
bonding guide for RHEL5

RHEL5 bonding ドライバ利用ガイド

Version 0.3

Copyright © 2009 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。したがって、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
1.1. はじめに.....	1
1.2. ネットワーク環境.....	1
1.3. サーバ構成	2
1.4. 参考文献.....	2
2. bonding ドライバについて.....	3
2.1. 利用要件	3
2.2. 動作モード	3
2.3. リンク障害の監視	4
2.4. ARP テーブルの更新	5
2.5. MAC テーブルの更新.....	5
3. 設定方法	6
3.1. 物理ネットワークの構成.....	6
3.2. モジュール設定	7
3.3. インタフェース設定.....	7
3.4. 設定の反映	8
3.5. 耐障害性の確認	10
3.6. スイッチ障害の対策	12
3.7. 物理アダプタの手動切替え	14

1. 概要

1.1. はじめに

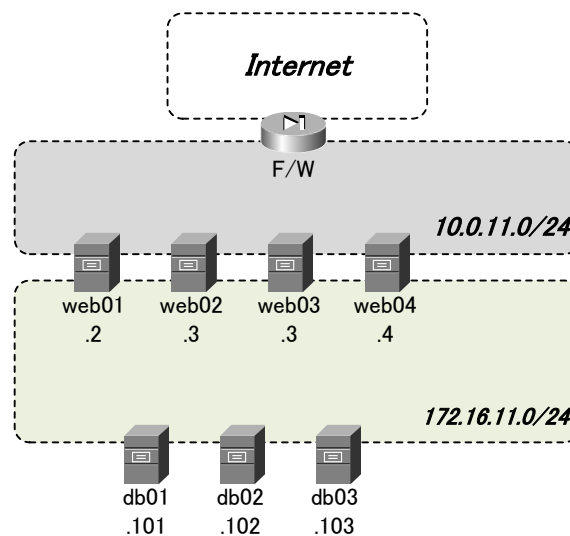
本書は Linux の bonding ドライバの利用情報等について記述したものです。

サーバの耐障害性を高める具体的な方法として各種のハードウェアを冗長化する試みがありますが、本書では NIC の冗長化について焦点を合せています。

NIC の冗長化は一般的にチーミング(teaming)と呼ばれますが、Linux では bonding ドライバを使用して設定を行い、複数の物理イーサネットアダプタ上に論理的なアダプタを構成し、イーサネットアダプタの冗長化を実現します。

1.2. ネットワーク環境

下図のサーバ群が、ネットワーク上に配置されています。これらの環境下でチーミングを行います。また、サーバ筐体は Dell PowerEdge 1950、スイッチ筐体は Dell PowerConnect 5324 によって構成されています。



1.3. サーバ構成

全サーバともプラットフォームは **RedHat Enterprise Linux Server release 5.2**(64ビット)、カーネルのバージョンは **2.6.18-92.el5** となります。各サーバの機能概要は下表のとおりです。

ノード名	機能
F/W	ルータ + ファイアウォール。
web01	Web サーバ。
web02	
web03	
web04	
db01	データベースサーバ。
db02	
db03	

1.4. 参考文献

下記の文書も併せてご参照ください。

文書名	リンク
Bonding ドライバ利用ガイド (RHEL3, RHEL4)	http://www-06.ibm.com/jp/domino01/mkt/cnpages7.nsf/page/default-0046E089
Bonding ドライバ利用ガイド for RHEL5	http://www-06.ibm.com/jp/domino01/mkt/cnpages7.nsf/page/default-000C97B9
bonding 機能紹介と展望	http://osdn.jp/event/kernel2005/pdf/nec.pdf
Linux イーサネット結合ドライバミニ Howto	http://www.linux.or.jp/JF/JFdocs/kernel-docs-2.4/networking/bonding.txt

