Linuxで作るお家サーバ

アンカーテクノロジー株式会社 國武功一 <kunitake@anchor.jp>

今日取り上げる話題について

- **お家サーバといえども立派なサーバ管理者**
  - **運用における見逃しがちなTIPSを中心に紹介**
- **取り上げる話題**
  - **サーバ構築について**
  - DynamicDNS
  - Web**サーバの運用**
  - **セキュリティ**
  - **その他**

サーバ構築について

- 使用用途の決定
- ディストリビューションの決定
- バックアップ計画
- パーティションの決定

※) ここではハードウェアスペックについては触れません

使用用途について

- **提供サービスの決定**
  - **どのような構成にするか、なにをインストールすべきかがこれで決まってくる。**
  - **今回は、以下のサービスを提供するものとする。**
    - Apache
    - SSH
- **リソース監視ツール**
  - sysstat
  - HotSaNIC
- **バックアップについて**
  - LVM Snapshot**機能**

ディストリビューションの決定

- RedHat**系**
  - Fedora Core
    人気が高い。ただし安定稼動には向かない一面も。
  - CentOS
    RHEL**互換クローンで比較的セキュリティ対応も早い。**
    RHEL4**と同様、**CentOS4**系は**SELinux**にも対応**
- SUSE
  - **ヨーロッパを中心に人気のあるディストリビューション。多くの管理コマンドは**RedHat**系を踏襲している。**
- Debian GNU/Linux
  - **最近** sarge**がリリースされる。リリースサイクルが遅いという指摘もあるが、管理自体は楽。**

- バックアップ計画
  - 対象について
    - 設定ファイル
    - データなど
  - 頻度について
- 障害時の対応
  - バックアップデータからの復旧

- なぜパーティションを分割するのか？

分割の一例:

```
kunitake@xen:~$ df -h
Filesystem                Size  Used Avail Use% Mounted on
/dev/hdc1                 912M  114M  750M  14% /
tmpfs                      62M     0   62M   0% /dev/shm
/dev/hdc2                 228M   39M  178M  18% /boot
/dev/mapper/system-home    22G  9.9G   13G  45% /home
/dev/hdc5                 456M  8.1M  424M   2% /tmp
/dev/mapper/system-usr    5.0G  1.1G  4.0G  21% /usr
/dev/mapper/system-var    8.0G  561M  7.5G   7% /var
```

パーティション分割の必要性

- **問題が発生した際にその被害を最小限に抑える**
  例
  - /var/spool/mail **が溢れても**/var**には影響を与えないように**
  - **異常プロセスが**/tmp**を使い切っても他への影響を最小限に抑える**
- **用途に応じて、マウントオプションを使い分けることが可能になる。**

•noatime
最終アクセス時刻を記録しない
•noexec
バイナリの実行を禁止する
•nosuid(*1)
suid **および**sgid**ビットを無効にする。**
•nodev
デバイスの作成を無効にする
•ro
ファイルシステムをリードオンリーでマウントする

どこまで分割するべきか？

例

マウントポイント	想定されるサイズ	概要
/	256MB～	ルートファイルシステム
/boot	64MB～	カーネルなど起動に必要なデータが置かれる
/usr	2GB	プログラムなどが置かれる
/home	?GB	一般ユーザのホームディレクトリ
/var	768MB～	ログやサーバデータなどが置かれることも
/srv	?BG～	そのサーバでサービスされるデータなどが置かれる
/tmp	256MB～	プログラムの一時ファイルが作られる
スワップ	メモリ搭載量x2	メモリが溢れた際に仮想的なメモリとして利用

どのように分割すべきか、また割り当て領域については、使用用途により異なるため熟考が必要。

SWAPについて

- スワップは搭載メモリの2倍を確保すべし
  - これはなぜ？
    - 搭載されるメモリが少なかったころの名残りか。
    - SWAPが起きている時点で、システムとしてはかなりのペナルティを抱えるが、それを最小限に抑えるための工夫。
    - 複数ディスクにSWAP領域を設けることで、高速化を図る例もある。

/etc/fstab一例

```
/dev/sda2    none   swap    sw,pri=5
/dev/sdb2    none   swap    sw,pri=5
```

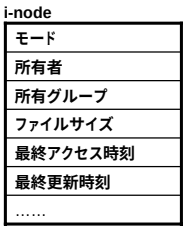
(*)priは0から32767までの優先度を示す値。32767が利用される優先度が一番高い

ファイルシステムについて

- EXT2
  - Linuxで広く使われているファイルシステム。安定性は高い。
- EXT3
  - EXT2にジャーナリングの機能を追加したもの。
- ReiserFS
  - ジャーナリングファイルシステムの1つ。小さなファイルに対して効率がよい。比較的高機能だが、安定性に欠けるとする向きもある。dumpツール非対応
- XFS
  - SGIが提供したジャーナリングファイルシステムの1つ。dumpツールについては、xfsdumpツールが提供されている。ただし従来のdumpツールと互換性はなし。
- JFS
  - IBMが提供したジャーナリングファイルシステムの1つ。ここで列挙したファイルシステムの中では一番使われる率が低い？dumpツール非対応

