

# MySQL on CLUSTERPRO for Linux HOWTO

## 1 はじめに

この文章は、CLUSTERPRO for Linux上でMySQLを動作させる際に参考となる情報を記述したものです。MySQLを片方向および双方向スタンバイで運用するための設定方法や注意点を述べます。

この文章を書くにあたって次のディストリビューションと同梱されているMySQLを使用しました。このほかのバージョンのディストリビューションやMySQLを別途インストールした場合でも、いくつかの設定項目の読み替えでクラスタ化できると思われます。

- TurboLinux 7 + MySQL(3.23.39.4)
- RedHat Linux 7.2 + MySQL(3.23.41)
  
- CLUSTERPRO SE for Linux Ver2.0

## 2 責任範囲

この文章は、MySQLをクラスタ環境で使用するための注意点や設定例を参考情報として示すものであり、これらの動作保証を行うものではありません。

## 3 事前準備

MySQLを両サーバにインストールし、MySQLのユーザを作成しておいてください。

(詳しくは、MySQLのドキュメントをご覧ください)

Linuxのインストール時にディストリビューションのインストール画面でMySQLのインストールを指定している場合は、通常、特に準備を行う必要はありません。

## 4 片方向スタンバイ用MySQL環境の構築手順

以下にMySQLを片方向スタンバイで運用する際の構築方法を説明します。双方向スタンバイにする際の片側のシステムも同様の方法で構築します。双方向スタンバイのもう片方のシステムの構築は、後述します。

### 4.1 前提環境

以下のような片方向スタンバイ環境を想定し、説明します。

クラスタサーバ環境

	サーバ1 (運用系)	サーバ2 (待機系)
実IPアドレス	10.0.0.1	10.0.0.2
切替パーティション	/mnt/mysql1	

MySQLのクライアント環境

	クライアント1	クライアント2	クライアント3
実IPアドレス	10.0.1.1	10.0.1.2	10.0.1.3

## グループ情報

	グループ 1
フローティングIPアドレス	10.0.0.11
切替パーティション	/mnt/mysql1

## データベース環境

データベース名	DATABASE1
データファイルの場所	/mnt/mysql1
コンフィグレーションファイル名	/mnt/mysql1/my.cnf
TCP/IP接続時のポート番号	3306
ローカル接続用に使用されるソケットファイル名	/var/lib/mysql/mysql.sock
pidファイル名	/var/lib/mysql/mysql.pid

## 4.2 データベース領域の初期化

下記の手順で、データベース領域を初期化します。

- (1)グループ 1 をサーバ 1 で起動する。
- (2)MySQLユーザでサーバ 1 のLinuxにログインする。
- (3)mysql\_install\_db --datadir=/mnt/mysql1

注意)本段階では、グループ 1 のスクリプトでMySQLの起動・終了用の記述はありません。既定値のスクリプトのままでよいです。スクリプトの記述は、4.6の段階で行います。

## 4.3 コンフィグレーションファイルの作成

下記の内容で、MySQLの起動時の動作設定を行うコンフィグレーションファイル(/mnt/mysql1/my.cnf)を作成します。

```
[client]
port = 3306
socket = /var/lib/mysql/mysql.sock

[mysqld]
port = 3306
socket = /var/lib/mysql/mysql.sock
pid-file = /var/lib/mysql/mysql.pid
datadir = /mnt/mysql1/
```

## 4.4 データベースの作成

下記の手順で、MySQLのデータベースを作成します。

まず(1)でMySQLを起動して、(2)でデータベースの作成となります。

```
(1)safe_mysqld --defaults-file=/mnt/mysql1/my.cnf &
(2)mysqladmin --defaults-file=/mnt/mysql1/my.cnf create DATABASE1
```

## 4.5 ネットワーク接続権限の作成

下記の手順で、ネットワークの接続権限を作成します。実運用においては、MySQLのクライアント環境によって適宜指定パラメータ値を変更してください。

(1)でデータベースに接続し、(2)でそのデータベースに対するネットワーク権限を作成します。  
(3)でMySQLを終了し、(4)以降、サーバ2で同様の操作を行います。

```
(1)mysql -S /var/lib/mysql/mysql.sock DATABASE1
(2)mysql> grant all on * to root@'10.0.0.11/255.255.255.0';
```

```
mysql> grant all on * to root@'10.0.1.1/255.255.255.255';
mysql> grant all on * to root@'10.0.1.2/255.255.255.255';
mysql> grant all on * to root@'10.0.1.3/255.255.255.255';
mysql> quit
```