2. bonding ドライバについて

2.1. 利用要件

- RedHat Enterprise Linux 3、および RedHat Enterprise Linux 4 では標準機能として組み込まれており、特別なパッケージを導入することなく利用することができます。
- bonding ドライバの利用にはカーネル側での対応が必要とされますが、カーネルバージョン 2.4.18 より標準で利用可能です。
- 利用可能なイーサネットカードは、Linux で動作可能な NIC であればメーカ、バージョン、帯域差異等、混在していても問題ありません。

2.2. 動作モード

bonding ドライバは以下の動作モードを有しています。

モード	動作内容
mode 0 balance-rr	送信時に選択される物理アダプタはラウンドロビンで使用(1 パケット毎に)。 受信時に選択される物理アダプタはスイッチ側の判断による。 ※トラッキングをサポートするスイッチと接続する。
mode 1 active-backup	複数の物理アダプタを稼働系・待機系として使用する。 稼働系アダプタの障害時には待機系ポートへ切り替える。
mode 2 balance-xor	送信時に選択される物理アダプタは送信先の MAC アドレスをキーとするハッシングで使用。 受信時に選択される物理アダプタはスイッチ側の判断による。 ※トラッキングをサポートするスイッチと接続する。
mode 3 broadcast	すべての物理アダプタからパケットを送出する。
mode 4 802.3ad	IEEE 802.3ad に準ずるリンク・アグリゲーションを行う。 ※IEEE 802.3ad をサポートするスイッチと接続する。
mode 5 balance-tlb	送信時に選択される物理アダプタは速度、負荷に応じて使用。 受信時に選択される物理アダプタは稼働系ポート(固定)を使用し、障害時には他のポートに切り替える。
mode 6 balance-alb	送信・受信時に選択される物理アダプタは速度、負荷に応じて使用。 ※受信時は通信相手の IP アドレスに対応した固定のアダプタを使用。

以降、本書では **active-backup モード** に特化した記述を行っています。

2.3. リンク障害の監視

bonding ドライバは物理アダプタの障害、または物理アダプタに接続されたスイッチとの接続が切れた場合など、物理アダプタがパケットを受信することができない状態を監視する機能があります。

リンク障害の監視方法には以下の種類があります。

- **MII リンク監視**

MII(Media Independent Interface)規格のリンク情報を利用します。

この場合、物理アダプタが **MII** に対応している必要があります。

【対応している場合】

```
# ethtool eth0
Settings for eth0:
Supported ports: [ TP ]
    Supported link modes:   10baseT/Half 10baseT/Full
                           100baseT/Half 100baseT/Full
                           1000baseT/Full
    Supports auto-negotiation: Yes
    Advertised link modes:  10baseT/Half 10baseT/Full
                           100baseT/Half 100baseT/Full
                           1000baseT/Full
    Advertised auto-negotiation: Yes
    Speed: 1000Mb/s
    Duplex: Full
    Port: Twisted Pair
    PHYAD: 0
    Transceiver: internal
    Auto-negotiation: on
    Supports Wake-on: umbg
    Wake-on: d
    Current message level: 0x00000007 (7)
    Link detected: yes
#
```

【対応していない場合】

```
# ethtool eth0
Settings for eth0:
No data available
#
```

- **ARP 監視**

物理アダプタの受信パケットカウントによってリンク状態を確認します。

物理アダプタが受信するパケットを強制的に生成するため、事前に定義した IP アドレスへ **ARP** リクエストを送信します。

基本的には **MII** リンク監視を使用し、物理アダプタのドライバが **MII** 規格に対応していない場合に代替手段として **ARP** 監視を使用します。

2.4. ARP テーブルの更新

active-backup モードでは、全ての物理アダプタ、および論理アダプタに同一の論理 MAC アドレスが設定されます。

通常、最初に認識した物理アダプタの物理 MAC アドレスに統一されます。

```
# /sbin/ifconfig | grep HWaddr
bond0 Link encap:Ethernet HWaddr 00:14:22:XX:YY:ZZ
eth0   Link encap:Ethernet HWaddr 00:14:22:XX:YY:ZZ
eth1   Link encap:Ethernet HWaddr 00:14:22:XX:YY:ZZ
#
```

