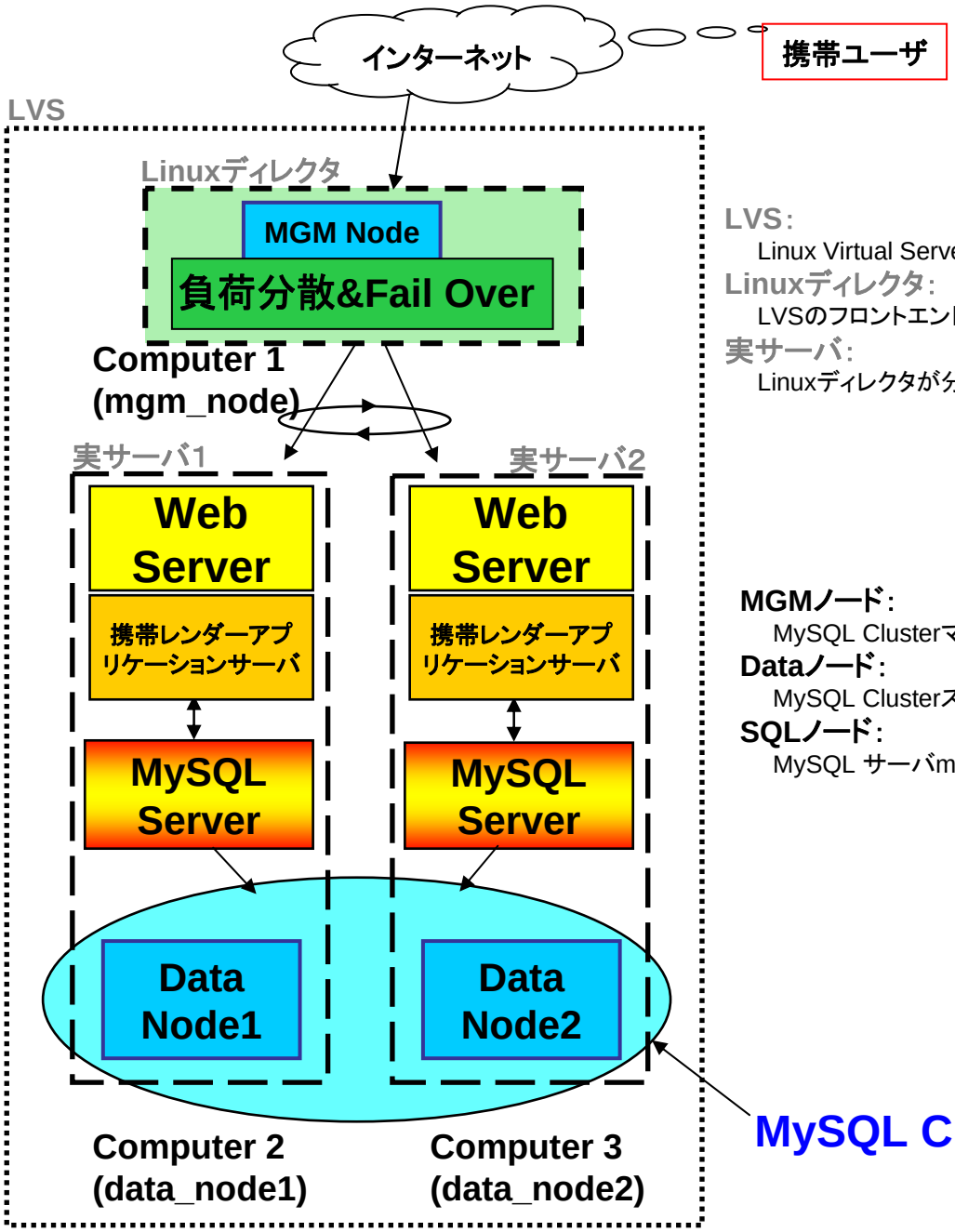
# Linux Virtual Server による MySQL Cluster対応Webアプリケーションサーバの負荷分散

