HA cluster construction memo (DRBD & Heartbeat)

HA クラスタ構築メモ(DRBD & Heartbeat)

Version 0.2

Copyright © 2008 LA TIGRE.

保証免責
本書は記載事項またはそれに関わる事項について、明示的あるいは暗黙的な保証はいたしておりません。
したがいまして、これらを原因として発生した損失や損害についての責任を負いません。

著作権
本書および本書に記載されておりますソフトウェア等は、著作権により保護されております。
また非商用以外に本書を、複製、再頒布することをかたく禁止いたします。

表記について

本書では以下の書体を使用しています。

- イタリック文字

  本文中でのコマンド、ファイル名、変数など可変なパラメータ値を表します。

- 等幅文字

  ファイルの内容やコマンドの入出力例に使います。入力の場合にはボールドで表します。

```
$ cd /usr/src/sys/i386/conf
$ ls
GENERIC         Makefile      OLDCARD       SMP
GENERIC.hints   NOTES         PAE           gethints.awk
$
```

- 省略文字

  ファイルの内容やコマンドの入出力例を省略する場合に'...'を使います。

```
$ vi /etc/rc.conf
  ...
sshd_enable="YES"
named_enable="YES"
  ...
$
```

- プロンプト

  一般または、管理権限を持った実行環境をそれぞれ、'$'(ドル)、'#'(シャープ)のプロンプトで表します。

```
$ su
Password: root's passwd
#
```

目次

1. 概要

1.1. はじめに

本書は安価な HA クラスタの構築手順について記述したものです。

LAMP スタックによるアプリケーションを構築する上でインフラ部分での高可用性を向上させるため、サーバ 2 台構成による DRBD と Heartbeat の組合せで HA クラスタを構成します。

また、サーバ間でストレージデバイスを共有します。

1.2. ハードウェア構成

サーバ 2 台、スイッチ 5 台、および外部ストレージ 1 台による HA 構成となります。

要素	モデル	備考
サーバ	Dell PowerEdge 2950 Ⅲ	2 台で HA 構成を行う。
スイッチ	Dell PowerConnect 5424	公開側は 2 台冗長し、サーバ側では NIC チーミングを行う。 防御側は DRBD 帯域用に 1 台、DB アクセス用に 1 台を確保。 RAC 用に 1 台。
ストレージ	Dell PowerVault MD1000	エンクロージャは 1 台だが、EMM モジュールや電源ユニットはデュアル構成となっている。 サーバ間を DAS 接続し、スプリットモードで動作させる。 利用可能ディスク容量要件(プライマリ/セカンダリともに 7TB を確保)により RAID レベルを 0(ストライピング)とする。

1.3. サーバ構成

2 台とも同じバージョンの OS を導入します。

サーバ名	プラットフォーム	カーネル
web01	Red Hat Enterprise Linux Server release 5.3 (x86_64)	2.6.18-128.1.1.el5
web02	Red Hat Enterprise Linux Server release 5.3 (x86_64)	2.6.18-128.1.1.el5

各サーバには下表のソフトウェアが導入されます。

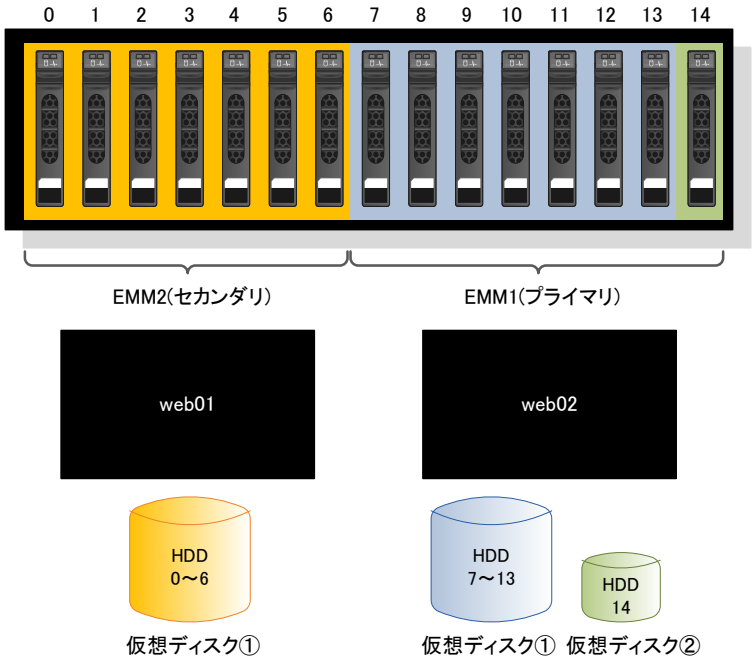
ソフトウェア名	バージョン	用途
DRBD	8.3.0	ネットワーク経由でのミラーリング。 4TB 以上のディスクと取扱うために、このバージョン以降を使用する必要があります。
Heartbeat	2.1.4	HA クラスタを構成するサーバ間の状態監視。
Apache	2.2.11	サービス用ウェブサービス。
MySQL Enterprise 5.1 Pro	5.1.31	サービス用データベース。

Copyright© 2008 LA TIGRE.

1.4. 仮想ディスク構成

導入した PowerVault MD1000 には 15 個の HDD を搭載し、1 個の HDD で 1TB、すなわち合計で 15TB のディスク容量が確保可能です。

下図のように、MD1000 をスプリットモードで動作させることで各サーバに約半分ずつディスク容量を割り当てます。



DRBD パーティションのサイズを揃えるため、ドライブ 14 は別の仮想ディスクとして設定しておきます。各種設定方法は下記を参照してください。

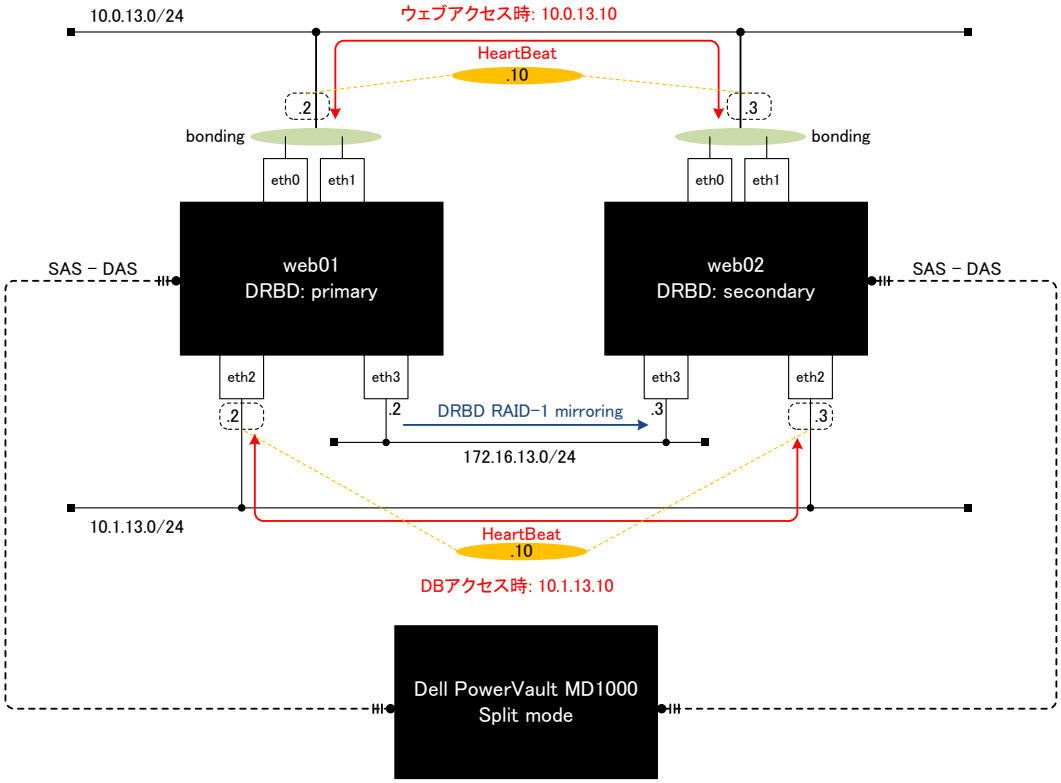
- Dell PowerVault MD1000 ハードウェアオーナーズマニュアル
- Dell Q&A 検索 LSI PERC5 BIOS からの Virtual Disk 作成方法(文書番号:195116)