稼働系アダプタが待機系アダプタに切り替わった場合でも、論理 MAC アドレスは同じため、アクセス元の ARP テーブルを更新する必要はありませんし、実際行われません。

2.5. MAC テーブルの更新

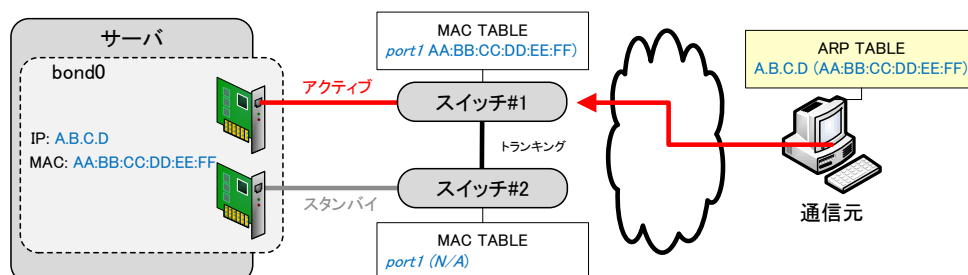
スイッチは各ポートに接続された機器の MAC アドレスを記憶していますが、チーミング設定を行っている場合には、稼働系アダプタの切り替えに伴い、MAC テーブルも更新されなければ通信が再開されません。

この問題には、以下の対処により克服することが可能です。

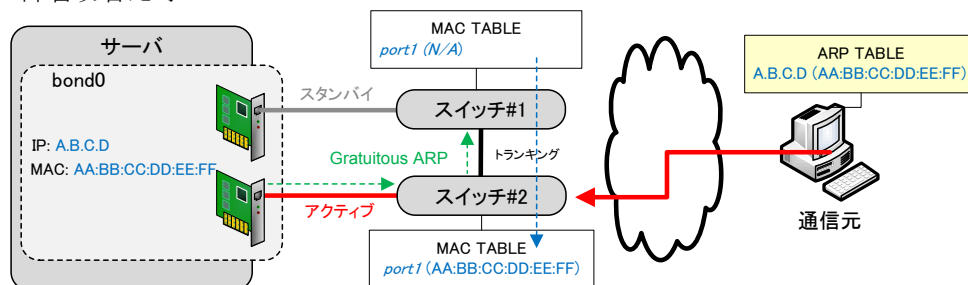
- スwitchの MAC アドレス記憶時間(MAC aging)を短めに設定する。
- Gratuitous ARP を発行する。

本書では RHEL5.2(正確には RHEL4U4 以降)を使用しておりますので、MII リンク監視を使用していますが、稼働系アダプタの切り替え時に Gratuitous ARP を発行します。