2006年9月18日

Made in Jul.12 2006 at Tronto
OSC2006-Hokkaido in Jul.15 2006
OSRI Technical Seminar 2006 Edition

**桑村　潤**




**LVS:**
Linux Virtual Server による仮想クラスタサーバ。

**Linuxディレクタ:**
LVSのフロントエンドで負荷を分散させるサーバ。

**実サーバ:**
Linuxディレクタが分散したパケットを実際に処理するサーバ。

**MGMノード:**
MySQL Clusterマネージャーndb_mgmdを起動させるサーバ。

**Dataノード:**
MySQL Clusterストレージエンジンndbdを起動させるサーバ。

**SQLノード:**
MySQL サーバmysqldを起動させるサーバ。

# クラスタのタイプ

- 並列演算クラスタ
  - Beowulf クラスタ型。並列処理プログラム
- 高可用性クラスタ
  - 冗長化構成。フォールトトレラントコンピュータ
- 負荷分散クラスタ
  - 単一サービスの分散実行。ロードバランサ

# MySQL Cluster

- オープンソース開発モデル（２重ライセンス）
- **NDB Cluster**をストレージエンジンに使用
- 非共有型並列分散RDBMS
- 負荷分散と高可用性を併せ持つ
- インメモリデータベース
- 参照

  http://k.hirohama.biz/wiki/index.php/MySQL_Cluster

# NDB Cluster

- Ericsson Business Innovationのベンチャ Alzato社が開発
  - Alzatoは通信業界向けデータ管理ソフトウェアベンダー
  - AlzatoをMySQL ABが2003年に買収
- オープンアーキテクチャ互換API
- 高品質のデータ管理システム
- 高可用性と高スループット

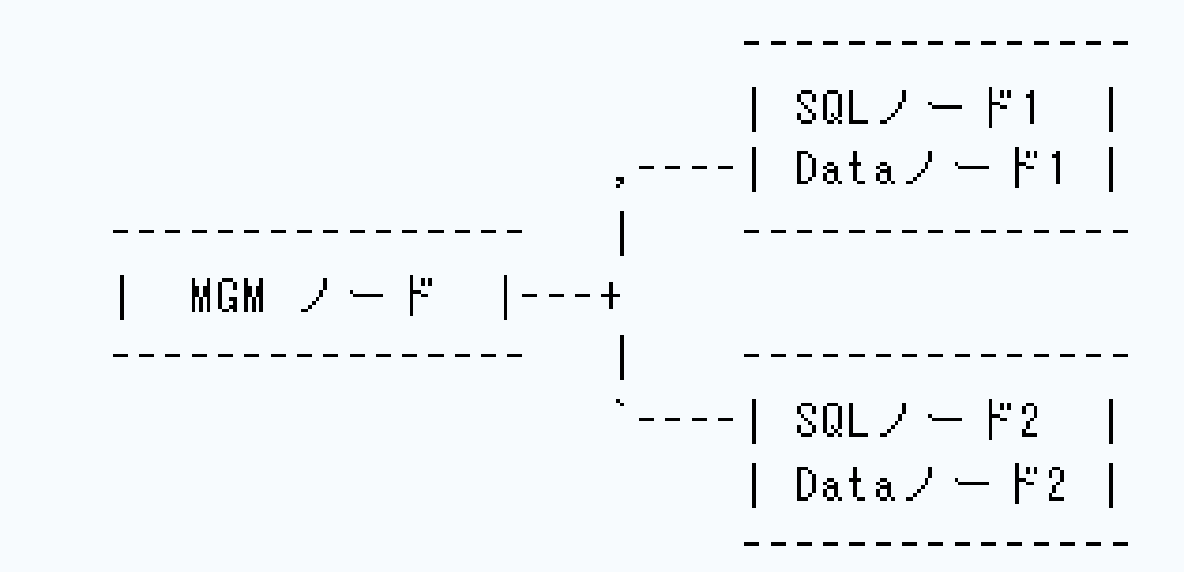
# MySQL Clusterシステム 構築の注意点

- 均一な構成要素の実現
- 負荷分散と高可用性の実現
- スケーラビリティの実現
- アプリケーションの配備システム
- バックアップ
- 参照

  Software Design 2006年9月号 LAMP特集 第5章 「LVS+MySQL Clusterで構築する スケーラブルLAMPシステムの構築と運用」（Acorn Software Technologies、オープンソース総合研究所）参照