補足：ジャーナリングファイルシステムとは？

- ファイルシステムの更新記録を、ジャーナリング（ログ）として記録する。ファイル更新時にシステムが予期せぬ障害でダウンしても、メタデータ（ファイルの構成情報）の整合性を保つことができる。
  - 利点：システムに障害が発生した場合、従来のext2などは、すべてのファイルに対して、整合性チェックを行うため、かなりの時間を要するが、数百GBでも、数秒でチェックが行える。
  - 留意点：保護されるのはあくまでファイルシステムであり、実データは保障対象外なので、過信は禁物



LVMのススメ

- LVM(Logical Volume Manager)とは？
  - 乱暴に言ってしまえばディスクパーティションを仮想化してしまう。複数のHDDを束ねて1つのパーティションに見せることも可能。
  - 多くのディストリビューションで、インストール時にLVMを選択することが可能。
- LVMのメリット
  - パーティション個数制限からの解放
  - ディスク容量をあとで変更可能
  - Snapshotによるバックアップに対応

LVMの構成要素 –PV, VG, LV-

- PV(Physical Volume)
  - **従来の**/dev/hda1, /dev/hda2 **などを示す。これを束ねて**VG**を構成する**
    (VG**に組み込まれたディスクパーティションを**PV**と呼ぶと思えばよい。**
    PV**にされたパーティションは**LVM**と表示される)**

```
# pvcreate –v /dev/hdc6
pvcreate -- physical volume "/dev/hdc6" successfully created

# fdisk -l /dev/hdc

Disk /dev/hdc: 160.0 GB, 160000000000 bytes
255 heads, 63 sectors/track, 19452 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hdc1               1         124      995998+  83  Linux
/dev/hdc2             125         155      249007+  83  Linux
/dev/hdc3             156         404     2000092+  82  Linux swap
/ Solaris
/dev/hdc4             405       19452   153003060    5  Extended
/dev/hdc5             405         466      497983+  83  Linux
/dev/hdc6             467       19452   152505013+  8e  Linux LVM
```


LVMの構成要素 –PV, VG, LV-

- VG(Volume Group)
  - **1つ、もしくは複数の**PV**を束ねて構成される。設定する**VG**にはそれぞれ**
    任意の名前をつける必要がある。

```
# vgcreate system /dev/hdc6

# vgdisplay
  --- Volume group ---
  VG Name               system
  System ID
  Format                lvm2
  Metadata Areas        1
  Metadata Sequence No  18
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                7
  Open LV               6
  Max PV                0
  Cur PV                1
  Act PV                1
  VG Size               145.44 GB
  PE Size               4.00 MB
  Total PE              37232
  Alloc PE / Size       14080 / 55.00 GB
  Free  PE / Size       23152 / 90.44 GB
  VG UUID               nXH9H5-5Yky-2wcG-5v9m-8lfC-r7VI-Kgoh6Z
```

LVMの構成要素 –PV, VG, LV-

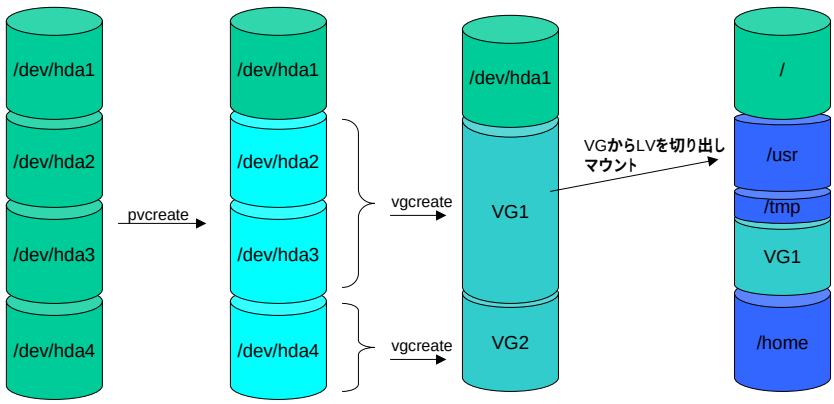
- LV(Logical Volume)
  - **仮想化されたパーティションだと思えばよい。作成されたLVをパーティションとして指定してファイルシステムを作成、マウントすれば従来どおり、通常のパーティションとして利用できる。作成されたディスクは下記の例だと**
    - /dev/system/srv　　(/dev/VG名/LV名)