- 通常時



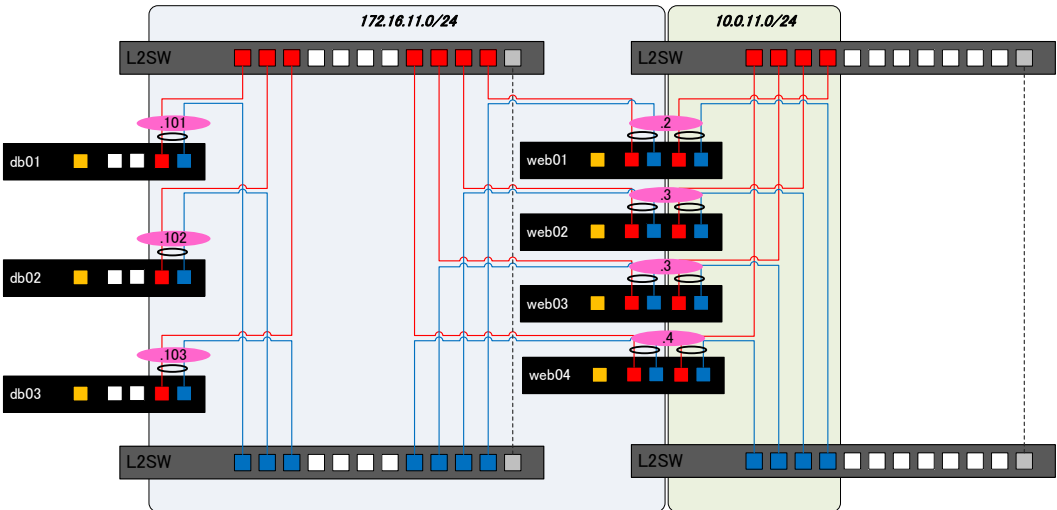
- 障害切替え時



3. 設定方法

3.1. 物理ネットワークの構成

前節「1.2. ネットワーク環境」から詳細な物理結線を下図に示します。



上図では、赤と青の結線が対となってチーミングされていることを表しており、サーバのインタフェースは下記の並びで構成されています。

—	172.16.11.0/24(バックエンド)		10.0.11.0/24(フロントエンド)	
■	■	■	■	■
DRAC 用	eth2	eth3	eth0	eth1
I/F 諸元	I/F 諸元(PCI 接続)		I/F 諸元(オンボード)	
—	Intel Corporation 82571EB Gigabit Ethernet Controller		Broadcom Corporation NetXtreme II BCM5708 Gigabit Ethernet	

上述の構成に基づき、以降チーミング設定について記述していきます。

本章では web01 についての設定方法を記述していますが、他のサーバも IP アドレス等を読み替えて同様に設定してください。

3.2. モジュール設定

物理 NIC のドライバ、および bonding ドライバの設定を確認し、必要に応じて以下の設定を行います。

```
# vi /etc/modules.conf
...
alias eth0 bnx2
alias eth1 bnx2
alias eth2 e1000e
alias eth3 e1000e
alias bond0 bonding
alias bond1 bonding
...
#
```

論理アダプタ名は **bondN** を使用する必要があります。論理アダプタが複数ある場合には、**N** は通し番号として **0** から始めます。

3.3. インタフェース設定

① 10.0.11.2 の論理アダプタの設定を行います。

```
# cd /etc/sysconfig/network-scripts
# vi ifcfg-bond0
DEVICE=bond0
BOOTPROTO=static
ONBOOT=yes
IPADDR=10.0.11.2
NETMASK=255.255.255.0
BONDING_OPTS="mode=1 primary=eth0 miimon=100 updelay=5000"
...
#
```

② 物理アダプタの設定を行います。

```
# cd /etc/sysconfig/network-scripts
# vi ifcfg-eth0
DEVICE=eth0
MASTER=bond0
SLAVE=yes
...
# vi ifcfg-eth1
DEVICE=eth1
MASTER=bond0
SLAVE=yes
...
#
```

上記の設定は、物理アダプタ 2 つ(eth0、eth1)を論理アダプタ bond0 として構成し、かつ通信上の主従関係としてマスタアダプタを bond0、スレーブアダプタをそれぞれ eth0、eth1 と定義しています。

bond0 へ渡すオプションとして BONDING_OPTS には下表にあるパラメータを設定することができます。

パラメータ名	設定説明
Mode	前節「2.2. 動作モード」を参照してください。 ここでは active-backup モードで動作するよう指定しています。
primary	稼働系物理アダプタを指定します。 本書においては、通常は 10.0.110/24 セグメントへの通信時には eth0 を使用し、また 172.16.11.0/24 セグメントへの通信時には eth2 を使用しますが eth0 、 eth2 に障害が発生した場合には、それぞれのスレーブとして eth1 、 eth3 へ通信が切り替わります。 eth1 、 eth3 が復旧した場合、自動的に eth0 、 eth2 へ通信が切り替わります(フェイルバック)。
Miimon	MII リンク監視においてリンク確認の時間間隔をミリ秒で指定します。 ここでは 100 ミリ秒を指定しています。
updelay	MII リンク監視においてリンクアップを検知後に物理アダプタに切り替えるまでの遅延時間をミリ秒で指定します。このパラメータは上位スイッチの状態安定にかかる時間として意味づけられ、 miimon の整数倍を指定する必要があります。
arp_interval	ARP 監視においてリンク確認の時間間隔をミリ秒で指定します。 miimon と同時に指定することはできません。
arp_ip_target	ARP 監視において ARP 要求を送信する IP アドレスを指定します。 複数指定することができ、最大 16 個の IP アドレスを指定可能です。