# MySQl Clusterの構成

- 最小構成




MGMノード: MySQL Clusterマネージャーndb_mgmdを起動させるNDB管理サーバ。

Dataノード:MySQL Clusterのストレージエンジンndbdを起動させるサーバ。

SQLノード: MySQLサーバmysqldを起動させるサーバ。SQL処理を行うのみならず、ndbdとのインタフェースを行う。

# ロードバランサ

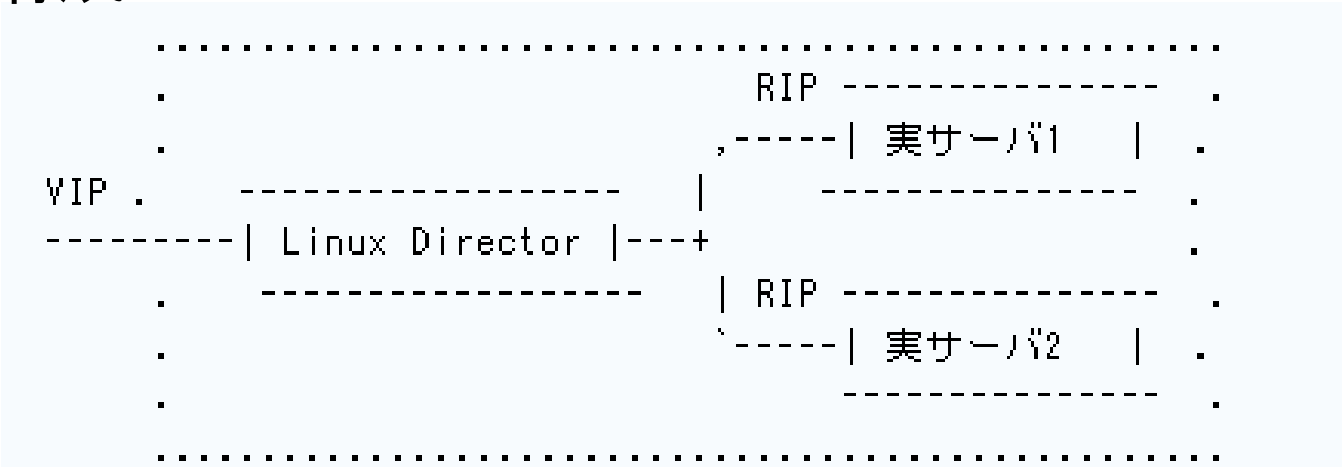
- DNSラウンドロビン
  - 設定が簡単(Aレコードに複数アドレス)
- IPTables DNAT
  - 送信先を振り分ける(逆はSNAT）
- LVS(Linux Virtual Server)
  - 冗長化、フェイルオーバーにも対応可能
    (linux-ha, ldirectord)

# LVS(Linux Virtual Server)

- Linux Virtual Serverプロジェクトが開発
  http://www.linuxvirtualserver.org/
- レイヤ4スイッチ
  - TCP,UDPパケットレベルでの振り分け(Webアプリは要注意)
- IP Virtual ServerがNetfilterモジュールとして稼動
  - カーネル内転送でオーバヘッドが少ない
- Linux Director Daemonでフェールオーバー
- Linux HA(High Availability)Hartbeatで冗長化

