

OpenSolaris 3分クッキング

☆レシピ 第10巻☆

Zone のクローニング(複製)

Solaris 3分クッキング: Zone クローニングデモ

今夜の3分クッキングは、OpenSolaris を使って、サーバー上に存在する ローカル Zone の複製のデモを行います。セミナーでは簡単なデモだけ実演いたします。詳細はこちらのレシピを参照ください。

Zone クローニングの概要

OpenSolaris では、ローカル Zone の環境は、ZFS ファイルシステム上に構築されます。ZFS の特性から、Zone 環境の複製(クローニング)は、非常に高速に、しかもディスク容量をあまり消費せず作成することができます。Zone の複製後、複製した Zone のそれぞれのシステム構成を変更するだけで、それぞれの Zone は独立した仮想環境として利用することができます。開発環境の構築や、一台のマシンで仮想 LAN を構築したりとか、とっても便利です。

デモ環境

実際の現場とかではありえない環境ですが、ミニノート PC をサーバーに見立て、AMP とフリーのブログソフト、Wordpress が構築済みのローカル Zone の複製(3 個)を約 3 分で行います。

サーバー環境

PC	HP2133 Via C7-M 1.2GHz
メモリーサイズ	2GB
ディスクサイズ	8GB
OS	OpenSolaris Dev Release snv_108

マスター・ローカル Zone (test1) の構成内容

- OpenSolaris で Zone をインストールしただけの最小構成 (約 240MB)
- %pfexec pkg install amp で、AMP (Apache, mysql, php) を追加 (+ 約 516MB)
- wordpress の zip ファイル解凍用に unzip も追加
%pfexec pkg install SUNWunzip
- WordPress 日本語版を <http://ja.wordpress.org/> からダウンロード
/var/apache2/2.2/htdocs の下に展開 (+ 約 4MB)
- Network には、b105 から入った、Crossbow の機能を使って vnuc を使用

デモの目的

今回のデモは全部スクリプトの実行で行います。ブログサーバーのマスター Zone (test1)の複製を、test2, test3, test4 として3つ作成し、それぞれが独立したブログサーバとして機能することを確認します。Zone の複製が、高速にしかも資源(ディスクやメモリ)をあまり消費せずにできることをご覧ください。

デモ環境構築の解説

今回のデモ環境構築までを、詳しく解説致します。

手順1 : CrossBow を使って仮想スイッチと仮想 NiC を構築

参考 URL: <http://opensolaris.org/os/project/crossbow/>
<http://blogs.sun.com/sayama/entry/crossbow/> 佐山さんのブログ

仮想スイッチ (etherstub0) 一個と、それにつなげた仮想 nic (vnic100 ~)を適当な数作っておきます。

```
hara@opensolaris:~$ pfexec dladm create-etherstub etherstub0
hara@opensolaris:~$ pfexec dladm create-vnic -l etherstub0 vnic100
hara@opensolaris:~$ pfexec dladm create-vnic -l etherstub0 vnic101
hara@opensolaris:~$ pfexec dladm create-vnic -l etherstub0 vnic102
hara@opensolaris:~$ pfexec dladm create-vnic -l etherstub0 vnic103
hara@opensolaris:~$ pfexec dladm create-vnic -l etherstub0 vnic104
hara@opensolaris:~$ dladm show-link
LINK          CLASS      MTU      STATE    OVER
rge0          phys      1500    up       --
wpi0          phys      1500    down    --
etherstub0    etherstub 9000    unknown --
vnic100       vnic      9000    up       etherstub0
vnic101       vnic      9000    up       etherstub0
vnic102       vnic      9000    up       etherstub0
vnic103       vnic      9000    up       etherstub0
vnic104       vnic      9000    up       etherstub0
hara@opensolaris:~$
```

グローバルゾーン上で、vnic100 に IP を振ります。

```
hara@opensolaris:~$ pfexec ifconfig vnic100 plumb
hara@opensolaris:~$ pfexec ifconfig vnic100 192.168.5.1/24 up
```

NAT 設定して、グローバルゾーンの物理的 nic から構築するローカルゾーンで vnic 経由でインターネットアクセス可能にしておきます。

```
hara@opensolaris:~$ routeadm -u -e ipv4-forwarding
hara@opensolaris:~$ cd /etc/ipf
hara@opensolaris:/etc/ipf$ pfexec vi ipnat.conf
map rge0 192.168.5.0/24 -> 0/32 portmap tcp/udp auto
map rge0 192.168.5.0/24 -> 0/32
hara@opensolaris:/etc/ipf$ pfexec svcadm enable svc:/network/ipfilter:default
```

手順2 : マスターローカル Zone (test1) の構築

zonecfg で、test1 Zone を構成します。通常と違うところは、ip-type=exclusive にすることと、network に仮想 nic(vnic101) を設定するところ、IP アドレスの設定は、ここでは必要ありません。

```
hara@opensolaris:~$ pfexec zonecfg -z test1 'create;set autoboot=true; set
zonepath=/export/zones/test1;set ip-type=exclusive;add net;set
physical='vnic101';end;verify;commit;exit'
```