- ③ 172.16.11.2 の論理アダプタの設定を行います。

```
# cd /etc/sysconfig/network-scripts
# vi ifcfg-bond1
DEVICE=bond1
BOOTPROTO=none
ONBOOT=yes
IPADDR=172.16.11.2
NETMASK=255.255.255.0
BONDING_OPTS="mode=1 primary=eth2 miimon=100 updelay=5000"
...
#
```

- ④ 物理アダプタの設定を行います。

```
# cd /etc/sysconfig/network-scripts
# vi ifcfg-eth2
DEVICE=eth2
MASTER=bond1
SLAVE=yes
...
# vi ifcfg-eth3
DEVICE=eth3
MASTER=bond1
SLAVE=yes
...
#
```

複数の論理アダプタを構成するためには、単に **ifcfg-bond1**、**ifcfg-bond2**、...**ifcfg-bondN** と定義していけばよいだけです。もちろん **if-bondN** に対応する物理アダプタ(**ethM**)が存在していなければならず、かつ物理アダプタの設定(**ifcfg-ethM**)も必要です。

3.4. 設定の反映

下記コマンドを投入し、上述の設定を反映させます。

```
# /etc/rc.d/init.d/network restart
...
Setting network parameters:      [ OK ]
Bringing up loopback interface:  [ OK ]
Bringing up interface bond0:     [ OK ]
```



```
Bringing up interface bond1:      [ OK ]
#
```

上記では、物理アダプタの存在は隠されて、論理アダプタ(bondM)とループバックだけが初期化されていることがわかります。Ifconfig コマンドでアダプタのステータスを確認します。

```
# /sbin/ifconfig
bond0  Link encap:Ethernet  HWaddr 00:1E:C9:xx:yy:zz
        inet addr:10.0.11.2  Bcast:10.0.11.255  Mask:255.255.255.0
        UP BROADCAST RUNNING MASTER MULTICAST  MTU:1500  Metric:1
        RX packets:12521874  errors:0  dropped:0  overruns:0  frame:0
        TX packets:6677472  errors:0  dropped:0  overruns:0  carrier:0
        collisions:0 txqueuelen:0
        RX bytes:1821112205 (1.6 GiB)  TX bytes:2684345064 (2.4 GiB)

bond1  Link encap:Ethernet  HWaddr 00:15:17:ii:jj:kk
        inet addr:172.16.11.2  Bcast:172.16.11.255  Mask:255.255.255.0
        UP BROADCAST RUNNING MASTER MULTICAST  MTU:1500  Metric:1
        RX packets:13902469  errors:0  dropped:0  overruns:0  frame:0
        TX packets:9470298  errors:0  dropped:0  overruns:0  carrier:0
        collisions:0 txqueuelen:0
        RX bytes:7752455013 (7.2 GiB)  TX bytes:1974429153 (1.8 GiB)

eth0   Link encap:Ethernet  HWaddr 00:1E:C9:xx:yy:zz
        UP BROADCAST RUNNING SLAVE MULTICAST  MTU:1500  Metric:1
        RX packets:9585920  errors:0  dropped:0  overruns:0  frame:0
        TX packets:6677450  errors:0  dropped:0  overruns:0  carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:1633186388 (1.5 GiB)  TX bytes:2684339869 (2.4 GiB)
        Interrupt:169  Memory:f8000000-f8012100

eth1   Link encap:Ethernet  HWaddr 00:1E:C9:xx:yy:zz
        UP BROADCAST RUNNING SLAVE MULTICAST  MTU:1500  Metric:1
        RX packets:2935954  errors:0  dropped:0  overruns:0  frame:0
        TX packets:22  errors:0  dropped:0  overruns:0  carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:187925817 (179.2 MiB)  TX bytes:5195 (5.0 KiB)
        Interrupt:169  Memory:f4000000-f4012100

eth2   Link encap:Ethernet  HWaddr 00:15:17:ii:jj:kk
        UP BROADCAST RUNNING SLAVE MULTICAST  MTU:1500  Metric:1
        RX packets:2040202  errors:0  dropped:0  overruns:0  frame:0
        TX packets:82775  errors:0  dropped:0  overruns:0  carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:203371488 (193.9 MiB)  TX bytes:11266089 (10.7 MiB)
        Memory:fc3e0000-fc400000

eth3   Link encap:Ethernet  HWaddr 00:15:17:ii:jj:kk
        UP BROADCAST RUNNING SLAVE MULTICAST  MTU:1500  Metric:1
        RX packets:11862267  errors:0  dropped:0  overruns:0  frame:0
        TX packets:9387523  errors:0  dropped:0  overruns:0  carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:7549083525 (7.0 GiB)  TX bytes:1963163064 (1.8 GiB)
        Memory:fc3a0000-fc3c0000

...
#
```