HDD0～6(セカンダリ)、HDD7～13(プライマリ)は DRBD パーティションとして利用し、ネットワーク経由でミラーリングを行いますので、仮想ディスク構成、LVM 構成ともにディスク諸元を揃えておきます。

Copyright© 2008 LA TIGRE.

1.5. ネットワーク構成

下図のネットワーク構成で HA 環境を構築します。



また各サーバの IP アドレスは下表のように割り当てます。

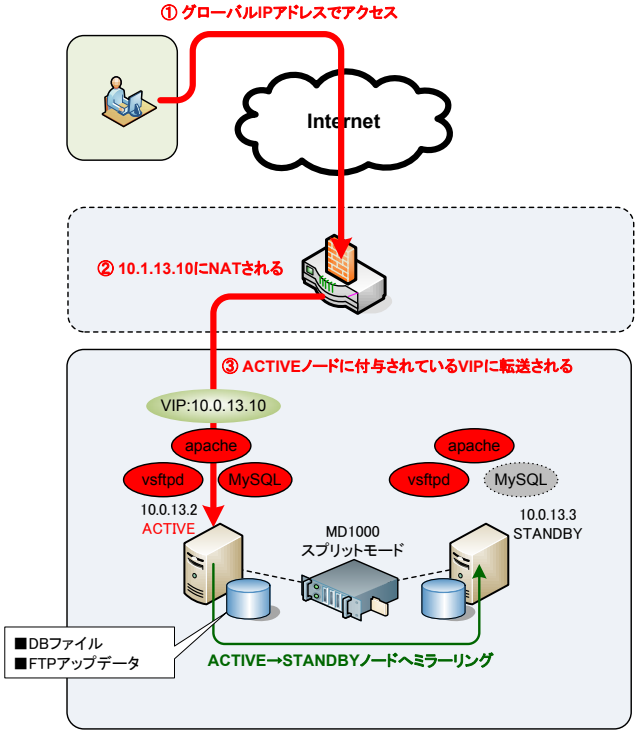
NIC 名	ノード名		用途
	web01	web02	
bond0	10.0.13.2	10.0.13.3	公開ネットワークセグメント用(web アクセス)の実 IP アドレス。
bond0:0	10.0.13.10	10.0.13.10	公開ネットワークセグメント用(web アクセス)の仮想 IP アドレス。 稼働系にのみ付与される。
eth0	—	—	bond0 のスレーブ用物理アダプタ。
eth1	—	—	bond0 のスレーブ用物理アダプタ。
eth2	10.1.13.2	10.1.13.3	保護ネットワークセグメント用(DB アクセス)の実 IP アドレス。
eth2:0	10.1.13.10	10.1.13.10	保護ネットワークセグメント用(DB アクセス)の仮想 IP アドレス。 稼働系にのみ付与される。
eth3	172.16.13.2	172.16.13.3	保護ネットワークセグメント用(DRBD アクセス)の実 IP アドレス。
RAC	10.2.13.2	10.2.13.3	保護ネットワークセグメント用(DRAC アクセス)の実 IP アドレス。

Copyright© 2008 LA TIGRE.

1.6. サービス構成

インターネットからのアクセスは全て稼動系ノードが処理します。

Heartbeat は稼働系ノードに仮想 IP アドレスを付与し、DRBD は共有リソースとして DB ファイル、FTP データファイル等をミラーリングします。



データベースファイルが共有リソースとなっているため、待機系では MySQL は非稼動状態になっています。グローバル IP アドレスでのアクセスは常に稼動系へ NAT されるため、待機系でも apache、vsftpd サービスは常時稼動状態で問題ありません。

Copyright© 2008 LA TIGRE.

2. DRBD 導入・設定

2.1. DRBD について

DRBD(Distributed Replicated Block Device)とは、2 つのノード間で特定のディスクパーティションをネットワーク経由で 1 方向ミラーリング(RAID1 相当)するソフトウェアです。

ディスクパーティションはブロックデバイスとして認識され、前章の「1.4. 仮想ディスク構成」に示したように、各サーバから認識されるブロックデバイス名は/dev/sdb となり、このパーティション上に共有データを配置します。

DRBD の配布物にはバイナリパッケージとソースコードがそれぞれ用意されています。

- バイナリパッケージ
  http://www.linbit.com/support/ に RHEL5 用の rpm パッケージがありますが、LIBNET 社のサポート契約が必要です。また、LInux カーネルのバージョンと密接に関係しているため、導入するサーバの Linux カーネルバージョンと合ったパッケージを取得する必要があります。
- ソースコード
  http://oss.linbit.com/drbd/ より最新バージョンのソースコードを取得してください。
  本書ではソースコードからビルドを行います。

DRBD 8.3.0 以降より 4TB 以上のブロックデバイスを扱えるようになりましたので、8.3.0 以降のバージョンを使用してください。

2.2. LVM の設定

ディスク増設等による拡張が行われた場合に、複数の物理パーティションを 1 つの論理パーティションとしてグルーピングするため LVM(Logical Volume Manager)の設定を行います。
これにより、DRBD パーティションを 1 つにまとめることが可能です。

```
# pvcreate /dev/sdb
# vgcreate -s 32m VolGroup01 /dev/sdb
# lvcreate -l 208543 -n DRBD00 VolGroup01
```

lvcreate の-l オプションで LE(Logical Extent)数を指定していますが、この値は pvdisplay コマンドの Total PE 値から得ることができます。-L オプションでバイトサイズを指定することも可能ですが端数が丸められる場合があり、全てのパーティション領域を割り当てる際には、-l オプションが有用です。

Copyright© 2008 LA TIGRE.

2.3. DRBD のインストール

① rpm-build ユーティリティをインストールします。

DRBD のソースコードには spec ファイルが含まれています。ソースから rpm パッケージを生成するため、このユーティリティを先にインストールしてください。

```
# yum install rpm-build
```

② ソースコードをダウンロードし、展開します。

```
# cd /usr/local/src
# wget http://oss.linbit.com/drbd/8.3/drbd-8.3.0.tar.gz
# tar ztvf drbd-8.3.0.tar.gz
```

③ rpm パッケージを生成します。

```
# cd drbd-8.3.0
# make rpm
```

④ rpm パッケージをインストールします。

```
# cd dist/RPMS/x86_64
# rpm -ivh drbd-8.3.0-3.x86_64.rpm
# rpm -ivh drbd-km-2.6.18_128.1.1.el5-8.3.0-3.x86_64.rpm
```

2.4. ミラーリング用の通信許可

前章の「1.5. ネットワーク構成」で示したように、DRBD ミラーリング帯域として 172.16.13.0/24 を確保しています。

本書ではこの帯域内で tcp/7788 を使用して通信を行いますので、ファイアウォール(iptables、TCP Wrappers)を動作させている場合は、上述のポートを開放します。

2.5. 名前解決

DRBD、および後述する HeartBeat はホスト名に基づきプライマリ、セカンダリを認識します。各ノードの/etc/hosts ファイルに対向のホスト情報を登録しておきます。

```
# vi /etc/hosts
  ...
10.1.13.2         web01
10.1.13.3         web02
  ...
```

また、/etc/nsswitch.conf を下記のように変更しておいた方がよいでしょう。

```
# vi /etc/nsswitch.conf
  ...
#hosts:  db files nisplus nis dns
hosts:   files dns
  ...
```

Copyright© 2008 LA TIGRE.