    のように見える。

```
# lvcreate --name srv --size 4G system
 Logical volume "srv" created

# lvdisplay
 --- Logical volume ---
  LV Name                /dev/system/srv
  VG Name                system
  LV UUID                V9XL4b-ezNf-MVgJ-Tcjs-uXQh-IuWr-EiCQzM
  LV Write Access        read/write
  LV Status              available
  # open                 1
  LV Size                4.00 GB
  Current LE             1024
  Segments               1
  Allocation             inherit
  Read ahead sectors     0
  Block device           254:0
```

<= "system"というVGから4GBを切り出し、"srv"という名のLVを作成

LVMの構成要素 –PV, VG, LV-



LVMで作られた領域をマウントする。

- **利用したいファイルシステムでフォーマットして通常通りマウントする。**

フォーマット

```
EXT3の場合
# mk2fs -j /dev/system/srv

ReiserFSの場合
# mkreiserfs /dev/system/srv
```

領域の拡張

```
# lvextend -L+2G /dev/system/srv     <= 2GB追加拡張

ReiserFSの場合
# resize_reiserfs -s +2G /dev/system/srv

ext2, ext3 の場合(*1)
# ext2online /dev/system/srv
```

(*1) remountは不要だが、kernelへのパッチ要。umountした上でext2resizeコマンドを利用することでも実現可能。

LVMのsnapshot機能

- **不整合の起きづらいsnapshotでサービスを稼動させながらデータをバックアップ可能**

```
# modprobe dm-snapshot

# lvcreate --size=512M --snapshot --name snap /dev/system/srv
# mount /dev/system/snap /mnt

# df -h
Filesystem                Size  Used Avail Use% Mounted on
/dev/hdc1                 912M  114M  750M  14% /
tmpfs                      62M     0   62M   0% /dev/shm
/dev/hdc2                 228M   39M  178M  18% /boot
/dev/mapper/system-home    22G  9.9G   13G  45% /home
/dev/hdc5                 456M  8.1M  424M   2% /tmp
/dev/mapper/system-usr    5.0G  1.1G  4.0G  21% /usr
/dev/mapper/system-var    8.0G  560M  7.5G   7% /var
/dev/mapper/system-srv    7.9G  1.2G  6.4G  15% /srv
/dev/mapper/system-snap   7.9G  1.2G  6.4G  15% /mnt
```

マウントしてしまえばsnapshotを取った時点でのLVの内容そのものとなる。Snapshotは基本的に、元の領域の差分だけを保持するため、激しくデータの更新が行われないようなら、実際にはほとんど容量を消費しない。

インストール直後のチェック項目

- **パッケージを最新に！**
  - RedHat**系**

    # yum update
  - Debian

    # apt-get update && apt-get upgrade
- **不要サービスの停止**

不要サービスの停止

- **現状のネットワークサービスを確認**

```
# netstat -ltupn
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      452/portmap
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      559/httpd
tcp        0      0 0.0.0.0:5680            0.0.0.0:*               LISTEN      468/cannaserver
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      534/master
tcp        0      0 0.0.0.0:443             0.0.0.0:*               LISTEN      559/httpd
tcp        0      0 :::22                   :::*                    LISTEN      1034/sshd
```

- **不要なサービスの停止**

```
サービスの停止
RedHat系
# /sbin/service canna stop
# /sbin/service portmap stop

Debian系  サービスの停止
# /etc/init.d/canna stop
# /etc/init.d/portmap stop

自動起動の停止
RedHat系
# /sbin/chkconfig canna off
# /sbin/chkconfig --del portmap

Debian系
# update-rc.d –f canna remove
# update-rc.d –f portmap remove
```

- **正しい時刻で刻まれるログの大切さ**
  - **トラブルが起きたとき、その発生時刻を正確に把握することは大切。他のサーバと付き合わせた時、時刻がずれていると悲惨な目に……**

もし利用しているISPがNTPサーバを提供していればそれを優先的に使おう！

もしなければ

時刻情報提供サービス for Public
http://www.jst.mfeed.ad.jp/

を使ってみよう。

Cronへの設定例

```
20 4 * * * /usr/sbin/ntpdate ntp.jst.mfeed.ad.jp
```


- **設定に入る前に……**
  - **RCSを使ってみよう！**

RCSとは？

- Revision Control System（履歴管理システム）
  - ファイルの更新履歴などが記録できる。
    - 設定ファイルなどの履歴管理に使うと便利！
  - やることは
    - チェックイン（ファイルを登録）
    - チェックイン時、履歴情報を記述
    - チェックアウト（ファイルを取り出す）

    基本的にこれだけ。

RCSを使う　～下準備　その1～

- まず管理したいファイルと同じディレクトリに“RCS”というディレクトリを作成

```
# cd /etc/apache
# mkdir RCS
```

- ファイルを登録（チェックインと初期化）

```
# ci -i httpd.conf
RCS/httpd.conf,v  <--  httpd.conf
enter description, terminated with single '.' or end of file:
NOTE: This is NOT the log message!
>> default setting.
>> .
initial revision: 1.1
done
```