上記のように出力されていれば、問題なく動作しています。確認観点は以下のとおりです。

- インタフェースのリンク状態
- 論理アダプタ(bondM)に対して IP アドレスが付与されているか
- MASTER と SLAVE の対応
- 論理アダプタ(bondM)と物理アダプタ(ethM)の MAC アドレスが等しい

また、物理アダプタ、および論理アダプタのステータスを確認するのに `proc` ファイルシステムを参照します。`syslog` メッセージ(`/var/log/messages`)でも確認可能です。

```
# cat /proc/net/bond0/info
Ethernet Channel Bonding Driver: v3.2.4 (January 28, 2008)

Bonding Mode: fault-tolerance (active-backup)
Primary Slave: eth0
Currently Active Slave: eth0
MII Status: up
MII Polling Interval (ms): 100
Up Delay (ms): 5000
Down Delay (ms): 0

Slave Interface: eth0
MII Status: up
Link Failure Count: 1
Permanent HW addr: 00:1e:c9:xx:yy:zz

Slave Interface: eth1
MII Status: up
Link Failure Count: 0
Permanent HW addr: 00:1e:c9:ii:jj:kk
#
```

上記は、`bond0` 論理アダプタが、`active-backup` モードで動作していること、稼働系物理アダプタが `eth0` で通信していること、および全てのアダプタがリンクアップしていることを表しています。

3.5. 耐障害性の確認

この節では、通信障害を人為的に引き起こし、稼働系から待機系へ物理アダプタが切り替わった後、通信が継続される様子を追跡していきます。

障害を発生させる前に、継続した通信状態を再現します。

適当な Windows 機から `web01` 機(`172.16.11.2`)へ `ping` を発行し続けている様子を下記に示します。`172.16.11.2` はバックエンド側のセグメントに属しています。

```
C:\> ping 172.16.11.2 -t

Pinging 172.16.2.11 with 32 bytes of data:

Reply from 172.16.11.2: bytes=32 time<1ms TTL=64
Reply from 172.16.11.2: bytes=32 time<1ms TTL=64
Reply from 172.16.11.2: bytes=32 time<1ms TTL=64
...
```

この状態で、通信障害を発生させます。つまり `web01` 機の `eth2` 側に結線されている LAN ケーブルを抜線します。

```
...
Reply from 172.16.11.2: bytes=32 time<1ms TTL=64
Reply from 172.16.11.2: bytes=32 time<1ms TTL=64
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Reply from 172.16.11.2: bytes=32 time<1ms TTL=64
Reply from 172.16.11.2: bytes=32 time<1ms TTL=64
```

```
...
```