2.6. DRBD の設定

設定ファイルのサンプルをコピーし、編集します。

drbd.conf は web01 と web02 の両方に配置し、かつ同じ内容でなくてはなりません。

```
# cp /usr/share/doc/drbd-8.3.0/drbd.conf /etc/drbd.conf
# vi /etc/drbd.conf
  ...
```

具体的な設定内容は下記のとおりです。設定に関する詳細な情報は DRBD.CONF(5)のマニュアルページを参照してください。

```
 1 global {
 2    # minor-count 64;
 3    # dialog-refresh 5;
 4    # disable-ip-verification;
 5    usage-count yes;
 6 }
 7
 8 common {
 9 }
10
11 resource r0 {
12    protocol C;
13    syncer {
14      rate 125M;
15    }
16
17    startup {
18      wfc-timeout 0;
19      degr-wfc-timeout 120;
20    }
21
22    disk {
23      on-io-error detach;
24    }
25
26    net {
27    }
28
29    on web01 {
30
31      device /dev/drbd0;
32      disk /dev/VolGroup01/DRBD00;
33      meta-disk internal;
34      address 172.16.13.2:7788;
35
36    }
37    on web02 {
38
39      device /dev/drbd0;
40      disk /dev/VolGroup01/DRBD00;
41      meta-disk internal;
42      address 172.16.13.3:7788;
43
44    }
45 }
```

セクション/パラメータ	セマンティクス

global {
  # minor-count 64;
  # dialog-refresh 5;
  # disable-ip-verification;
  usage-count yes;
}
	global セクションは広域な設定パラメータを記述するのに使用される。
	global セクションは 1 回だけ記述できる。

	minor-count: DRBD カーネルモジュールを再ロードすることなく、多数のリソースを定義したい場合、このパラメータを指定する。デフォルトにより、現在定義されている(1 個:r0)リソースに加え、さらに 11 個のリソースが定義できる。実際にはカーネルモジュールを再ロードすることなく、32 個のリソースが定義可能。
	dialog-refresh: ダイアログの再描画時間間隔。デフォルトにより、1 秒間隔となる。

Copyright© 2008 LA TIGRE.

セクション/パラメータ	セマンティクス
	disable-ip-verification: drbdadm に対して IP アドレスの検証を行わせない。
	usage-count: DRBD 利用者統計に参加するには yes を指定する。
common { }	common セクションに記述したパラメータは全てのリソース(resource セクション)に継承される。
resource r0 { protocol C;	DRBD リソースを定義する。resource セクションには 2 つの on host セクションを定義しなければならない。
	protocol: DRBD プロトコルを指定する。 **A** ローカルディスクとローカル TCP 送信バッファへのデータ書込みをもって同期が完了したと判断する。 **B** ローカルディスクとリモートバッファキャッシュへのデータ書込みをもって同期が完了したと判断する。 **C** ローカルディスクとリモートディスク両方へのデータ書込みをもって同期が完了したと判断する。

syncer {
rate 125M;
}
	syncer セクションには同期デーモンの振舞いをチューニングするパラメータを定義する。

	rate: 同期に利用する帯域幅を指定する。 単位は bps ではなく Bps であることに注意。本書ではスイッチ、サーバ NIC、LAN ケーブルが 1Gbps であるため、換算して 125MBps と定義する。

startup {
wfc-timeout 0;
degr-wfc-timeout 120;
}
	startup セクションには同期デーモンの振舞いをチューニングするパラメータを定義する。

	wfc-timeout: ローカル/リモートサーバ間での接続タイムアウト値を指定する。 0 を指定すると恒久的に待機する。
	degr-wfc-timeout: ローカル/リモートサーバ間での接続タイムアウト値を指定する。

disk {
on-io-error detach;
}
	disk セクションにはストレージに対する扱いをチューニングするパラメータを定義する。

	on-io-error: デバイスに I/O エラーが発生した場合に実行するハンドラを定義する。 **pass_on** プライマリ側はファイルシステムへエラーを報告し、セカンダリ側では何も報告しない。 **call-local-io-error** ローカル定義したハンドラを実行する。 **detach** デバイスを切り離し、ディスクレスモードで処理を実行する。
net { }	net セクションにはネットワークに対する扱いをチューニングするパラメータを定義する。

on web01 {
 	
device /dev/drbd0;
disk /dev/VolGroup01/DRBD00;
meta-disk internal;
address 172.16.13.2:7788;
 	
}
 	
on web02 {
 	
device /dev/drbd0;
disk /dev/VolGroup01/DRBD00;
meta-disk internal;
address 172.16.13.3:7788;
	on セクションにはリソースを構成するノードについて定義する。
	ノード名は uname –n で得られる正規名でなければならない。

	device: 定義するリソースに対応するブロックデバイス名を指定する。 このブロックデバイスは DRBD によって自動的に生成される。 後述する **disk** パラメータに指定したデバイス名をこのパラメータに決して指定してはならない。
	disk: ブロックデバイス名を指定する。DRBD はこのデバイスに対してアクセスする。DRBD の動作中はこのデバイスに対して別の方法で決してアクセスしてはならない。

Copyright© 2008 LA TIGRE.

セクション/パラメータ	セマンティクス
} }	**meta-disk:** DRBD がミラーリングを行う際に利用するメタデータ確保場所を定義する。 **internal** を指定するとリソース定義したブロックデバイスの最後部分にメタデータが確保される。その場合、アプリケーションで利用可能なディスクサイズは僅かに減少する。 リソース定義外のブロックデバイスに対してメタデータを確保する場合には、meta-disk /dev/sdc1、または meta-disk /dev/sdc1[0]、meta-disk /dev/sdc[1]と書くことができる。 この場合のブロックデバイスは別のパーティションに分割し、メタデータ専用にしなければならない。
	address: 各リソースの同期対象となるノードの IP アドレスを指定する。 さらにTCPポート番号を指定しなければならないが、複数リソースを定義(本書で言えば、resource r0 {} に加えて resource r1 {} を定義した場合)する際は、同一 *IP:port* を共有することはできない。

2.7. DRBD メタデータの作成

プライマリ、セカンダリの両方でメタデータを作成します。

```
# drbdadm create-md r0
v08 Magic number not found
v07 Magic number not found
Writing meta data...
initialising activity log
NOT initialized bitmap
New drbd meta data block sucessfully created.
success
#
```

2.8. DRBD の起動

① 最初に web01 側から DRBD を起動します。

```
web01# /etc/rc.d/init.d/drbd start
Starting DRBD resources:    [ d(r0) s(r0) n(r0) ].
..........
***************************************************************
 DRBD's startup script waits for the peer node(s) to appear.
 - In case this node was already a degraded cluster before the
   reboot the timeout is 120 seconds. [degr-wfc-timeout]
 - If the peer was available before the reboot the timeout will
   expire after 0 seconds. [wfc-timeout]
   (These values are for resource 'r0'; 0 sec -> wait forever)
 To abort waiting enter 'yes' [  15]:yes
#
```

この時点では対向の web02 上で DRBD が動作していないため、上記のメッセージが出力されます。この場合はひとまず yes と入力します。

Copyright© 2008 LA TIGRE.