履歴ログを入力

•ファイルを取り出す(チェックアウト)

```
# co httpd.conf
RCS/httpd.conf,v  -->  httpd.conf
revision 1.1
done
```


• まずロックする

(編集中に自分以外の人がチェックインしまわないように)

```
# co -l httpd.conf
RCS/httpd.conf,v  -->  httpd.conf
revision 1.1 (locked)
done
```

•設定ファイルを編集 (お好きなエディタで)

```
# vi httpd.conf
```

•修正内容をRCSレポジトリに登録
•同時にロックも解除される。

```
# ci –u httpd.conf
RCS/httpd.conf,v  <--  httpd.conf
new revision: 1.2; previous revision: 1.1
enter log message, terminated with single '.' or end of file:
>> アクセスが集中。MaxClientsを200へ増やした。
>> .
done
```


- rcsdiff
  - 設定ファイルの差分が確認できる

```
$ rcsdiff httpd.conf
```

  - リビジョンを指定することも可能

```
$ rcsdiff –r1.1 –r1.2 httpd.conf
or
$ rcsdiff –u –r1.1 –r1.2  httpd.conf
===================================================================
RCS file: RCS/httpd.conf,v
retrieving revision 1.1
retrieving revision 1.2
diff -u -r1.1 -r1.2
--- httpd.conf        2005/11/01 08:18:49     1.1
+++ httpd.conf        2005/11/02 08:19:13     1.2
@@ -62,7 +62,7 @@
 #MinSpareServers     10
 MinSpareServers     10
 MaxSpareServers     10
-MaxClients          50
+MaxClients          200
 MaxRequestsPerChild  0
 </IfModule>
```

RCSはこんなに便利！(Cont)

- 昔の設定ファイルも取り出し可能
  - 万が一、オペミスでファイルを削除してしまっても復旧可能

```
$ rm httpd.conf
$ co -u httpd.conf
RCS/httpd.conf,v  -->  httpd.conf
revision 1.2 (unlocked)
Done
$ ls httpd.conf
httpd.conf
```

  - リビジョンを指定して取り出すことも可能

```
$ co -r1.1 -u httpd.conf
RCS/httpd.conf,v  -->  httpd.conf
revision 1.1 (unlocked)
done
```


RCSはこんなに便利！(Cont)

- 作業履歴もバッチリ
  - rcs2log

```
$ rcs2log httpd.conf
2005-11-02  KUNITAKE Koichi  <kunitake@stardust>

        * httpd.conf: アクセスが集中。MaxClientsを200へ増やした。

        * httpd.conf: ServerNameをwww.example.comへ変更

2005-11-01  KUNITAKE Koichi  <kunitake@stardust>

        * httpd.conf: default setting.
```

もしRCSを使わないとこんな悪夢が待ってるかも？

```
$ ls
access.conf            httpd.conf.20050911     httpd.conf.bak  srm.conf
httpd.conf             httpd.conf.20051101     httpd.conf.old
httpd.conf.20040609    httpd.conf.20051101_01  magic
httpd.conf.2005.04.19  httpd.conf.20051101_02  mime.types
```

manの基礎の基礎

- TIPSの多くが実はmanを書いてあることが多い。

```
$ man –f swap
```

引数で示された名前のマニュアルページを探し出し、要約を表示する。

```
$ man –k swap
```

引数で示されたものをキーワードとして要約から検索を掛ける

このことさえ、manに書いてある！

```
$ man man
```


DynamicDNSについて

DynamicDNSとは？

- 従来、固定アドレスを想定したDNSに対して、動的に短期間にレコードを更新しなければならないようなホストに対応するために作られた仕組み。
  - これにより固定IPアドレスの割り当てを受けなくても、FQDNによるアクセスが可能となる。
- もともとはDHCPなどと連携することを主眼に作られたが、これを一般ユーザ向けにサービスするところも現れている。今回は、そのサービスの利用について取り上げる。

Dynamic DNSサービスの仕組み

- DynamicDNSはおおまかに下記のようにIPアドレスの通知およびレコードの更新が行われる。

1. 登録したいIPアドレスが割り当てられているクライアントからIPアドレス通知用プログラムが起動
2. IPアドレス通知用プログラムからIPアドレスの情報を取得（認証含む）
3. 通知内容を元に、該当FQDNのIPアドレスを更新

DynamicDNSサービスの利用

- DynamicDNSはサービス提供者によってユーザにとっての利用方法は、さまざま
  - HTTPのリクエストによりIPアドレスを通知
    - フォーマットはさまざま
  - POP3サーバへの接続によりIPアドレスを通知
  - FTPサーバへの接続によりIPアドレスを通知
  - Webのフォームを利用してIPアドレスを登録

