 HL-PROD-mysql-rp02 (10.23.1.181) 都有收到來自 10.23.2.140/10.23.2.141/10.25.2.142 的連線
19. 系統組通知 DM 的 RD：檢查服務是否有異常
20. (若無異常) 系統組通知 DM 的 RD：系統調整完成

## 確認有無 nano

```
## 確認有無 vim
rpm -qa | grep vim

## 確認有無 nano
rpm -qa | grep nano

## 安裝 nano
sudo yum -y install nano
```

## 確認服務連線狀態(Server 端)

```
## 要預先知道服務的 ip(10.23.2.140/10.23.2.141/10.25.2.142)
## 或是服務連線只剩下自己的 master/slave，就是已經切得乾淨了
netstat -an | grep 3306
netstat -na | grep 3306

# 兩指令的結果相同
```

## 確認服務連線狀態(MySQL 端)

```
## 搜尋結果為空，才是所有服務已斷開
mysql> SELECT user, left(host, 11) FROM information_schema.PROCESSLIST where user = 'lduser' group by user, left(host, 11);
```

## 進入 MySQL(Slave) 暫停

```
mysql> STOP SLAVE;
mysql> FLUSH TABLES;
# (if it gets stuck here, then you might need to fix it. You can try UNLOCK TABLES.)
```

## 暫停 Slave Server

```
$ sudo systemctl stop mysqld
```

## 新增參數到 /etc/my.cnf

```
#### 以下為 2022-07-14 由 VS 要求的參數調整

slave_parallel_workers = 16
# 平行處理總數 | Elvis 建議可以 CPU/2 來決定

slave_pending_jobs_size_max = 134217728
# 每次 pending 工作不會超過 128M

slave_parallel_type = 'LOGICAL_CLOCK'
# 以 timestamp 順序來執行平行處理

binlog_transaction_dependency_tracking = 'COMMIT_ORDER'
# 開啟平行處理的 default

slave_preserve_commit_order = 1
# 依照 master 來的順序執行平行處理

log_slave_updates = 1
# slave_preserve_commit_order 如果開啟，則 log_slave_updates 必須為 1
# 記得要關閉原本的 log-slave-updates = 0

slave_transaction_retries = 10
# 重試次數 | 10 = default

master_info_repository = 'TABLE'
# slave 紀錄 master 的狀態，以 table 方式記錄。則資料庫中會有資料表 mysql.slave_master_info

relay_log_info_repository = 'TABLE'
# slave 同步 master 的訊息紀錄，以 table 方式記錄。則資料庫中會有資料表 mysql.slave_relay_info

relay_log_recovery = 1
# 從上次斷線時間重新抄寫 relay-log
```

## 啟動 Slave Server

```
$ sudo systemctl start mysqld
```

## 進入 MySQL(Slave) 重新啟動

```
mysql> START SLAVE;
# 實作發現：不用做這件事，SLAVE 也會自動回到同步狀態
# (normally this would be done automatically when you restart mysql)
mysql> SHOW SLAVE STATUS\G
# (check that replication is working properly)
```

## 結果檢視

```
## 在 Slave 檢視
mysql> show slave status\G
***** 1. row *****
      Slave_IO_State: Waiting for master to send event
          ...: ...
      Slave_IO_Running: Yes
      Slave_SQL_Running: Yes
          ...: ...
      Seconds_Behind_Master: 0
          ...: ...
      Slave_SQL_Running_State: Slave has read all relay log; waiting for more updates
```

1. 兩 Slave 與 Master 的同步狀態正常
2. seconds\_behind\_master 沒有秒數延遲
3. 現有16個 worker thread進行同步處理（設定 slave\_parallel\_workers = 16）
4. 持續觀察交易高峰時段，cpu 與 memory 的狀態

## 備註

1. 本次版更為 Matrix 主動發起，客戶端 RD 配合(14:00 為它們決定)。
2. 此次版本更新時間約歷時 100 分鐘(14:00 - 15:40)。
3. 未來此類調整，傾向於由 Matrix 發動，避免在版本更新時若出問題，責任難以釐清。

## 參考來源

- <http://www.jonathanlevin.co.uk/2010/05/how-to-safely-restart-your-mysql-slave.html>