DRBD のステータスを確認すると、下記のように接続待ち状態であることがわかります。

```
web01# cat /proc/drbd
version: 8.3.0 (api:88/proto:86-89)
GIT-hash: 9ba8b93e24d842f0dd3fb1f9b90e29 build by admin@web01, 2009-03-01 15:10:34
 0: cs:WFConnection ro:Secondary/Secondary ds: Inconsistent/Inconsistent C r---
    ns:0 nr:0 dw:0 dr:0 al:0 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:6726102892
#
```

また、初回起動時にはどちらのノードもセカンダリ(Secondary/Secondary)の状態であり、ディスクも未同期(Inconsistent)のままです。

② web02 側の DRBD を起動します。

```
web02# /etc/rc.d/init.d/drbd start
Starting DRBD resources:    [ d(r0) s(r0) n(r0) ].
#
```

DRBD のステータスを確認すると、下記のように接続状態であることがわかります。

```
web02# cat /proc/drbd
version: 8.3.0 (api:88/proto:86-89)
GIT-hash: 9ba8b93e24d842f0dd3fb1f9b90e29 build by admin@web02, 2009-03-01 15:14:41
 0: cs:Connected ro:Secondary/Secondary ds: Inconsistent/Inconsistent C r---
    ns:0 nr:0 dw:0 dr:0 al:0 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:6726102892
#
```

web01 も接続状態になっているはずです。

```
web01# cat /proc/drbd
version: 8.3.0 (api:88/proto:86-89)
GIT-hash: 9ba8b93e24d842f0dd3fb1f9b90e29 build by admin@web01, 2009-03-01 15:10:34
 0: cs:Connected ro:Secondary/Secondary ds: Inconsistent/Inconsistent C r---
    ns:0 nr:0 dw:0 dr:0 al:0 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:6726102892
#
```

m

③ プライマリ、セカンダリのロールを決定します。

web01 をプライマリ、web02 をセカンダリとしてパーティションイメージを初期フル同期(initial full synchronization)を行います。プライマリ側で下記のコマンドを実行します。

```
web01# drbdadm -- --overwrite-data-of-peer primary r0
drbd0: Forced to consider local data as UpToDate!
web01#
```

同期の方向は常にプライマリ→セカンダリとなり、プライマリ側では直ちにブロックデバイスが利用可能になります。初期フル同期の完了には時間が掛かりますので、作業に当たっては並行してファイルシステム作成、マウント、ファイル生成等を行った方がよいでしょう。

初回フル同期実行中の DRBD のステータスは下記のようになります。

```
web01# cat /proc/drbd
version: 8.3.0 (api:88/proto:86-89)
GIT-hash: 9ba8b93e24d842f0dd3fb1f9b90e29 build by admin@web01, 2009-03-01 15:10:34
 0: cs:SyncSource ro:Primary/Secondary ds: UpToDate/Inconsistent C r---
    ns:991232 nr:0 dw:0 dr:991232 al:0 bm:60 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:6101027921
        [>....................] sync'ed:  0.2% (610102/672610)M
        finish: 09:40:13 speed: 121,260 (121,324) K/sec
#
```

Copyright© 2008 LA TIGRE.

```
web02# cat /proc/drbd
version: 8.3.0 (api:88/proto:86-89)
GIT-hash: 9ba8b93e24d842f0dd3fb1f9b90e29 build by admin@web02, 2009-03-01 15:14:41
 0: cs:SyncTarget ro:Secondary/Primary ds:Inconsistent/UpToDate C r---
    ns:991232 nr:0 dw:0 dr:991232 al:0 bm:60 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:6101027921
        [>....................] sync'ed:  0.2% (610102/672610)M
        finish: 09:40:13 speed: 121,260 (121,324) K/sec
#
```

2.9. ファイルシステムの作成

プライマリ側の DRBD パーティションにファイルシステムの作成、チューニングを行います。

```
web01# mkfs.ext3 /dev/drbd0
web01# tune2fs -i 0 /dev/drbd0
web01# tune2fs -L DRBD /dev/drbd0
web01# tune2fs -l /dev/drbd0
```

2.10. DRBD の動作確認

① プライマリ側でファイルシステムをマウントします。

```
web01# cd /
web01# mkdir /drbd
web01# mount -t ext3 -orw /dev/drbd0 /drbd
```

② 適当なテスト用ファイルを生成してみます。

```
web01# cd /drbd
web01# touch foo.txt
web01#
web01# df -h
Filesystem            Size  Used  Avail  Use%  Mounted on
/dev/mapper/VolGroup00-LogVol00
                      534G  10G   496G   2%    /
/dev/mapper/VolGroup00-LogVol02
                      4.9G  139M  4.5G   3%    /snapshot
/dev/sda1             99M   25M   70M    27%   /boot
tmpfs                 2.0G  0     2.0G   0%    /dev/shm
/dev/drbd0            6.3T  100M  6.2T   1%    /drbd
web01#
```

③ web01 をセカンダリに降格します。

```
web01# umount /drbd
web01# drbdadm -- secondary r0
web01# cat /proc/drbd
version: 8.3.0 (api:88/proto:86-89)
GIT-hash: 9ba8b93e24d842f0dd3fb1f9b90e29 build by admin@web01, 2009-03-01 15:10:34
 0: cs:Connected ro:Secondary/Secondary ds:UpToDate/UpToDate C r---
    ns:6726102892 nr:6726102888 dw:0 dr:0 al:2 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:0
web01#
web01# df -h
Filesystem            Size    Used    Avail    Use%    Mounted on
/dev/mapper/VolGroup00-LogVol00
                      534G    10G     496G     2%      /
/dev/mapper/VolGroup00-LogVol02
                      4.9G    139M    4.5G     3%      /snapshot
/dev/sda1             99M     25M     70M      27%     /boot
tmpfs                 2.0G    0       2.0G     0%      /dev/shm
web01#
```

④ web02 をプライマリに昇格し、ファイルシステムをマウントします。

```
web02# drbdadm -- primary r0
web02# cat /proc/drbd
version: 8.3.0 (api:88/proto:86-89)
GIT-hash: 9ba8b93e24d842f0dd3fb1f9b90e29 build by admin@web01, 2009-03-01 15:10:34
 0: cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r---
    ns:6726102892 nr:6726102888 dw:0 dr:0 al:2 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:0
web02#
web02# cd /
web02# mkdir drbd
```

Copyright© 2008 LA TIGRE.

```
web02# mount -t ext3 -orw /dev/drbd0 /drbd
web02#
```

⑤ web02 上で先ほど生成したテスト用ファイルとファイルシステムを確認。

```
web02# ls /drbd
total 1
-rw-r--r-- 1 root root  0 Mar 13 15:21 foo.txt
web02#
web02# df -h
Filesystem            Size  Used  Avail  Use%  Mounted on
/dev/mapper/VolGroup00-LogVol00
                      534G  10G   496G   2%    /
/dev/mapper/VolGroup00-LogVol02
                      4.9G  139M  4.5G   3%    /snapshot
/dev/sda1             99M   25M   70M    27%   /boot
tmpfs                 2.0G  0     2.0G   0%    /dev/shm
/dev/drbd0            6.3T  100M  6.2T   1%    /drbd
web02#
```