などなど……

DiCEを使ってみよう！

- DiCEとは？
  - 数多くのDynamicDNSサービスに対応したDynamicDNSのクライアント
- 公開先
  - http://www.hi-ho.ne.jp/yoshihiro_e/dice/linux.html

```
$ tar zxvf diced01912.tar.gz
# mv DiCE /usr/local
# cd /usr/local/DiCE
# ./diced
=-=-=- DiCE DynamicDNS Client -=-=-=
Version 0.19 for Japanese
Copyright(c) 2001 sarad

:?    <= 使い方が表示される。
```

1. 利用したDynamicDNSサービスを“add”コマンドで、イベントとして登録
2. “ex <イベント番号>”、“ev <イベント番号>”で、動作確認
3. daemonとして起動

DiCE使用例

1. イベントの登録

```
:add
新しくイベントを追加します

DynamicDNSサービス名を入力してください
"?"で対応しているサービスを一覧表示します
(P)戻る
>ddo.jp
-------------------------------------------------
<< Dynamic DO!.jp >>
URL: http://ddo.jp/
*** 情報 ***
ユーザー名の入力は不要です
独自ドメインの場合はドメイン名を"ホスト"の所へ入力してください
=================================================
ドメイン名を入力してください
"?"でドメイン一覧を表示します
(P)戻る
>ddo.jp
=================================================
ホスト名を入力してください
(P)戻る
>asteroid
=================================================
ログインユーザ名を入力してください
(P)戻る
>
=================================================
ログインパスワードを入力してください
(P)戻る
>XXXXXX
=================================================
登録するIPアドレスを入力してください
空白にすると現在のIPアドレスを自動検出します
(P)戻る
>
=================================================
このイベントに題名を付けてください
(P)戻る
>ddosetup
```

```
=================================================
このイベントを実行するスケジュールを設定します
-------------------------------------------------
実行する頻度を指定してください (番号入力)
(0)1回のみ (1)1日1回 (2)1週間に1回 (3)1ヵ月に1回
(4)その他の周期 (5)IPアドレス変化時 (6)起動時
(P)戻る
>1
-------------------------------------------------
時刻を指定してください
入力例) 23:05
(P)戻る
>05:00
=================================================
詳細オプションを設定します
-------------------------------------------------
[ サービスタイプ ]
(0)無料 (1)有料
番号>0
-------------------------------------------------
[ SSL ]
(0)使用する (1)使用しない
番号>1
-------------------------------------------------
[ オフライン ]
(0)No (1)Yes
番号>1
=================================================
このイベントを有効にしますか? (Y/N)
(イベントの有効/無効は"EN/DIS"コマンドで切替えられます)
=================================================
このイベントを有効にしますか? (Y/N)
(イベントの有効/無効は"EN/DIS"コマンドで切替えられます)
>Y
=================================================
イベントを保存しますか? (Y/N)
>Y
イベント"ddosetup"を保存しました
=================================================
```


DiCE使用例

2. 動作テスト

```
:list
(No.) (イベント名)              (スケジュール)          (次回予定)
  0 *  ddosetup                 1日に1回      23:04     11/02 05:00
:ex 0
+ 11/1 18:29 にddosetupが実行されました
  IPアドレスを更新しました
:ev 0
-------------------------------------------------
[イベント名       ] ddosetup
[状態             ] 有効
[DNSサービス      ] ddo.jp
[更新ホスト       ] asteroid.ddo.jp
[ユーザ名         ]
[IPアドレス       ]
[スケジュール     ] 1日に1回 05:00

[次回更新日時     ] 2005年11月2日、05:00:00
[最終実行日時     ] 2005年11月1日、18:29:24
[最終更新IPアドレス] 192.0.2.123
[最終更新結果     ] IPアドレスを更新しました
(ID:000000)
-------------------------------------------------
```

3.daemonとして起動

```
# /usr/local/DiCE
# ./diced –l -d
```

SSHについて

- SSHへのパスワードアタック
  - sshによるログインは禁止するのが一般的になってましたが、期待通りに動いていますか？

```
Protocol 2
PermitRootLogin no
RSAAuthentication yes
PubkeyAuthentication yes
RhostsAuthentication no
IgnoreRhosts yes
RhostsRSAAuthentication no
HostbasedAuthentication no
PermitEmptyPasswords no
PasswordAuthentication no
```

このあたりは一般的な設定

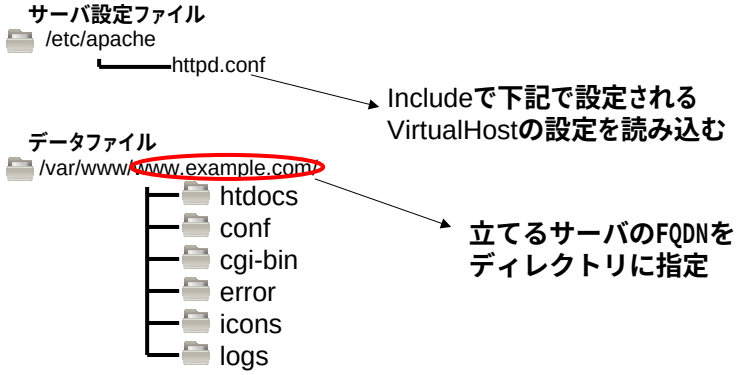
```
UsePAM yes    <= せっかく PasswordAuthentication noにしても
                 ChallengeResponseAuthenticationがYesだと、
                 パスワードでのログインが可能
```