MII 監視のリンクアップ遅延時間を 5 秒としていることから、LAN ケーブル抜線時に約 5 秒間の ICMP パケットが不通状態になったと考えられます。

LAN ケーブル抜線時の bonding ドライバのステータスは下記のとおりです。

```
# cat /proc/net/bond1/info
Ethernet Channel Bonding Driver: v3.2.4 (January 28, 2008)

Bonding Mode: fault-tolerance (active-backup)
Primary Slave: eth2
Currently Active Slave: eth3
MII Status: up
MII Polling Interval (ms): 100
Up Delay (ms): 5000
Down Delay (ms): 0

Slave Interface: eth2
MII Status: down
Link Failure Count: 2
Permanent HW addr: 00:15:17:xx:yy:zz

Slave Interface: eth3
MII Status: up
Link Failure Count: 0
Permanent HW addr: 00:15:17:ii:jj:kk
#
```

稼働系物理アダプタ(eth2)から、待機系物理アダプタ(eth3)へフェイルオーバーした事象が報告されています。

続いて web01 機の eth0 側の LAN ケーブルをスイッチへ結線し直します。

```
...
Reply from 172.16.11.2: bytes=32 time<1ms TTL=64
Reply from 172.16.11.2: bytes=32 time<1ms TTL=64
Reply from 172.16.11.2: bytes=32 time<1ms TTL=64
Reply from 172.16.11.2: bytes=32 time<1ms TTL=64
...
```

Bonding ドライバのステータスを確認するとフェイルバックした事が報告され、上記の ping コマンドからはタイムアウトは報告されません。

```
# cat /proc/net/bond0/info
Ethernet Channel Bonding Driver: v3.2.4 (January 28, 2008)

Bonding Mode: fault-tolerance (active-backup)
Primary Slave: eth2
Currently Active Slave: eth2
MII Status: up
MII Polling Interval (ms): 100
Up Delay (ms): 5000
Down Delay (ms): 0

Slave Interface: eth2
MII Status: up
Link Failure Count: 2
Permanent HW addr: 00:1e:c9:xx:yy:zz

Slave Interface: eth3
MII Status: up
```

```
Link Failure Count: 0
Permanent HW addr: 00:1e:c9:ii:jj:kk
#
```

物理アダプタ切り替わりの過程は下記のような **syslog** メッセージで出力されます。

※日付、ホスト名は省略しています。

```
# tail -f /var/log/messages
...

※ケーブル抜線

e1000e: eth2 NIC Link is Down
bonding: bond1: link status definitely up for interface eth2.
bonding: bond1: making interface eth3 the new active one.

※ケーブル結線

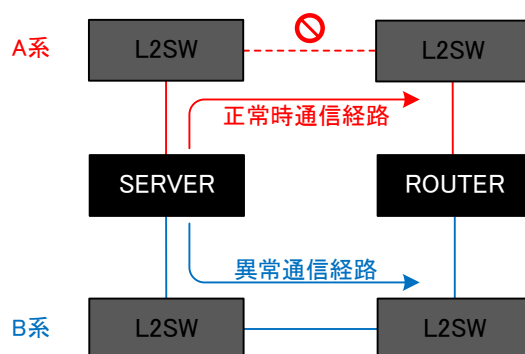
e1000e: eth2 NIC Link is Up, 1000Mbps full duplex, receive & transmit flow
control ON
bonding: bond1: link status up for interface eth2, enabling it in 5000 ms.
bonding: bond1: link status definitely up for interface eth3.
bonding: bond1: making interface eth3 the new active one.
bonding: bond1: first active interface up!
ADDRCONF(NETDEV_CHANGE): bond1: link becomes ready
bonding: bond1: link status definitely up for interface eth2.
```