# LVSの構成（負荷分散）

- 最小構成




Linux Director：LVSを稼動させるLinuxホスト

実サーバ：実際にサービスを行う

仮想IPアドレス（VIP）：Linux Directorに割り当てるIPアドレス

実IPアドレス（RIP）：実サーバのIPアドレス

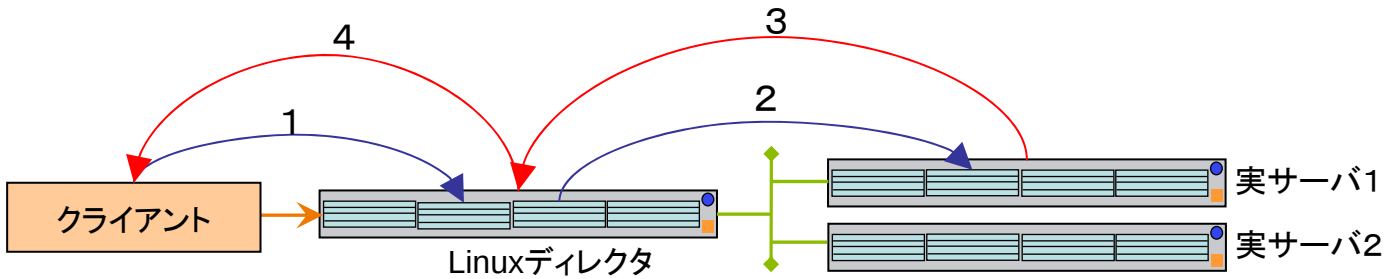
# LVSのパケット転送方式

- NAT（Network Address Translation）
  - IPマスカレード(RFC 1918)のような方式
  - もっとも実装が簡単
- ダイレクト・ルーティング
  - IPパケットをそのまま転送
  - 応答は実サーバから直接クライアントホストへ
- IP-IPカプセル（IPトンネル）
  - ethernetフレームを操作しないでIPパケットにカプセル化

# LVS NAT

- アドレス変換をしてパケットを実サーバへ転送
- ファイアウォール経由のパケットに対応可能

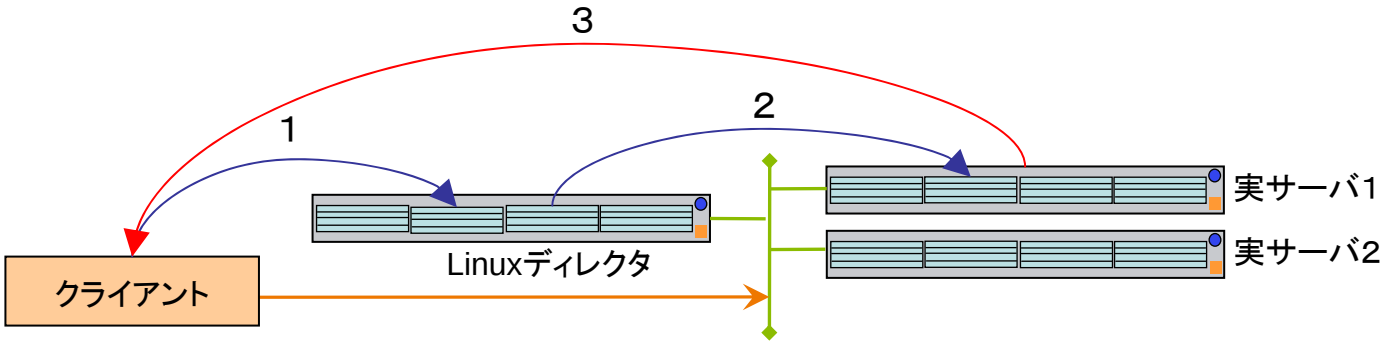
LVS NATのパケット転送



# LVS ダイレクトルーティング

- パケットをそのまま実サーバのMACアドレスに転送
- 応答がLinux Directorを経由しないぶん高速
- ループバック･デバイスがarpに応答しないようにする必要がある(カーネルパッチ）

