

Apache on CLUSTERPRO for Linux HowTo

1 はじめに

この文章は、Linuxの標準 HTTP サーバである Apache と CLUSTERPRO for Linux を組み合わせてフェイルオーバークラスタを構成するのに必要な情報を記述したものです。HTTP サーバをフェイルオーバークラスタ化することにより、単一 Linux サーバでは達成し得ない可用性・信頼性の高い HTTP サーバを構築することができます。

この文章を書くにあたって次のバージョンのソフトウェアを使用しました。この他のバージョンのソフトウェアを使用した場合でも、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。

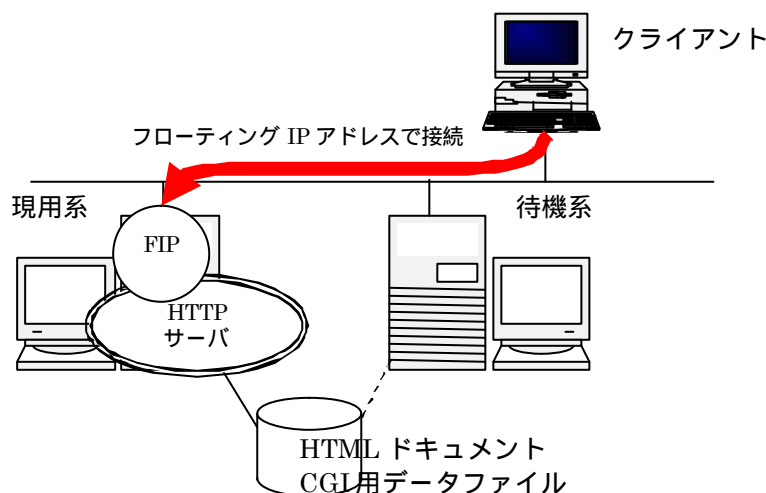
- ・ Red Hat Enterprise Linux AS release 3
- ・ CLUSTERPRO SE for Linux Ver 3.0
- ・ Apache-2.0.46-25

2 責任範囲

この文章は、Apache をクラスタ化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 どのようにクラスタ化するか

- (1) HTML ドキュメント、および CGI で用いるデータファイルを共有ディスク上のパーティションに配置・設定を行うことで、HTTP データの引き継ぎを行います。これによって、現用系サーバの障害時に、待機系サーバで HTTP サービスの提供が可能になります。
- (2) この文章で扱っているのは、HTTP サーバが 1 サーバあたり 1 つのみ動作する、片方向スタンバイシステムの構築例です。



図は HTTP サーバを 2 台のサーバでクラスタ化したときの構成イメージです。CLUSTERPRO のフェイルオーバーグループには、フローティング IP アドレス、HTML ドキュメントを格納する切替パーティション、Apache の起動スクリプト、終了スクリプトを設定します。また、Apache が起動するサーバ上では、HTML ドキュメントを置く共有ディスク内のディレクトリを設定します。これによって、現用系サーバに障害が発生したときに、待機系でフェイルオーバーグループを起動することで、クライアントからはサービスを提供するサーバがどちらかを気にせずに HTTP サービスを享受できます。

4 事前準備

1. ユーザ nobody のユーザ ID、グループ nobody のグループ ID が、クラスタを構成する全てのサーバで同一であることを確認してください
(Apache が nobody のユーザアカウントを用いるため)

```
id nobody
uid=99(nobody) gid=99(nobody) 所属グループ=99(nobody) *1
```

*1 uid:ユーザ ID gid:グループ ID 所属グループ:グループ ID
それぞれの登録 ID が、おのこのサーバで同一である必要があります。

2. HTML ドキュメントを共有ディスク内に置く必要があります。
まず、ディレクトリを作成します。
以下に、共有ディスクのマウントポイントを /mnt/sdb2 に置いた場合のコマンド入力例を示します。

```
umask 022 .....ディレクトリ初期マスク値設定
( ディレクトリ所有者以外は書き込み不可 )
mkdir /mnt/sdb2/httpd .....httpd 用ディレクトリ
mkdir /mnt/sdb2/httpd/html ..... HTML 用ディレクトリ
mkdir /mnt/sdb2/httpd/cgi-bin ..... CGI 用ディレクトリ
mkdir /mnt/sdb2/httpd/icons .....アイコン用ディレクトリ
```

次に、HTML ドキュメントの各ファイルを、

- /mnt/sdb2/httpd/html
- /mnt/sdb2/httpd/cgi-bin
- /mnt/sdb2/httpd/icoens

に配置します。

5 クラスタ化のための設定

5.1 Apache コンフィグファイルの編集

クライアント側から HTML データの要求を受けたときに、共有ディスク内のファイルが読み込まれるようにするため、httpd.conf の変更が必要です。
なお、設定はクラスタを構成する全てのサーバで行ってください。

以下説明内で、網掛け斜体字の部分は、環境に合わせて変更してください。

/etc/httpd/conf 内の設定ファイルを編集します。

以下に設定例を示します。これを参考に httpd.conf の設定ファイルを編集してください。
(注 : 本サンプルをそのまま使用しないでください)