Apacheについて

ここで触れるApacheの範囲

- 設定ファイルなどのレイアウト
- 安全な設定を目指して
- アクセス制御
- ログのローテーションについて
- 知ってると便利なツール

- **お奨めの設定ファイルのレイアウト**



•メリット
たとえ運用するVirtualHostが増減してもディレクトリレイアウトが崩れない
VirtualHost毎のフルバックアップもディレクトリ指定でOK。

•デメリット
標準構成から崩れるため logrorateなどを使う場合は別途見直しも必要

- **標準の設定ファイルをそのまま使わない**
  - **一から設定する**
    - **不要な設定を削り、なにが設定されていて、なにが設定されていないのかを把握し、理解を深める**
    - **不要なモジュールが削減され、パフォーマンス向上にも繋がる。**
  - **ただし、大いに参考にする**
    - **コメントはTIPSの宝庫**
    - **この設定いらないんじゃない？　と思った設定こそ重要だったりするかも**

これだけは注意して設定しておこう

- 安全な設定を目指して
- 攻撃者に楽をさせないために
- 見逃しがちな設定
- 設定したあとには？

Apacheの設定　～安全な設定を目指して～

- まず安全とされる設定を記述

```
<Directory />
  Options None
  AllowOverride None
  Order Deny,Allow
  Deny from all
</Directory>
```

誤ってスペースを入れてしまわないように注意。

=>誤って設定してしまい、サーバのファイルすべてを公開してしまわないために。

```
DocumentRoot /var/www/htdocs
<Directory /var/www/htdocs>
  Order Allow,Deny
  Allow from all
</Directory>
```

- Apacheのユーザ/グループ権限を別に設定する

```
User www-data
Group www-data
```

万が一にでもApache経由で進入された際に、被害を最小限に防ぐ

- UserDirを有効にした際に

```
UserDir public_html
UserDir disabled root    <= /~root/ でのアクセスを有効にしないために

<Directory /home/*/public_html>
        AllowOverride FileInfo AuthConfig Limit
        Options SymLinksIfOwnerMatch IncludesNoExec
</Directory>
```


- .htaccessへのアクセスを禁止

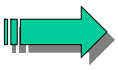
```
AccessFileName .htaccess

<Files ~ "^¥.ht">
   Order allow,deny
   Deny from all
</Files>
```

クライアントからの、".ht" から始まるファイル(.htpassswdや.htaccessなど)への参照を禁止する。

- Limitディレクティブへの誤解

よくある間違い

```
AuthUserFile  .htpasswd
AuthName "valid user only"
AuthType Basic
<LIMIT GET  POST>
  require valid-user
</LIMIT>
```



これで十分

```
AuthUserFile  .htpasswd
AuthName "valid user only"
AuthType Basic
require valid-user
```

Apacheの設定　～攻撃者に楽をさせないために～

```
ServerTokens Productonly
```

-与える情報を最低限に（攻撃者に楽をさせないためにも）

```
kunitake@stardust:~$ telnet localhost 80 (*1)
Trying 127.0.0.1...
Connected to localhost.localdomain.
Escape character is '^]'.
HEAD / HTTP/1.0

HTTP/1.1 200 OK
Date: Thu, 03 Nov 2005 14:45:43 GMT
Server: Apache/2.0.55 (Debian) DAV/2 SVN/1.2.3 PHP/4.4.0-4
Last-Modified: Thu, 15 Sep 2005 09:02:16 GMT
ETag: "13747-0-bbcd5600"
Accept-Ranges: bytes
Connection: close
Content-Type: text/html
```

Before

After

```
kunitake@stardust:~$ telnet localhost 80
Trying 127.0.0.1...
Connected to localhost.localdomain.
Escape character is '^]'.
HEAD / HTTP/1.0

HTTP/1.1 200 OK
Date: Thu, 03 Nov 2005 14:47:28 GMT
Server: Apache
Last-Modified: Thu, 15 Sep 2005 09:02:16 GMT
ETag: "13747-0-bbcd5600"
Accept-Ranges: bytes
Connection: close
Content-Type: text/html
```

(*1) Apacheが**IPv6に対応していれば、**“**telnet ::1 80**”**でももちろん**OK.