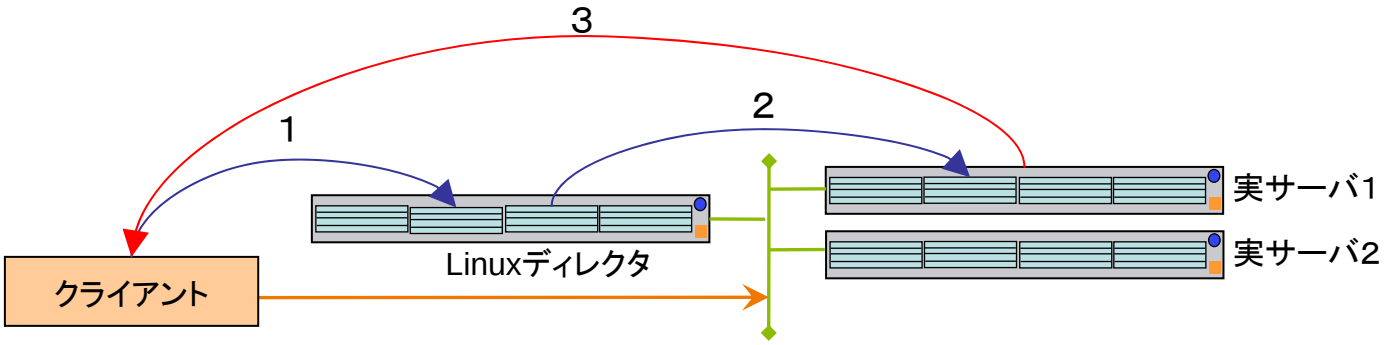
LVSダイレクトルーティングのパケット転送



# LVS トンネリング

- ダイレクト・ルーティングと類似
- パケットを転送する際にIPパケットにカプセル化する
- 実サーバが別のネットワーク上にあっても構わない

LVSトンネリングのパケット転送



# LVSの冗長化

- Linux Director Daemon(ldirectord)
  - 起動の自動化
  - 実サーバのヘルスチェック、フェールオーバー

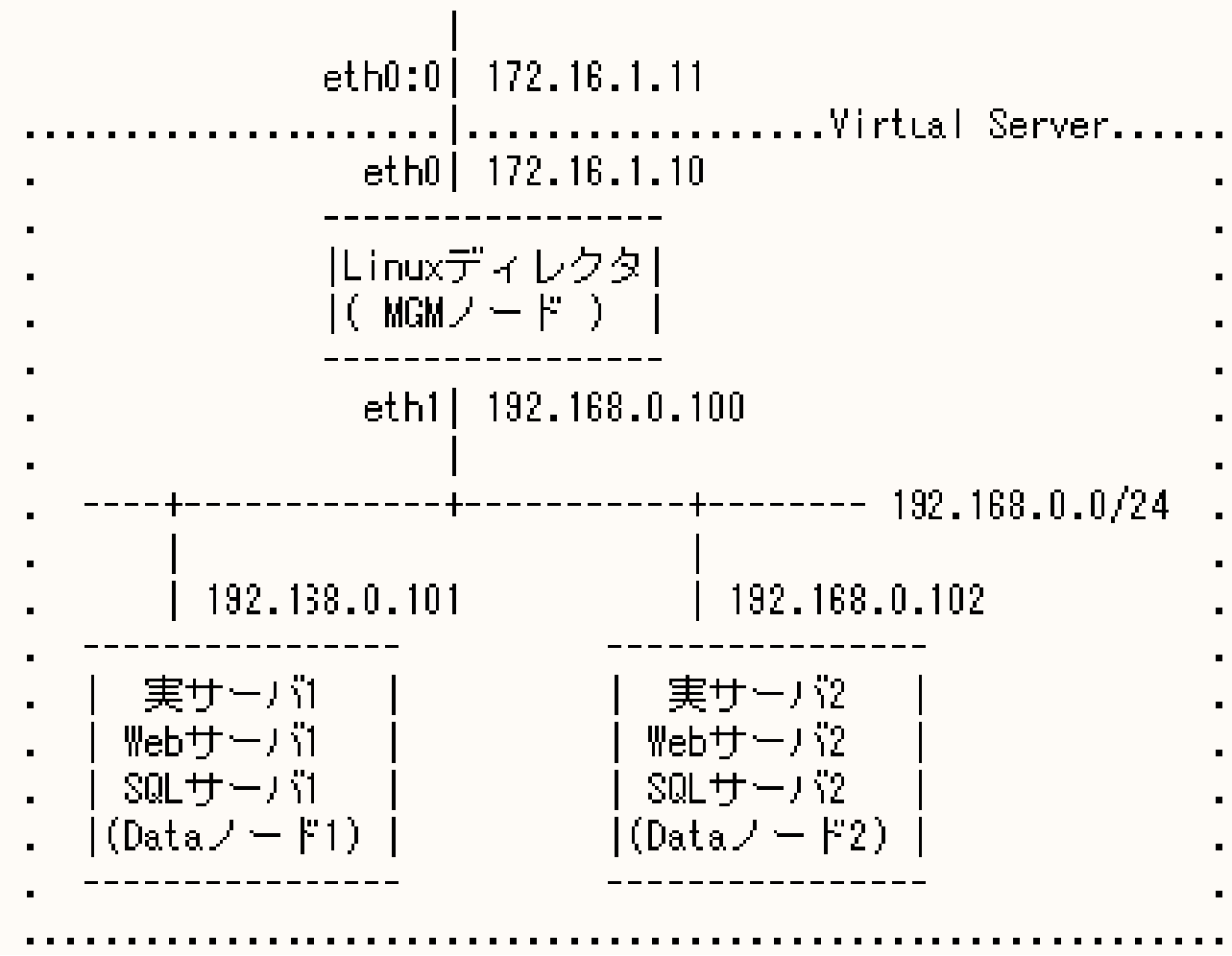
    http://www.vergenet.net/linux/ldirectord/
- Linux HA Hartbeat の導入
  - ロードバランサの二重化