2.11. 外部スクリプトの配置

① 下記のスクリプトをプライマリ、セカンダリに用意します。

```
# cd /etc/rc.d/init.d
# vi drbd-wait
#!/bin/bash
#
# chkconfig: 345 70 8
# description: drbd-wait
#

### BEGIN INIT INFO
# Provides: drbd-wait
# Required-Start: $network $syslog sshd
# Required-Stop:
# Default-Start:  3 5
# Default-Stop:   0 1 2 6
# Description:    Wait drbd sync on target
### END INIT INFO

DRBDSETUP=/sbin/drbdsetup
DEVICE=/dev/drbd0
INITWAIT=5

case "$1" in
    start)
    echo -n "Waiting DRBD: "
    sleep ${INITWAIT}

    CSTATE=`${DRBDSETUP} ${DEVICE} cstate`
    if [ "${CSTATE}" = "SyncTarget" ]; then
        echo -n "Syncing... "
        ${DRBDSETUP} ${DEVICE} wait-sync
    fi

    echo

    ;;
    stop)
    ;;
    *)
    echo "Usage: /etc/init.d/drbd-wait {start|stop}"
    exit 1
    ;;
esac

exit 0
:wq
#
```

Copyright© 2008 LA TIGRE.

② drbd-wait スクリプトをサービスとして登録します。

```
# chmod 755 /etc/rc.d/init.d/drbd-wait
# chkconfig –add drbd-wait
```

初期フル同期が未完了のままセカンダリをリブートした場合などに、初期フル同期が完了するまで、セカンダリのブートシーケンスを待機させます。

これは初期フル同期が未完了の状態で、フェイルオーバが発生し、セカンダリがプライマリに昇格したとしてもファイルシステムをマウントすることができないためです。

2.12. fstab へ登録する際の注意

何らかの理由により、セカンダリとなるサーバの/etc/fstab に DRBD パーティションを登録する場合には、6 番目のオプションには 0 に指定する必要があります。

1 にすると OS のブート時にファイルシステム検査が行われ、/dev/drbd0 を見つけることができず、メンテナンスモードで停止してしまうためです。

2.13. DRBD の起動順序

起動スクリプトは drbd、drbd-wait の順序で起動するよう調整します。

```
# grep chkconfig /etc/rc.d/init.d/drbd
# chkconfig: 345 70 8
# grep chkconfig /etc/rc.d/init.d/drbd-wait
# chkconfig: 345 70 8
```

各ランレベルで S70drbd、S70drbd-wait といったシンボリックリンクが作成されているはずですが、開始優先度(数値昇順)によりスクリプトは実行されていきます。等しい場合(70)はスクリプト名の辞書順で実行されるでしょう。

Copyright© 2008 LA TIGRE.

3. Heartbeat 導入・設定

3.1. Heartbeat について

Heartbeatとは、2台構成でアクティブ/スタンバイ型のHAクラスタ機能を実装しているソフトウェアです。Heartbeat は主として共有リソースの引継ぎ、サーバ・サービスの稼働監視の機能を有しています。

Heartbeat の配布物にはバイナリパッケージとソースコードがそれぞれ用意されていますが、本書ではバイナリパッケージにより導入しました。
http://wiki.linux-ha.org/ja/ReleaseRpm_ja?action=AttachFile&do=get&target=heartbeat-2.1.4-1.rhel5.x86_64.RPMS.tar.gz よりダウンロード可能です。

3.2. Heartbeat のインストール

① 依存パッケージをインストールします。

Heartbeat のインストール前に下記のパッケージをインストールします。

```
# yum install PyXML
# yum install libtool-ltdl
```

② Heartbeat のパッケージを展開、インストールします。

```
# cd /usr/local/src
# tar ztvf heartbeat-2.1.4-1.rhel5.x86_64.RPMS.tar.gz
# cd heartbeat-2.1.4-1.rhel5.x86_64.RPMS
# rpm -ivh heartbeat-2.1.4-1.x86_64.rpm
# rpm -ivh pils-2.1.4-1.x86_64.rpm
# rpm -ivh stonith-2.1.4-1.x86_64.rpm
```

3.3. ハートビート用通信許可

前章の「1.5. ネットワーク構成」で示したように、ハートビート帯域として 10.0.13.0/24、および 10.1.13.0/24 を確保しています。

この帯域内では udp/694 を使用して通信を行いますので、ファイアウォール(iptables)を動作させている場合は、上述のポートを開放します。

Copyright© 2008 LA TIGRE.

3.4. Heartbeat の設定

各種設定ファイルを編集します。設定に関する詳細な情報は下表のリファレンスを参照してください。

対象設定ファイル	リファレンス
authkeys	http://moin.linux-ha.org/lha/authkeys?highlight=%28authkeys%29
ha.cf	http://moin.linux-ha.org/lha/ha.cf?highlight=%28ha.cf%29
haresources	http://moin.linux-ha.org/lha/haresources?highlight=%28haresources%29

Heartbeat ではバージョン 1 系とバージョン 2 系で設定方法に差異があり、バージョン 2 系でもバージョン 1 系の設定方法について後方互換性があります。本書ではバージョン 1 系の設定について記述しています。

① 設定ファイルのサンプルをコピーします。

```
# cd /etc/ha.d
# cp /usr/share/doc/heartbeat-2.1.4/authkeys .
# cp /usr/share/doc/heartbeat-2.1.4/ha.cf .
# cp /usr/share/doc/heartbeat-2.1.4/haresources .
```

② authkeys ファイルを編集します。

本書では、ハートビート通信の認証は CRC による整合性検査に留めており、同じ設定を稼動系/待機系に対して行います。

```
1 auth 1
2 1 crc
3 #2 sha1 HI!
4 #3 md5 Hello!
```

③ ha.cf ファイルを編集します。

稼動系/待機系で一部異なる箇所があります。

	稼動系(web01)	待機系(web02)
1	debugfile /var/log/ha-debug	debugfile /var/log/ha-debug
2	logfile /var/log/ha-log	logfile /var/log/ha-log
3	logfacility local0	logfacility local0
4	keepalive 2	keepalive 2
5	deadtime 30	deadtime 30
6	warntime 10	warntime 10
7	initdead 120	initdead 120
8	udpport 694	udpport 694
9	ucast eth2 10.1.13.3	ucast eth2 10.1.13.2
10	ucast bond0 10.0.13.3	ucast bond0 10.0.13.2
11	auto_failback off	auto_failback off
12	watchdog /dev/watchdog	watchdog /dev/watchdog
13	node web01	node web01
14	node web02	node web02
15	ping 10.0.254.254	ping 10.0.254.254
16	ping 10.1.254.254	ping 10.1.254.254
17	respawn hacluster /usr/lib64/heartbeat/ipfail	respawn hacluster /usr/lib64/heartbeat/ipfail
18	apiauth ipfail gid=haclient uid=hacluster	
19	respawn root /usr/local/cluster/check primary	

パラメータ	セマンティクス
debugfile	デバッグメッセージの出力ファイルを設定する。
logfile	デバッグメッセージを除いた全てのメッセージの出力ファイルを設定する。
logfacility	syslog のロギングファシリティを設定する。 本書では syslog 経由のロギングを行わない。

Copyright© 2008 LA TIGRE.