(3)mysqladmin --defaults-file=/mnt/mysql1/my.cnf shutdown

(4)グループ 1 をサーバ 2 へ移動する。

(5)サーバ 2 へmysqlユーザでログインする。

(6)safe mysqld --defaults-file=/mnt/mysql1/my.cnf &

(7)mysql -S /var/lib/mysql/mysql.sock DATABASE1

```
(8)mysql> grant all on * to root@'10.0.0.11/255.255.255.0';
mysql> grant all on * to root@'10.0.1.1/255.255.255.255';
mysql> grant all on * to root@'10.0.1.2/255.255.255.255';
mysql> grant all on * to root@'10.0.1.3/255.255.255.255';
mysql> quit
```

(9)mysqladmin --defaults-file=/mnt/mysql1/my.cnf shutdown

(10)グループ 1 を停止する。

注意)grant文でネットワーク権限を作成しないと、リモートからMySQLに接続することができません。

## 4.6 スクリプトの作成

グループ 1 にMySQLの起動・終了スクリプトを作成します。

以下にそれぞれのサンプルスクリプトを示します。

start.bat

```
#! /bin/sh
#*****
#*                START.BAT                *
#*****
```

```

if [ "$ARMS_EVENT" = "START" ]
then
    if [ "$ARMS_DISK" = "SUCCESS" ]
    then
        armlog normal1
        if [ "$ARMS_SERVER" = "HOME" ]
        then
            armlog normal2
        else
            armlog on_other1
        fi
        armlog "MySQL start"
        safe_mysqld --defaults-file=/mnt/mysql1/my.cnf &
    else
        armlog "ERROR_DISK from START"
    fi
elif [ "$ARMS_EVENT" = "RECOVER" ]
then
    armlog recover
elif [ "$ARMS_EVENT" = "FAILOVER" ]
then
    if [ "$ARMS_DISK" = "SUCCESS" ]
    then
        armlog failover1
        if [ "$ARMS_SERVER" = "HOME" ]
        then
            armlog failover2
        else
            armlog on_ohter2
        fi
        armlog "MySQL start"
        safe_mysqld --defaults-file=/mnt/mysql1/my.cnf &
    else
        armlog "ERROR_DISK from FAILOVER"
    fi
else
    armlog no_arm
fi
armlog exit

```

## stop.bat

```

#!/bin/sh
#*****
#*                STOP.BAT                *
#*****

arm_rel_path() {
    while [ "$1" != "" ]
    do
        armrelpath $1 > /dev/null 2>&1
        relret=$?

        if [ "$relret" = "0" ]
        then
            armlog "KILL NO PROCESS"
        elif [ "$relret" = "1" ]
        then
            armlog "KILL SOME PROCESS"
        else
            armlog "ARMRELPATH ERROR"
        fi
        shift
    done
}

arm_rel_mntpoint() {
    mntpoint=`armlsmnt -l $ARMS_RESOURCELIST`
    mntret=$?
}

```

```

    if [ "$mntret" = "0" ]
    then
        if [ "$mntpoint" != "" ]
        then
            arm_rel_path $mntpoint
        else
            armlog "NO MOUNT POINT"
        fi
    else
        armlog "ARMLSMNT ERROR"
    fi
}

if [ "$ARMS_EVENT" = "START" ]
then
    if [ "$ARMS_DISK" = "SUCCESS" ]
    then
        armlog "NORMAL1"
        if [ "$ARMS_SERVER" = "HOME" ]
        then
            armlog "NORMAL2"
        else
            armlog "ON_OTHER1"
        fi
        armlog "MySQL stop"
        mysqladmin --defaults-file=/mnt/mysql/my.cnf shutdown
        arm_rel_mntpoint
    else
        armlog "ERROR_DISK from START"
    fi
elif [ "$ARMS_EVENT" = "FAILOVER" ]
then
    if [ "$ARMS_DISK" = "SUCCESS" ]
    then
        armlog "FAILOVER1"
        if [ "$ARMS_SERVER" = "HOME" ]
        then
            armlog "FAILOVER2"
        else
            armlog "ON_OTHER2"
        fi
        armlog "MySQL stop"
        mysqladmin --defaults-file=/mnt/mysql/my.cnf shutdown
        arm_rel_mntpoint
    else
        armlog "ERROR_DISK from FAILOVER"
    fi
else
    armlog "NO_ARM"
fi
armlog "EXIT"
exit 0