    http://www.linux-ha.org/

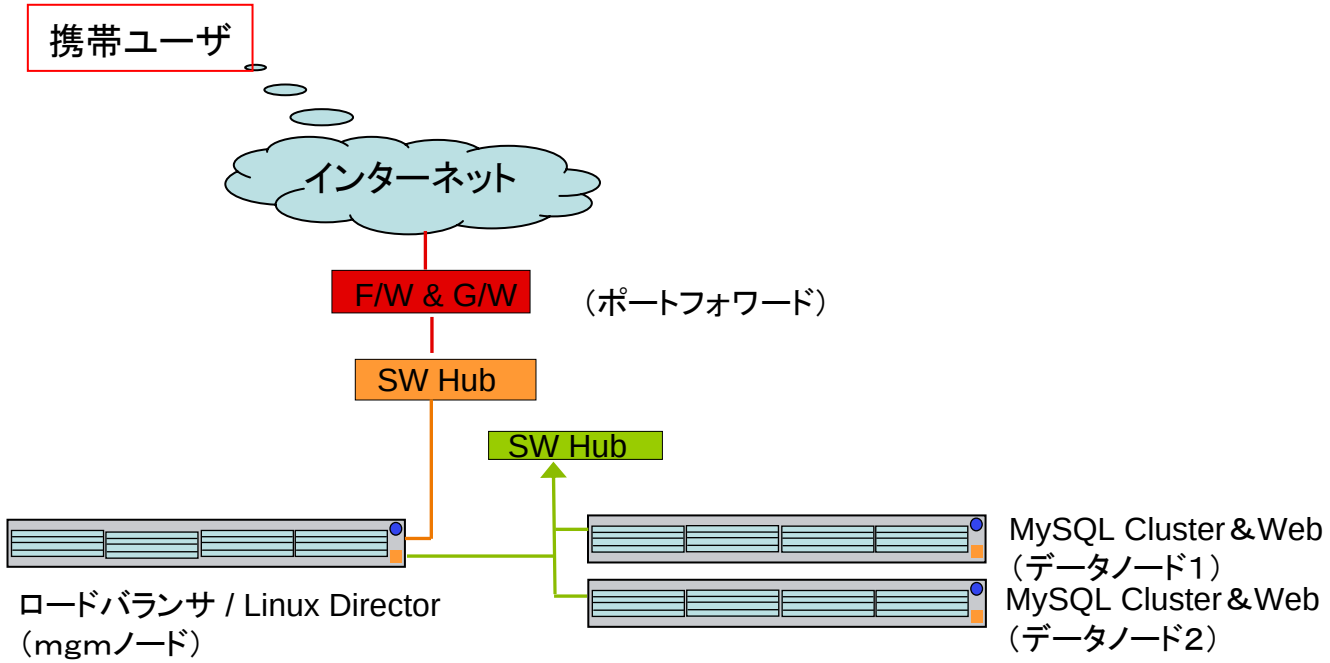
    http://ultramonkey.jp/papers/lvs_tutorial/