パラメータ	セマンティクス
keepalive	ハートビートパケットの送信間隔。
deadtime	障害検出時間。 Heartbeat がクラスタのノードが停止したと判断する時間を設定する。
warntime	ハートビート遅延警告の時間。
Initdead	Heartbeat が起動した際にクラスタのノードが停止したと判断する時間を設定する。 ※サーバをリブートした際の通信初期化過程を考慮したパラメータ。
udpport	ハートビートパケットの送受信用ポートを設定する。
ucast	ハートビートパケットの送信先を設定する。 このパラメータはハートビート通信を行う対向ノードについて記述する。
auto_failback	フェイルバックポリシを設定する。 本書では、自動ファイルバックを無効にする。
watchdog	Watchdog デバイスファイル名を設定する。 Heartbeat プロセスを Watchdog(Linux カーネルのモジュール)に監視させ、Heartbeat 自身に障害が発生した場合には OS ごと再起動を行う。
node	Heartbeat クラスタのメンバを設定する。 ノード名は uname –n で得られる正規名でなくてはならない。
ping	仮想的なクラスタメンバを設定する。 このパラメータの意義は上流ネットワークの疎通性確認にあり、メンバノード間だけではなく上流ネットワーク不達の場合にもフェイルオーバするように指示することである。 このパラメータに設定されたノードに対して ping を実行し、その応答性を確認する。 本書では上流ネットワークに位置するゲートウェイのインタフェースに対して ping を実行する。
respawn	Heartbeat の起動に伴い、他の監視プログラムも実行する。 ipfail は前述の ping パラメータで指定されたノードに対して ping を実行し、応答がなければフェイルオーバを実施する。 check_primary はプライマリ側で動作する MySQL の稼動監視を行う。
apiauth	API 認証命令。 特定の API グループに接続するユーザ/グループを設定する。

④ haresources ファイルを編集します。

同じ設定を稼動系/待機系に対して行います。

```
1  web01  IPaddr::10.1.13.10/24/eth2/10.1.13.255      drbddisk::r0     Filesystem::/dev/drbd0::/drbd::ext3     mysql
2  web01  IPaddr::10.0.13.10/24/bond0/10.0.13.255     drbddisk::r0     Filesystem::/dev/drbd0::/drbd::ext3     mysql
```

上記の設定により、web01 を稼動系として仮想 IP アドレスを 2 つ割り当てます(eth2:0 に 10.1.13.10/24、bond0:0 に 10.0.13.10/24)。

また共有リソースの引継ぎ対象として DRBD リソース(MD1000 のストレージ)、および MySQL デーモンを定義します。

3.5. 外部スクリプトの配置

本書ではフェイルバックを手動で行うポリシですので、下記のスクリプトを web01 にのみ用意します。このスクリプトは MySQL のサービス監視、および仮想 IP アドレスの有効性を検査します。

```
web01# cd /usr/local/cluster
web01# vi check_primary
#!/bin/sh
#
# check_primary for MySQL
#
PATH=/bin:/usr/bin:/sbin:/usr/sbin
export PATH
```

Copyright© 2008 LA TIGRE.

```
TOOL=`basename $0`

TOOLDIR=/usr/local/cluster
TMPFILE=/tmp/${TOOL}.$$
HEARTBEAT=/usr/lib64/heartbeat/heartbeat

#################################################################################
# Variables
#################################################################################
INITWAIT=60
INTERVAL=30
RETRYCOUNT=2
RETRYINTERVAL=10

FACILITY=local0
ADMIN="mailaddress"

CVIP=10.1.13.10
MOUNTPOINT=/drbd

MYSQL_USER=username
MYSQL_PASSWD=password

#################################################################################
# Check MySQL service
#################################################################################
checkMysql() {
   N=0
   while true
   do
      isRunning
      if [ $? -ne 0 ]; then
         rm -f ${TMPFILE}
         return 0
      fi

      RESULT=`/usr/bin/mysqladmin -u${MYSQL_USER} -p${MYSQL_PASSWD} ping 2> /dev/null`
      if [ "${RESULT}" = "mysqld is alive" ]; then
         rm -f ${TMPFILE}
         return 0
      fi

      N=`expr ${N} + 1`
      if [ ${N} -gt ${RETRYCOUNT} ]; then
         break
      fi

      sleep ${RETRYINTERVAL}
   done

   rm -f ${TMPFILE}
   return 1
}

#################################################################################
# Check cluster virtual IP address
#################################################################################
checkIPaddr() {
   N=0
   while true
   do
      isRunning
      if [ $? -ne 0 ]; then
         return 0
      fi

      STATUS=`/etc/ha.d/resource.d/IPaddr ${CVIP} status`
      if [ "${STATUS}" = "INFO:  Running OK" ]; then
         return 0
      fi

      N=`expr ${N} + 1`
      if [ ${N} -gt ${RETRYCOUNT} ]; then
         break
      fi

      sleep ${RETRYINTERVAL}
   done

   return 1
}


#################################################################################
# Running heartbeat ?
#################################################################################
isRunning() {
   /etc/init.d/heartbeat status > /dev/null 2>&1
   RC=$?
   return ${RC}
}

#################################################################################
# Send mail
#################################################################################
sendMail() {
```

```
cat << _EOM_ | sendmail -t
From: CHECK-PRIMARY <check_primary@example.net>
To: ${ADMIN}
Subject: Warning: ${TOOL}: `hostname`

$1
_EOM_
}

################################################################################
# Output syslog
################################################################################
outputLog() {
    logger -p ${FACILITY}.error -t ${TOOL} "$1"
    sendMail "$1"
}

################################################################################
# Main
################################################################################
# Wait
sleep ${INITWAIT}

# loop
while true
do
    # Check cluster virtual IP address
    checkIPaddr
    if [ $? -ne 0 ]; then
        outputLog "No IP address for cluster: ${CVIP}"
        ${HEARTBEAT} -k
        exit 100
    fi

    # Check MySQL
    checkMysql
    if [ $? -ne 0 ]; then
        outputLog "Failed to MySQL service"
        ${HEARTBEAT} -k
        exit 100
    fi

    sleep ${INTERVAL}
done
:wq
#
# chmod 755 /usr/local/cluster/check_primary
```

3.6. MySQL の設定

稼動系/待機系に対して同様の設定を行います。

① 自動起動の解除

MySQL のアクティビティは Heartbeat が管理するため、自動起動を解除しておきます。

```
# chkconfig –del mysql
```

② my.cnf の編集

MySQL のデータファイル、ソケットファイル群を DRBD パーティションに配置します。

```
# vi /etc/my.cnf
  ...
[client]
socket                      =  /drbd/lib/mysql/mysql.sock
  ...
[mysqld]
datadir                     =  /drbd/lib/mysql
socket                      =  /drbd/lib/mysql/mysql.sock
  ...
innodb_data_home_dir        =  /drbd/lib/innodb/
innodb_log_group_home_dir   =  /drbd/log/mysql/innLog/
  ...
```

Copyright© 2008 LA TIGRE.