Apacheの設定　～見逃しがちな設定～

```
UseCanonicalName On
```

- **この設定を有効にすると、“/”を省略した時、そこにそのURLでファイルがなければ、ServerNameで設定されたサーバ名を利用し、“/”を補完した上で、リダイレクトを掛けてくれる。**

```
http://www.example.com/hoge
```

hogeというファイルがなければ“/”を補完

```
http://www.example.com/hoge/
```

Apacheの設定　～見逃しがちな設定 その2～

```
AddDefaultCharset UTF-8
```

- **この設定があると、ドキュメントで指定されたMETAタグの設定が上書きされてしまい、文字化けの原因に。**

```
AddDefaultCharset UTF-8
```

↓ **不要なら、Offにしておく(*)**

```
AddDefaultCharset Off
```

```
ScriptAlias /cgi-bin/ /usr/lib/cgi-bin
```

- **AddTypeで指定されたファイルではなくても、指定されたディレクトリ以下のファイルはすべてCGIと解釈されるので、注意が必要**

(*)ただしコンテンツ側 (特にCGI)で、きちんと適切なMETAタグが指定されることが前提となる。指定されていないと、ブラウザの自動文字コード判別機能の誤作動により、思わぬところで、XSSの脆弱性を露呈させてしまうことがある。
cf. http://www.cert.org/advisories/CA-2000-02.html
http://slashdot.jp/security/article.pl?sid=05/12/21/2318216

Apacheの設定　～VirtualHost～

•/etc/apache/httpd.conf

```
Port 80
ServerName ddns.example.jp
NameVirtualHost *:80

# 2005/11/01 追加
Include /var/www/www.example.com/conf/vhost.conf
```

•/var/www/www.example.com/conf/vhost.conf

```
<VirtualHost *>
        ServerAdmin kunitake@example.com
        ServerName  www.example.com
        DocumentRoot /var/www/www.example.com/htdocs
        <Directory /var/www/www.example.com/htdocs>
                Options Indexes FollowSymLinks MultiViews
                AllowOverride None
                Order allow,deny
                allow from all
        </Directory>

        ErrorLog  "|/usr/sbin/cronolog /var/www/www.example.com/logs/error_log-%Y%m%d"
        CustomLog "|/usr/sbin/cronolog /var/www/www.example.com/logs/access_log-%Y%m%d" combined

</VirtualHost>
```

立てるサーバをすべてNameベースのVirtualHostにすることで、IPアドレスを総なめしてアタックするようなワームからのアクセスを避けることができるという副次的なメリットもあり(*)

(*)もう少し正確にいえば、サーバ自身としてはアクセスは受けるが、VirtualHost毎に見れば、複数台設定時などでは、アクセスを受けないものが出てくるということ。

Apache ～設定したあとには？～

- **設定修正後は、文法チェック**

```
# apachectl configtest
```

- VirtualHost**の設定確認**

```
# httpd -S
```


ログのローテーションについて

- **一般的なログのローテート方法**
  - logrotate**を用いる**
    - **たいていの**Distribution**で取られている手法**
    - **ローテートの際に、**Graceful restart**が掛けられる。一瞬リクエストを取りこぼす可能性も。**
- Apache**の**restart/reload**を嫌った場合は？**
  - **独自スクリプト**
    - **現状のファイルをコピーし、ログファイルを切り詰める**

      # cat /dev/null > /var/log/apache2/access.log
  - **パイプ経由での**logging
    - Apache**付属の**rotatelogs
    - cronolog（**今回はこれとの連携を説明**）
      - rotatelogs**より高機能**

logrotateを使う

- Apacheのlogrotate用の設定ファイルを用意

```
/etc/logorotate.d/apache


/var/log/httpd/*.log {
          monthly
          rotate 12
          compress
          delaycompress

          sharedscripts
          postrotate
                    apachectl graceful > /dev/null 2> /dev/null
          endscript
}
```

- 設定が間違っていないかをチェック

```
# logrotate -d /etc/logrotate.conf
```


Cronolog ～パイプ経由でのlogging～

設定は、Apacheのログ設定の箇所に記述する。

```
TransferLog "|/usr/sbin/cronolog /www/logs/%Y/%m/%d/access.log“
ErrorLog    "|/usr/sbin/cronolog /www/logs/%Y/%m/%d/errors.log"
```

- 上記の例のように、ディレクトリ毎にログを分けることも可能

表　Cronologで利用できるファイル名に関するオプション（※）

オプション	出力
%d	日 (01..31)
%m	月 (01..12)
%y	年 (00..99)
%Y	年 (1970..2038)

（※）他のオプションについてはhttp://cronolog.org/usage.htmlを参照

アクセスログ解析ソフトについて

Awstats

- グラフィカル
  - その動作の重さから、大規模には向かない
  - 静的なページへの吐き出しやPDF化なども可能

Awstats：cronologとの連携

•VirtualHostにおけるログの設定

```
CustomLog "|/usr/sbin/cronolog /var/www/www.example.com/logs/access_log-%Y%m%d " combined
ErrorLog  "|/usr/sbin/cronolog /var/www/www.example.com/logs/error_log-%Y%m%d"
```