    http://www.linuxvirtualserver.org/docs/hartbeat_ldirectord.html/

# MySQL Cluster使用Webアプリケーションのための LVS構成例



# クラスタ機器構成


携帯ユーザ
インターネット
F/W & G/W
（ポートフォワード）
SW Hub
SW Hub
ロードバランサ / Linux Director
（mgmノード）
MySQL Cluster＆Web
（データノード1）
MySQL Cluster＆Web
（データノード2）

# LVSのインストール例

- LinuxディストリビューションにはCentOS 4.0を使用
  - Linux HA がパッケージ化されている
- 利用パッケージ
  - LVS(ipvsadm)
  - Linux Director(heartbeat-ldirectord)

# LVS(ipvsadm)のインストール

mgm_node# **yum install ipvsadm**
...
  Installing: ipvsadm                    ############################## [1/1]

Installed: ipvsadm.i386 0:1.24-6
Complete!

mgm_node# **rpm -ql ipvsadm**
/etc/rc.d/init.d/ipvsadm
/sbin/ipvsadm
/sbin/ipvsadm-restore
/sbin/ipvsadm-save
/usr/share/doc/ipvsadm-1.24
/usr/share/doc/ipvsadm-1.24/README
/usr/share/man/man8/ipvsadm-restore.8.gz
/usr/share/man/man8/ipvsadm-save.8.gz
/usr/share/man/man8/ipvsadm.8.gz

# ipパケットのフォワード設定

- カーネルがIPパケットをフォワードする必要あり

  /etc/sysctl.conf に

  net.ipv4.ip_forward = 1

  を記述し、sysctl コマンドで設定を有効にする。

mgm_node# **/sbin/sysctl -p**
net.ipv4.ip_forward = 1
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.default.accept_source_route = 0
kernel.sysrq = 0
kernel.core_uses_pid = 1

# 仮想インターフェースの設定

- 仮想インターフェースeth0:0を設定します
  (eth0とeth1のインターフェースは設定済みとする)

- コマンド行:

```
# /sbin/ifconfig eth0:0 172.16.1.11
 netmask 255.255.0.0 broadcast
 172.16.255.255
```

（実際は1行）

# 仮想インターフェースの設定

- 起動時自動設定:
  /etc/sysconfig/network-scripts/ifcfg-eth0:0 ファイルに記述

```
DEVICE=eth0:0
BOOTPROTO=static
IPADDR=172.16.1.11
BROADCAST=172.16.255.255
NETMASK=255.255.0.0
NETWORK=172.16.0.0
ONBOOT=yes
```

- 仮想インターフェースを有効に

```
mgm_node# /sbin/ifup eth0:0
```

- インターフェースの状態の確認

```
mgm_node# /sbin/ifconfig eth0:0
```

# LVSの設定

- ipvsadm 管理コマンドを用いてIPバーチャルサーバを設定
  - 設定の追加 (-A, -aオプション)
  - テストのためにアクセス毎に切り替わるラウンドロビンスケジューラ (-s rr オプション)
- 設定を確認
  - リスト出力して確認 (-L -n オプション)