3.6. スイッチ障害の対策

本章では、リンク障害の監視について **MII** リンク監視による設定を行ってきました。

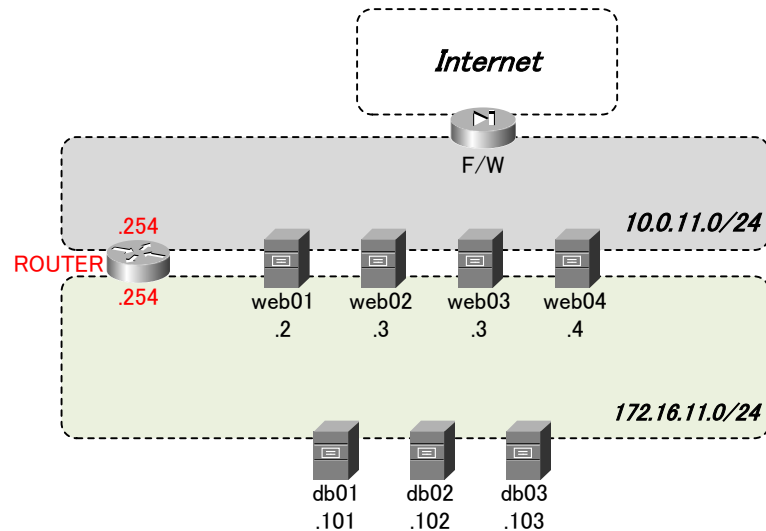
この方法は、サーバ側物理アダプタから直取したスイッチ側物理ポート間の接続断のみ検出対象となっていることに注意が必要です。

スイッチのアップリンクポートが故障するなど、サーバ～スイッチ間の接続性に問題がないが、**IP** 到達性が失われるパターンの障害が発生した場合には **MII** リンク監視は有効な方法となりません(実際に、この現象に遭遇しました)。



上図はネットワーク障害時の概略図ですが、**SERVER** と **A 系 L2SW** 間にリンク障害が発生しなくとも、**A 系** 上位と下位の **L2SW** 間に障害が発生すると、結局通信不可能となってしまいます。

この場合でも、SERVER は B 系 L2SW に向かうように通信させる必要があります。
上流側のネットワークの障害に応じて自動的に通信を迂回させるには、上流側に対する ARP 監視が有効です。



ARP 監視を行う対象は、より上流にある常時稼働状態のノードが適切です。本書の例では、各ネットワークを結ぶ ROUTER に対して ARP 監視を行いました。

論理アダプタの設定を下記のように変更します。

```
# cd /etc/sysconfig/network-scripts
# vi ifcfg-bond1
DEVICE=bond1
BOOTPROTO=none
ONBOOT=yes
IPADDR=172.16.11.2
NETMASK=255.255.255.0
BONDING_OPTS="mode=1 primary=eth2 arp_interval=1000 arp_ip_target=+172.16.254.254"
...
```

arp_ip_target に複数の対象を登録する場合には、arp_ip_target=+172.16.254.254
arp_ip_target=+172.16.254.253 ...と記述します。

BONDING_OPTS パラメータを変更した場合には、サーバの再起動か bonding モジュールの再ロードが必要です。モジュールの再ロードは下記のように一時ネットワークサービスを停止してから bonding モジュールを削除してください。

```
# /etc/rc.d/init.d/network stop
# /sbin/rmmod bonding
# /etc/rc.d/init.d/network start
```

3.7. 物理アダプタの手動切替え

本来であれば、物理 NIC の片側に通信障害が発生すれば自動的に切替りますが、何らかの理由により、自動切替えが行われない場合も考えられます。

その場合、下記のように手動で切替えます。

```
# /sbin/ifenslave -c bond1 eth3
```

上記は **eth2** がアクティブアダプタの状態で通信障害が発生した場合、論理アダプタ **bond1** のアクティブアダプタを **eth3** に切替える方法を示しています。

bonding guide for RHEL5

改版履歴

Version 0.1	2008/04/18	新規作成。
Version 0.2	2008/10/31	サービス用に展開した内容に改定。
Version 0.3	2009/03/06	下記の節を追加。 「3.6. スイッチ障害の対策」 「3.7. 物理アダプタの手動切替え」

製作

LA TIGRE

本書は 2009 年 3 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。