なお、ここでは以下の環境を想定しています。

共有ディスクパーティションのマウントポイント : /mnt/sdb2
フローティング IP のサーバ名 : www.clusterpro.co.jp

```
ServerName www.clusterpro.co.jp
DocumentRoot "/mnt/sdb2/httpd/html"

AddHandler cgi-script .cgi

<Directory "/mnt/sdb2/httpd/html">
    Options Indexes FollowSymLinks
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>

Alias /icons/ "/mnt/sdb2/httpd/icons/"
<Directory "/mnt/sdb2/httpd/icons">
    Options Indexes MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>

ScriptAlias /cgi-bin/ "/mnt/sdb2/httpd/cgi-bin/"
<Directory "/mnt/sdb2/httpd/cgi-bin">
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
</Directory>
```

5.2 CLUSTERPRO の設定

5.2.1 ファイルオーバグループの作成

HTTP サービスを実現するファイルオーバグループを一つ作成します。
ファイルオーバグループに設定するリソースは次のとおりです。

- ・ ディスクリソース
- ・ フローティングIP リソース
- ・ exec リソース

5.2.2 スクリプトの作成

クラスタの起動時やファイルオーバ時に、Apacheのサービスを適切に起動・停止させるために、以下のような手順で、ファイルオーバグループの開始スクリプト、終了スクリプトを編集する必要があります。

1. CLUSTERPRO のトレッキングツール内のグループを選択します。
2. リソース一覧で exec リソースを選択します。
3. 「リソースのプロパティ」ウインドウで、「詳細」タブを選択します。
4. スクリプト一覧から、start.sh と stop.sh を選択して編集します。
5. スクリプトを保存し、「リソースのプロパティ」ウインドウで「OK」を選択し設定を保存します。

ログの出力が必要な場合は CLUSTERPRO SE for Linux Ver3.0トレッキングツール編（グループリソース）を参照してください。

以下に、スクリプトの一例を示します。

下線が引かれたところが、デフォルトのスクリプトから追加された部分です。

start.sh

```
#!/bin/sh
#*****
#*          s t a r t . s h          *
#*****

if [ "$CLP_EVENT" = "START" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "NORMAL1"
    echo "/etc/rc.d/init.d/httpd start" ..... *1
    /etc/rc.d/init.d/httpd start ..... *2
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "NORMAL2"
    else
      echo "ON_OTHER1"
    fi
  else
    echo "ERROR_DISK from START"
  fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "FAILOVER1"
    echo "/etc/rc.d/init.d/httpd start" ..... *1
    /etc/rc.d/init.d/httpd start ..... *2
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "FAILOVER2"
    else
      echo "ON_OTHER2"
    fi
  else
    echo "ERROR_DISK from FAILOVER"
  fi
else
  echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

stop.sh

```
#!/bin/sh
#*****
#*                stop.sh                *
#*****

if [ "$CLP_EVENT" = "START" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "NORMAL1"
    echo "/etc/rc.d/init.d/httpd stop" .....*1
    /etc/rc.d/init.d/httpd stop .....*3
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "NORMAL2"
    else
      echo "ON_OTHER1"
    fi

  else
    echo "ERROR_DISK from START"
  fi
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
  if [ "$CLP_DISK" = "SUCCESS" ]
  then
    echo "FAILOVER1"
    echo "/etc/rc.d/init.d/httpd stop" .....*1
    /etc/rc.d/init.d/httpd stop .....*3
    if [ "$CLP_SERVER" = "HOME" ]
    then
      echo "FAILOVER2"
    else
      echo "ON_OTHER2"
    fi

  else
    echo "ERROR_DISK from FAILOVER"
  fi
else
  echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

*1 ログとして記述するには、このように記述します。

*2 httpd を起動します。

*3 httpd を停止します。

5.2.3 注意事項

CGI 等を用いた動的なページが存在する場合には、クライアント側から受信したデータなどを保存するファイルを共有ディスク上に置く必要があります。もしそうっていない場合は、HTML ドキュメントを書き換えるなどして対応してください。

6 動作が確認されたページ

下記のようなページで、動作を確認しました。

1. 静的なページ（サーバ側からクライアントに情報を提供するだけのページ）
（Apacheのデフォルトページで動作を確認しました）
2. CGIを用いたページ
下記のようなカウンタプログラムで、正常動作、フェイルオーバー時の適切なデータの引継ぎを確認しました。

sample.cgi

```
#!/usr/bin/perl

print "Content-type: text/html\n";
print "\n";

open(IN,"data/number.dat");
$number = <IN> + 0;
$number++;
close(IN);
open(OUT, ">data/number.dat");
print OUT $number;
close(OUT);

print "<html>\n";
print "$number\n";
open(IN, "/hostname");
print <IN>;
close(IN);
print "</html>\n";
```

3. Cookieを用いたページ
下記のようなカウンタプログラムで、正常動作時、フェイルオーバー時における動作確認を行いました。

cookie.html

```
<html><head>
<title>cookie sample</title></head>
<body>

<script language="JavaScript">

<!--
cookie = parseInt(document.cookie.substring(5,8));
if (isNaN(cookie)) {
    cookie=0;
} else {
    cookie++;
}
document.write(cookie + "<BR>");
document.cookie = "NAME=" + cookie;
// -->
</script>
</body>
</html>
```