# ipvsadmコマンドの実行

```
mgm_node# /sbin/ipvsadm -A -t 172.16.1.11:80 -s rr
mgm_node# /sbin/ipvsadm -a -t 172.16.1.11:80 -r 192.168.0.101:80 -m
mgm_node# /sbin/ipvsadm -a -t 172.16.1.11:80 -r 192.168.0.102:80 -m

mgm_node# /sbin/ipvsadm -L -n

 IP Virtual Server version 1.2.0 (size=4096)
 Prot LocalAddress:Port Scheduler Flags
   -> RemoteAddress:Port           Forward Weight ActiveConn
   InActConn
 TCP  172.16.1.11:80  rr
   -> 192.168.0.102:80             Masq    1      0          0
   -> 192.168.0.101:80             Masq    1      0          0
```

# LVSのテスト

- それぞれの実サーバを稼動させる
- それぞれのノード名が区別できるようなコンテンツを同じパスに設置

  例）

```
<?php
echo $_SERVER['HOSTNAME'];
?>
```

- 外部から仮想サーバ（仮想アドレス）へのアクセステスト

- アクセスの毎にノードが切り替わることを確認

  （ホスト名が変わる）

# LVSの設定解除

- LVSの設定はLinuxディレクタから実行可能
- ipvsadm 管理コマンドを用いる
  - 設定の削除コマンド(-D, -d オプション)

```
mgm_node# /sbin/ipvsadm -d -t 172.16.1.11:80 -r 192.168.0.101:80
mgm_node# /sbin/ipvsadm -d -t 172.16.1.11:80 -r 192.168.0.102:80
mgm_node# /sbin/ipvsadm -D -t 172.16.1.11:80
```

- 設定を確認
  - リスト出力して確認(-L -n オプション)

```
mgm_node# /sbin/ipvsadm -L
```

# ldirectordの設定

- 設定ファイルは　/etc/ha.d/ldirectord.cf
  - ipvsadm の実行パラメータの設定
  - 実サーバの状態検知（ヘルスチェック）のための方法を指定

# ldirectord.cf

```
# グローバル・ディレクティブ
checktimeout=10                          # タイムアウト
checkinterval=300                        # チェック間隔
autoreload=no                            # yes: 運用モード
logfile="local0"                         # syslog ユニット
quiescent=no                             # localhost は実サーバに含めない

# HTTPの仮想サーバ
virtual=172.16.1.11:80                   #  仮想サーバアドレス
        real=192.168.0.101:80 masq       # gate: dynamic routing, masq: lvs nat
        real=192.168.0.102:80 masq       # gate: dynamic routing, masq: lvs nat
        service=http                     # サービス
        request="lvscheck.html"          # ヘルスチェックリクエスト
        receive="C.*"                    # ヘルスチェック文字列
        scheduler=wrr                    # スケジューラ（wrr:重み付ラウンドロビン）
        protocol=tcp                     #プロトコル
        checktype=negotiate              #チェック方式
        #fallback=127.0.0.1:80           # フォールバックサービスがある場合に指定
```

# ldirectordの実行

- directordデーモンを起動

```
mgm_node# /etc/init.d/ldirectord start
 Starting ldirectord                          [  OK  ]
```

- LVSの状態を確認

```
mgm_node# /sbin/ipvsadm -L -n
```

- アクセステストを行ない問題がなければ LVS の設定は終了

- ldirectord自動起動のための登録

```
mgm_node# /sbin/chkconfig ldirectord on
mgm_node# /sbin/chkconfig –list ldirectord
```

# PHPアプリケーションの負荷分散サーバ対応

- セッション管理の問題
  - セッションハンドラ関数の定義
    - _open, _close, _read, _write, _destroy, _clean
  - session_set_hander()でセッションハンドラ関数の登録
- 参照

  「入門PHPセキュリティ」 (オライリー）

# MySQL　Cluster 事例

- 日本
  - 携帯電話アプリケーションサイト
    http://www.acornsoft.co.jp/products/mysql/index.html
- 海外
  - 8x8: アメリカのIP電話サービス会社
  - イタリアのイエローページサイト: NDB APIで実装
  - フランスの銀行:株式取引
  - 通信会社アルカテル社:位置情報管理
  - エリクソン:テレコムアプリケーション