•Awstatsにおけるログの指定

```
#LogFile="gzip -d < /var/www/www.examole.com/logs/access_log-%YY-24%MM-24%DD-24.log.gz ¥
#                  |/usr/local/bin/jconv4utf8.pl|"
LogFile="/var/www/www.example.com/logs/access_log-%YY-24%MM-24%DD-24"
```

- Cronologとawstatsは相性がよく、日付のフォーマットもほぼそのまま利用できる。Awstatsでは、前日のログを解析するため、"-24"とし、24時間前の日付をファイル名として指定する。

Webdruid

- 比較的解析は比較的速い
  - graphvizを使って、ユーザの導線なども把握可能

Webdruid：cronologとの連携

•VirtualHostにおけるログの設定(/var/www/www.example.com/conf/vhost.confなど)

```
CustomLog "|/usr/sbin/cronolog /var/www/www.example.com/logs/access_log-%Y%m%d ¥
          --link=/var/www/www.example.com/logs/access_log ¥
     --prev-symlink=/var/www/www.example.com/logs/yesterday.log" combined
ErrorLog  "|/usr/sbin/cronolog /var/www/www.example.com/logs/www_error_log-%Y%m%d"
```

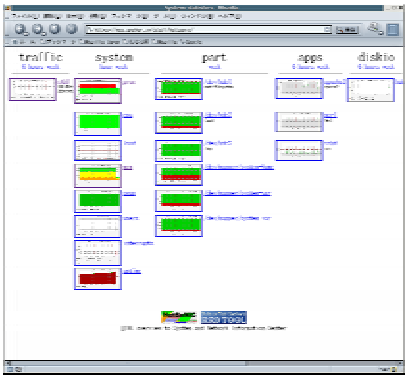
•Webdruidにおけるログの指定(/var/www/www.example.com/conf/webdruid.confなど)

```
LogFile         /var/www/www.example.com/logs/yesterday.log
```

- WebdruidはAwstatsのように、日付フォーマットをファイルには指定できないので、上のようなトリックを使う。--linkも指定しているのは、--linkがないと、--prev-symlinkが動作しないため。

サーバの状態把握

- sysstat
  - システムリソース監視の定番。
- HotSaNIC
  - RRDtoolを使って、サーバのリースをグラフ化



sar

- プロセスの生成個数表示(プロセス数/秒)

```
# sar -c
```

- コンテキストスイッチの切り替え頻度

```
# sar -w
```

- メモリーの利用率

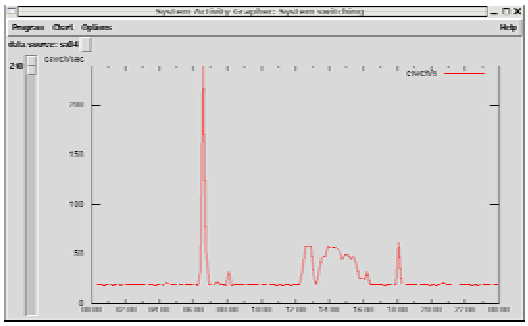
```
# sar -r
```

- CPUの使用率

```
# sar -u
```

- プロセスのキュー

```
# sar -q
```



HotSaNIC

- Linux-2.6系に対応しているのはCVS版

CVSから最新版を取得
$ cvs -d :pserver:anonymous@cvs.sourceforge.net:/cvsroot/hotsanic login
$ cvs -z3 -d :pserver:anonymous@cvs.sourceforge.net:/cvsroot/hotsanic co HotSaNIC

 # mv HotSaNIC /usr/local
 # ./setup.pl

設定ファイル
/usr/local/HotSaNIC/var/settings/*

- データを蓄積･グラフ化するためのプログラムを起動

```
$ cd /usr/local/HotSaNIC
# ./rrdgraph start
```

- **日々の運用**
  - **定常状態の把握**（sar, HotSaNIC**などを活用**）
  - **ログの監査**(logwatch**などを活用**)
  - **セキュリティホールへの対応**
    - RedHat**系**
      - yum**の利用**
    - Debian**系**
      - cron-apt**の利用**

- **標準インストールの構成は避ける**
  - http://www.example.com/cgi-bin/awstats.pl
  - http://www.example.com/cgi-bin/cacti…

    など。

スクリプトキディの格好の標的になることも。

※）もちろん変えたからといって安心はできないが、単純なアタックなら、ディレクトリ構成を変えるだけでも効果あり。

セキュリティに役立つツール

- **ネットワークにおけるアクセス制御**
  - tcp_wrappers
  - iptables(Netfilter)
- **システム改ざん検知**
  - Tripwire
  - AIDE
- **その他**
  - **ログ！**

最後に……

- **サーバを管理するのは大変**
  - **日々のリソース監視**
  - **セキュリティ情報の収集**

それでも昔、高嶺の花だったUNIXが身近にある幸せ

是非、楽しんでください☺