```

## 5 双方向スタンバイ用MySQL環境の構築手順

以下にMySQLを双方向スタンバイで運用する際の構築方法を説明します。

### 5.1 前提環境

以下のような双方向スタンバイ環境を想定し、説明します。

### クラスタサーバ環境

	サーバ1 (運用系)	サーバ2 (待機系)
実IPアドレス	10.0.0.1	10.0.0.2
切替パーティション	/mnt/mysql1 , /mnt/mysql2	

### MySQLのクライアント環境

	クライアント1	クライアント2	クライアント3
実IPアドレス	10.0.1.1	10.0.1.2	10.0.1.3

### グループ情報

	グループ1	グループ2
フローティングIPアドレス	10.0.0.11	10.0.0.12
切替パーティション	/mnt/mysql1	/mnt/mysql2

### データベース環境

データベース名	DATABASE1	DATABASE2
データファイルの場所	/mnt/mysql1	/mnt/mysql2
コンフィグレーションファイル名	/mnt/mysql1/my.cnf	/mnt/mysql2/my.cnf
TCP/IP接続時のポート番号	3306	3307
ローカル接続用に使用されるソケットファイル名	/var/lib/mysql/mysql.sock	/var/lib/mysql/mysql2.sock
pidファイル名	/var/lib/mysql/mysql.pid	/var/lib/mysql/mysql2.pid

## 5.2 双方向スタンバイの一方のデータベース環境の作成

双方向スタンバイの場合、2つのデータベース環境を作成する必要があります。その際、一方は、片方向スタンバイ時のデータベース環境の作成と同じ方法で作成します。一方のデータベース環境の作成が完了したら、以下の手順で、もう一方のデータベース環境を作成します。

## 5.3 データベース領域の初期化

下記の手順で、データベース領域を初期化します。

- (1)グループ2をサーバ1で起動する。
- (2)MySQLユーザでサーバ1のLinuxにログインする。
- (3)mysql\_install\_db --datadir=/mnt/mysql2

## 5.4 コンフィグレーションファイルの作成

下記の内容で、MySQLの起動時の動作設定を行うコンフィグレーションファイル(/mnt/mysql2/my.cnf)を作成します。

```
[client]
port = 3307
socket = /var/lib/mysql/mysql2.sock

[mysqld]
port = 3307
socket = /var/lib/mysql/mysql2.sock
pid-file = /var/lib/mysql/mysql2.pid
datadir = /mnt/mysql2/
```

## 5.5 データベースの作成

下記の手順で、MySQLのデータベースを作成します。

```
(1)safe_mysqlld --defaults-file=/mnt/mysql2/my.cnf &
```

```
(2)mysqladmin --defaults-file=/mnt/mysql2/my.cnf create DATABASE2
```

## 5.6 ネットワーク接続権限の作成

下記の手順で、ネットワークの接続権限を作成します。実運用においては、MySQLのクライアント環境によって適宜指定パラメータ値を変更してください。

```
(1)mysql -S /var/lib/mysql/mysql2.sock DATABASE2
```

```
(2)mysql> grant all on * to root@'10.0.0.12/255.255.255.0';
```

```
mysql> grant all on * to root@'10.0.1.1/255.255.255.255';
```

```
mysql> grant all on * to root@'10.0.1.2/255.255.255.255';
```

```
mysql> grant all on * to root@'10.0.1.3/255.255.255.255';
```

```
mysql> quit
```

```
(3)mysqladmin --defaults-file=/mnt/mysql2/my.cnf shutdown
```

(4)グループ2をサーバ2へ移動する。

(5)サーバ2へmysqlユーザでログインする。

```
(6)safe_mysqlld --defaults-file=/mnt/mysql2/my.cnf &
```

```
(7)mysql -S /var/lib/mysql/mysql2.sock DATABASE2
```

```
(8)mysql> grant all on * to root@'10.0.0.12/255.255.255.0';
```

```
mysql> grant all on * to root@'10.0.1.1/255.255.255.255';
```

```
mysql> grant all on * to root@'10.0.1.2/255.255.255.255';
```

```
mysql> grant all on * to root@'10.0.1.3/255.255.255.255';
```

```
mysql> quit
```

```
(9)mysqladmin --defaults-file=/mnt/mysql2/my.cnf shutdown
```