3.7. Heartbeat の起動

① 事前に稼動系/待機系の MySQL を停止しておきます。

```
# /etc/rc.d/init.d/mysql stop
```

② 稼動系の Heartbeat を起動します。

```
web01# /etc/rc.d/init.d/heartbeat start
```

ログファイルには下記のようなログが出力されます。

```
web01# tail -f /var/log/ha-log
ResourceManager[12292]:  2009/03/15_15:10:32 info: Running /etc/ha.d/resource.d/IPaddr
10.0.13.10/24 start
IPaddr[12299]:   2009/03/13_15:10:32 INFO: Using calculated nic for 10.0.13.10: bond0
IPaddr[12299]:   2007/09/13_15:10:32 INFO: Using calculated netmask for 10.0.13.10: 255.255.255.0
IPaddr[12299]:   2007/09/13_15:10:33 INFO: eval ifconfig bond0:0 10.0.13.10 netmask 255.255.255.0 broadcast 10.0.13.255
IPaddr[12304]:   2007/09/13_15:10:33 INFO: Success
ResourceManager[12305]:  2009/03/15_15:10:34 info: Running /etc/ha.d/resource.d/IPaddr 10.1.13.10/24 start
IPaddr[12311]:   2009/03/13_15:10:34 INFO: Using calculated nic for 10.1.13.10: eth2
IPaddr[12311]:   2007/09/13_15:10:34 INFO: Using calculated netmask for 10.1.13.10: 255.255.255.0
IPaddr[12311]:   2007/09/13_15:10:35 INFO: eval ifconfig eth2:0 10.1.13.10 netmask 255.255.255.0 broadcast 10.1.13.255
IPaddr[12316]:   2007/09/13_15:10:35 INFO: Success
```

③ 待機系の Heartbeat を起動します。

```
web02# /etc/rc.d/init.d/heartbeat start
```

3.8. Heartbeat の動作確認

① 稼動系のプロセス、インタフェース情報を確認します。

以下のように DRBD、Heartbeat、MySQL の関連プロセスが起動されています。

```
web01# ps ax
  ...
※DRBDプロセス
11990 ?        S      63:10 [drbd0_asender]
11998 ?        S      30:55 [drbd0_receiver]
11991 ?        S      42:48 [drbd0_worker]
※Heartbeat プロセス
12101 ?        S      0:00 ha_logd: read process
12102 ?        S      0:00 ha_logd: write process
12255 ?        SLs    0:14 heartbeat: master control process
12258 ?        SL     0:00 heartbeat: FIFO reader
12259 ?        SL     0:00 heartbeat: write: ucast eth2
12260 ?        SL     0:39 heartbeat: read: ucast eth2
12261 ?        SL     0:00 heartbeat: write: ucast bond0
12262 ?        SL     0:38 heartbeat: read: ucast bond0
12263 ?        SL     0:00 heartbeat: write: ping 10.0.254.254
12264 ?        SL     0:20 heartbeat: read: ping 10.0.254.254
12265 ?        SL     0:00 heartbeat: write: ping 10.1.254.254
12266 ?        SL     0:21 heartbeat: read: ping 10.1.254.254
12285 ?        S      0:00 /usr/lib64/heartbeat/ipfail
※MySQL プロセス
12965 ?        S      0:00 /bin/sh /usr/bin/mysqld_safe --datadir=/drbd/lib/mysql
 --pid-file=/drbd/lib/mysql/web01.pid
13143 ?        Sl     479:10 /usr/sbin/mysqld --basedir=/ --datadir=/drbd/lib/mysql --user=mysql
--log-error=/var/log/mysqld.log --pid-file=/drbd/lib/mysql/web01.pid -socket=/drbd/lib/mysql/
mysql.sock --port=3306
  ...
```

以下のように bond0、eth2 のインタフェースに IP エイリアスによる仮想 IP アドレスが付与されています。

```
web01# ifconfig
bond0     Link encap:Ethernet  HWaddr 00:22:19:B4:7A:2A
          inet addr:10.0.13.2  Bcast:10.0.13.255  Mask:255.255.255.0
          inet6 addr: fe80::222:19ff:feb4:7a2a/64 Scope:Link
          UP BROADCAST RUNNING MASTER MULTICAST  MTU:1500  Metric:1
          RX packets:261560073 errors:0 dropped:0 overruns:0 frame:0
          TX packets:191303644 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:228020575747 (212.3 GiB)  TX bytes:100155914630 (93.2 GiB)

bond0:0   Link encap:Ethernet  HWaddr 00:22:19:B4:7A:2A
          inet addr:10.0.13.10  Bcast:10.0.13.255  Mask:255.255.255.0
```

Copyright© 2008 LA TIGRE.

```
          UP BROADCAST RUNNING MASTER MULTICAST  MTU:1500  Metric:1
  ...
  ...
eth2      Link encap:Ethernet  HWaddr 00:15:17:A7:CE:9C
          inet addr:10.1.13.2  Bcast:10.1.13.255  Mask:255.255.255.0
          inet6 addr: fe80::215:17ff:fea7:ce9c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:3495217537 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5029916172 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:274773565428 (255.9 GiB)  TX bytes:7477848298458 (6.8 TiB)
          Memory:d5ee0000-d5f00000

eth2:0    Link encap:Ethernet  HWaddr 00:15:17:A7:CE:9C
          inet addr:10.1.13.10  Bcast:10.1.13.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Memory:d5ee0000-d5f00000

eth3      Link encap:Ethernet  HWaddr 00:15:17:A7:CE:9D
          inet addr:172.16.13.2  Bcast:172.16.13.255  Mask:255.255.255.0
          inet6 addr: fe80::215:17ff:fea7:ce9d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:657504777 errors:0 dropped:0 overruns:0 frame:0
          TX packets:903642082 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:52833721075 (49.2 GiB)  TX bytes:967787034784 (901.3 GiB)
          Memory:d5ea0000-d5ec0000
  ...
```

この時点で 10.0.13.10、10.1.13.10 の仮想 IP アドレスに対してブラウザや ssh などでアクセスすると web01 へ接続します。

② 待機系のプロセス、インタフェース情報を確認します。
待機系では MySQL プロセスが起動していないこと、インタフェースに仮想 IP アドレスが付与されていないことを確認してください。

③ 稼動系の Heartbeat を停止します。

```
web01# /etc/rc.d/init.d/heartbeat stop
```

待機系に共有リソース、仮想 IP アドレスが引き継がれます。
下記はフェイルオーバした際のログ内容です。

```
web02# tail -f /var/log/ha-log
ResourceManager[13292]:  2009/03/15_15:16:46 info: Running /etc/ha.d/resource.d/IPaddr 10.0.13.10/16 start
IPaddr[13338]:  2009/03/15_15:16:46 INFO: Using calculated nic for 10.0.13.10: bond0
IPaddr[13338]:  2009/03/15_15:16:46 INFO: Using calculated netmask for 10.0.13.10: 255.255.255.0
IPaddr[13338]:  2009/03/15_15:16:46 INFO: eval ifconfig bond0:0 10.0.13.10 netmask 255.255.255.0 broadcast 10.0.13.255
IPaddr[13345]:  2009/03/15_15:16:46 INFO: Success
mach_down[13346]:   2009/03/15_15:16:46 info: /usr/share/heartbeat/mach_down: nice_failback: foreign resources acquired
heartbeat[12122]: 2009/03/15_23:15:46 info: mach_down takeover complete.
ResourceManager[13401]:  2009/03/15_15:16:47 info: Running /etc/ha.d/resource.d/IPaddr 10.1.13.10/16 start
IPaddr[13409]:  2009/03/15_15:16:47 INFO: Using calculated nic for 10.1.13.10: eth2
IPaddr[13409]:  2009/03/15_15:16:47 INFO: Using calculated netmask for 10.1.13.10: 255.255.255.0
IPaddr[13409]:  2009/03/15_15:16:47 INFO: eval ifconfig eth2:0 10.1.13.10 netmask 255.255.255.0 broadcast 10.1.13.255
IPaddr[13414]:  2009/03/15_15:16:47 INFO: Success
mach_down[131420]:  2009/03/15_15:16:47 info: /usr/share/heartbeat/mach_down: nice_failback: foreign resources acquired
heartbeat[12122]: 2009/03/15_23:15:47 info: mach_down takeover complete.
```

④ フェイルオーバ後の稼動系(web02)のプロセス、インタフェース情報を確認してください。
①のように MySQL の起動プロセス、およびインタフェースへの仮想 IP アドレスが付与されていることが確認できます。

この時点で 10.0.13.10、10.1.13.10 の仮想 IP アドレスに対してブラウザや ssh などでアクセスすると web02 へ接続します。

Copyright© 2008 LA TIGRE.

3.9. Heartbeat の実行順序

前述したDRBD起動スクリプトの実行順序に加えてdrbd、drbd-wait、heartbeatの順序で起動するよう調整します。

```
# grep chkconfig /etc/rc.d/init.d/drbd
# chkconfig: 345 70 8
# grep chkconfig /etc/rc.d/init.d/drbd-wait
# chkconfig: 345 70 8
# grep chkconfig /etc/rc.d/init.d/heartbeat
# chkconfig: 2345 75 05
```

各ランレベルで S70drbd、S70drbd-wait、S75heartbeat といったシンボリックリンクが作成されているはずですが、開始優先度(数値昇順)によりスクリプトは実行されていきます。等しい場合(70)はスクリプト名の辞書順で実行されるでしょう。

終了優先度についても同様に、K05heartbeat、K70drbd、K70drbd-wait の順序でスクリプトが実行されます。

Copyright© 2008 LA TIGRE.

4. 障害試験の確認

4.1. プロセス障害

試験方法の内容とその結果は下表のとおりです。

サーバ	障害箇所	障害意図	実施内容	結果
web01	mysqld	MySQL プロセス障害	/etc/rc.d/init.d/mysql stop	待機系(web02)へ F/O
web01	heartbeat	Heartbeat プロセス障害	/etc/rc.d/init.d/heartbeat stop	待機系(web02)へ F/O

4.2. ネットワーク障害

試験方法の内容とその結果は下表のとおりです。

サーバ	障害箇所	障害意図	実施内容	結果
web01	bond0	公開 N/W セグメント 通信障害	ifconfig bond0 down	待機系(web02)へ F/O
web01	eth2	保護 N/W セグメント 通信障害(DB 帯域)	ifconfig eth2 down	待機系(web02)へ F/O
web01	eth2	保護 N/W セグメント 通信障害(DRBD 帯域)	ifconfig eth3 down	F/O は発生せず、ミラーリングが停止。両ノードで WFConnection 状態となり、eth3 が復旧するとミラーリングが再開される。
web01	全て	全 N/W セグメント 通信障害	/etc/rc.d/init.d/network stop	F/O は発生せず、スプリットブレインが発生。 両ノード、または片側ノードで StandAlone 状態となった。

4.3. ディスク障害

この試験は未実施です。下記は試験方法とその結果予測です。

① 稼動系(web01)に割り当てた仮想ディスクを構成している HDD の 1 つをオフラインにします。

② 仮想ディスクの RAID レベルは 0 ですので、ディスク I/O エラーが発生すると考えられます。

③ drbd.conf により、on-io-error に detach を指定しているため、DRBD はディスクレスモードで動作しようとします。

④ Heartbeat ではディスク障害について直接的な監視を行っていませんのでフェイルオーバは発生しません。別途ディスク監視をトリガとした手動フェイルオーバが必要と考えられます。

Copyright© 2008 LA TIGRE.

5. 運用

5.1. メンテナンス時

計画保守(カーネルバージョンアップ等)の場合、安全性を確保するために下記の手順で行います(web01 をプライマリ/稼動系、web02 をセカンダリ/待機系とする場合)。

① web02 に対してバージョンアップ作業を実施した後、OS 再起動。
② web02 の HA 状態がセカンダリ(待機系)のままであることを確認。
③ web01 の Heartbeat、または MySQL を停止する(手動フェイルオーバ)。
④ web02 の HA 状態がプライマリ(稼動系)に変化し、共有リソースや仮想 IP アドレスが引き継がれたことを確認。
⑤ web01 に対してバージョンアップ作業を実施した後、OS 再起動。
⑥ web01 の HA 状態がセカンダリ(待機系)のままであることを確認。

これ以降は任意ですが、フェイルバックする場合には下記の手順で行います。
⑦ web02 の Heartbeat、または MySQL を停止する(手動フェイルバック)。
⑧ web01 の HA 状態がプライマリ(稼動系)に変化し、共有リソースや仮想 IP アドレスが引き継がれたことを確認。

5.2. スプリットブレイン発生時

スプリットブレインとは DRBD を構成しているノード両方がプライマリと誤認識され、記憶領域の分裂が発生する事象を指します。

この状態では両ノードの接続状態が StandAlone に遷移し、ディスクのミラーリングは行われず、元々プライマリであったノード上でのみディスクの読み書きが可能です。

手動で復旧させる場合には、下記の手順で行ってみてください。

① プライマリノードを決定する。
スプリットブレインが発生した場合、ディスクのミラーリングは行われませんので当然両ノードのディスク内容は異なっています。

通常、プライマリとなって動作している DRBD パーティションがマウント可能なため、最新のデータはプライマリに存在すると考えられます。

Copyright© 2008 LA TIGRE.

以下のように ro:Primary/…の出力があれば、当該ノードがプライマリとして動作しています。

```
# cat /proc/drbd
version: 8.3.0 (api:88/proto:86-89)
GIT-hash: 9ba8b93e24d842f0dd3fb1f9b90e29 build by admin@web01, 2009-03-01 15:10:34
 0: cs:StandAlone ro:Primary/Unknown ds: UpToDate/Inconsistent C r---
#
```

② プライマリで DRBD 同期コネクションを試行する。

```
# drbdadm connect r0
```

接続状態が WFConnection の場合、いったん接続を解除する必要があるかも知れません。

```
# drbdadm disconnect r0
# drbdadm connect r0
```

③ セカンダリで DRBD 同期コネクションを試行する。

最初に明示的なセカンダリ降格を指示して後、プライマリと同期させます。

```
# drbdadm secondary r0
# drbdadm -- --discard-my-data connect r0
```

5.3. 障害時の切り分け観点

これは状況により対応が変化しますが、下記の切り分け観点に基づいてトラブルシュートしてください。

① NIC の状態を確認する。

ifconfig コマンドにより仮想 IP アドレスが付与されているか確認してみてください。

② 物理ディスクの正常性を確認する。

いったん両ノードの Heartbeat を停止します。

各ノードの/proc/drbd を確認し、プライマリノードでマウント可能かどうかを確認してください。プライマリノードのディスク障害の場合には、プライマリ、セカンダリの役割を変更する必要があるかも知れません。

Copyright© 2008 LA TIGRE.

HA cluster construction memo(DRBD & Heartbeat)

改版履歴

Version 0.1	2009/04/13	新規作成。
Version 0.2	2009/06/02	誤字修正。

製作

LA TIGRE

本書は 2009 年 6 月現在の情報を元に作成されております。本書に記載されております内容は、許可なく変更されることがあります。

Copyright© 2008 LA TIGRE.