(10)グループ2を停止する。

## 5.7 スクリプトの作成

グループ2にMySQLの起動・終了スクリプトを作成します。

以下にそれぞれのサンプルスクリプトを示します。

### start.bat

```
#!/bin/sh
#*****
#*                START.BAT                *
#*****

if [ "$ARMS_EVENT" = "START" ]
then
    if [ "$ARMS_DISK" = "SUCCESS" ]
    then
        armlog normal1
        if [ "$ARMS_SERVER" = "HOME" ]
        then
            armlog normal2
        else
            armlog on_other1
        fi
        armlog "MySQL start"
        safe_mysqld --defaults-file=/mnt/mysql2/my.cnf &
    else
        armlog "ERROR_DISK from START"
    fi
elif [ "$ARMS_EVENT" = "RECOVER" ]
then
    armlog recover
elif [ "$ARMS_EVENT" = "FAILOVER" ]
then
    if [ "$ARMS_DISK" = "SUCCESS" ]
    then
        armlog failover1
        if [ "$ARMS_SERVER" = "HOME" ]
        then
            armlog failover2
        else
            armlog on_ohter2
        fi
        armlog "MySQL start"
        safe_mysqld --defaults-file=/mnt/mysql2/my.cnf &
    else
        armlog "ERROR_DISK from FAILOVER"
    fi
else
    armlog no_arm
fi
armlog exit
```

### stop.bat

```
#!/bin/sh
#*****
#*                STOP.BAT                *
#*****
```

```
#####
```

```
arm_rel_path() {
    while [ "$1" != "" ]
    do
        armrelpath $1 > /dev/null 2>&1
        relret=$?

        if [ "$relret" = "0" ]
        then
            armlog "KILL NO PROCESS"
        elif [ "$relret" = "1" ]
        then
            armlog "KILL SOME PROCESS"
        else
            armlog "ARMRELPATH ERROR"
        fi
    shift
    done
}

arm_rel_mntpoint() {
    mntpoint=`armlsmnt -l $ARMS_RESOURCELIST`
    mntret=$?

    if [ "$mntret" = "0" ]
    then
        if [ "$mntpoint" != "" ]
        then
            arm_rel_path $mntpoint
        else
            armlog "NO MOUNT POINT"
        fi
    else
        armlog "ARMLSMNT ERROR"
    fi
}

if [ "$ARMS_EVENT" = "START" ]
then
    if [ "$ARMS_DISK" = "SUCCESS" ]
    then
        armlog "NORMAL1"
        if [ "$ARMS_SERVER" = "HOME" ]
        then
            armlog "NORMAL2"
        else
            armlog "ON_OTHER1"
        fi
        armlog "MySQL stop"
        mysqladmin --defaults-file=/mnt/mysql2/my.cnf shutdown
        arm_rel_mntpoint
    else
        armlog "ERROR_DISK from START"
    fi
elif [ "$ARMS_EVENT" = "FAILOVER" ]
then
    if [ "$ARMS_DISK" = "SUCCESS" ]
    then
        armlog "FAILOVER1"
        if [ "$ARMS_SERVER" = "HOME" ]
        then
            armlog "FAILOVER2"
        else
            armlog "ON_OTHER2"
        fi
        armlog "MySQL stop"
        mysqladmin --defaults-file=/mnt/mysql2/my.cnf shutdown
        arm_rel_mntpoint
    else
        armlog "ERROR_DISK from FAILOVER"
    fi
else
    armlog "NO_ARM"
fi
armlog "EXIT"
exit 0
```

## 6 ApacheなどからCGI経由でMySQLにアクセスする場合について

MySQLへのアクセス方法として、クライアント側からフローティングIPを利用して直接アクセスする方法と、各サーバ上のWebサービス(Apacheなど)にフローティングIPで接続し、WebサービスのCGIなどからアクセスする方法とがあります。いずれの場合でも、上記の設定方法でクラスタ化することができます。CGI経由でアクセスする場合は、grant文でネットワーク接続権限を作成する際に、クライアント側のアドレスの権限を設定する必要がありません。