Test1 Zone のインストール

```
hara@opensolaris:~/zones# zoneadm -z test1 install
A ZFS file system has been created for this zone.
  Authority: Using http://pkg.opensolaris.org/dev/.
  Image: Preparing at /export/zones/test1/root ... done.
  Cache: Using /var/pkg/download.
  Installing: (output follows)
DOWNLOAD                PKGS        FILES        XFER (MB)
Completed                53/53       7983/7983    77.47/77.47

PHASE                    ACTIONS
Install Phase            12199/12199
PHASE                    ITEMS
Reading Existing Index   9/9
Indexing Packages        55/55

      Note: Man pages can be obtained by installing SUNWman
Postinstall: Copying SMF seed repository ... done.
Postinstall: Working around http://defect.opensolaris.org/bz/show_bug.cgi?id=741
      Done: Installation completed in 151.125 seconds.

Next Steps: Boot the zone, then log into the zone console
             (zlogin -C) to complete the configuration process
hara@opensolaris:~/zones#
```

スーパーユーザで、`zlogin -C test1` でコンソールをとり、システム構成を行います。Vnic101 の構成で、IP アドレス 192.168.5.2 を設定します。システム構成が完了したら、test1 に root でログインし、インターネットアクセスのための設定を行います。見るのは以下の2つのファイル

`/etc/nsswitch.conf` (hosts と ipnodes に dns を追加)

hosts: files dns

ipnodes: files dns

`/etc/resolv.conf` (nameserver にグローバルゾーンと同じ DNS server のアドレスを指定します)

nameserver 192.168.1.1 ← これは例です。

インターネットへのアクセスが可能になったら、opensolaris のリポジトリから、AMP クラスタ (apache,mysql, php) をインストールします。

```
root@test1:~# pkg install amp
DOWNLOAD                PKGS        FILES        XFER (MB)
Completed                21/21       2601/2601    108.44/108.44

PHASE                    ACTIONS
Install Phase            3473/3473
PHASE                    ITEMS
Reading Existing Index   9/9
Indexing Packages        76/76
root@test1:~#
```

ついでに、SUNWunzip や SUNWwget もインストールします。<http://ja.wordpress.org> からブログソフト wordpress の日本語版もダウンロードします。

```
root@test1:~# pkg install SUNWunzip
```

```

root@test1:~# pkg install SUNWwget
DOWNLOAD                                PKGS      FILES      XFER (MB)
Completed                                1/1       41/41      0.60/0.60

PHASE                                    ACTIONS
Install Phase                            131/131
PHASE                                    ITEMS
Reading Existing Index                    9/9
Indexing Packages                          78/78
root@test1:~# cd /tmp
root@test1:~# wget http://ja.wordpress.org/wordpress-2.7.1-ja.zip
--22:20:50--  http://ja.wordpress.org/wordpress-2.7.1-ja.zip
           => `wordpress-2.7.1-ja.zip'
Resolving ja.wordpress.org... 72.233.56.138, 72.233.56.139
Connecting to ja.wordpress.org|72.233.56.138|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2,080,690 (2.0M) [application/octet-stream]

100%[=====>] 2,080,690    340.13K/s   ETA 00:00

22:20:58 (294.13 KB/s) - `wordpress-2.7.1-ja.zip' saved [2080690/2080690]
root@test1:~#

```

Test1 上で、apache と mysql を起動します。設定方法は以下のとおりです。

```

root@test1:~# /usr/sbin/svccfg import /var/svc/manifest/application/database/mysql.xml
root@test1:~# /usr/sbin/svccfg import /var/svc/manifest/network/http-apache22.xml
root@test1:~# /usr/sbin/svcadm enable network/http:apache22
root@test1:~# /usr/sbin/svcadm enable database/mysql:version_50
root@test1:~#

```

```

root@test1:~# /usr/mysql/bin/mysql -u root
...
mysql> create database wordpress;
Query OK, 1 row affected (0.00 sec)

mysql> grant all privileges on wordpress.* to wp@localhost identified by 'wp';
Query OK, 0 rows affected (0.00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)

mysql> quit;
Bye
root@test1:~#

```

Wordpress も設定します。

参考 URL: <http://www.netcube.ne.jp/wiki/cms/wordpress>
http://blogs.sun.com/natarajan/entry/famous_5_minute_installation_of

```

root@test1:~# cd /var/apache2/2.2/htdocs
root@test1:/var/apache2/2.2/htdocs# unzip /tmp/wordpress-2.7.1-ja.zip
root@test1:/var/apache2/2.2/htdocs# cd wordpress
root@test1:/var/apache2/2.2/htdocs/wordpress# cp wp-config-sample.php wp-
config.php
root@test1:/var/apache2/2.2/htdocs/wordpress# vi wp-config.php

```

```
define('DB_USER', 'usernamehere'); の箇所を define('DB_USER', 'wordpress');  
define('DB_USER', 'usernamehere'); の箇所を define('DB_USER', 'wp');  
define('DB_PASSWORD', 'yourpasswordhere'); の箇所を define('DB_PASSWORD', 'wp');
```

これで必要な設定は終わりです。

グローバルゾーンから、<http://192.168.5.2/wordpress/> にアクセスして、下の画面が出てきたら設定は成功です。



デモの概要

以上で、Zone 複製のための準備は完了ですが、ここまでで約30分はかかるので、デモではここまでの分は省略します。

デモでは、構築した test1 Zone(Wordpress ブログサーバー) の複製を3つ作ります。手順はいたって簡単です。

- 1) 複製する Zone の構成をしておく(zonecfg test2 ~4, test1 と同様に vnic102~ vnic104 を設定)
- 2) pfexec zoneadm -z test2 clone test1 でクローンを作成。 test3, test4 も同様
- 3) 作成したクローンを起動し、システム構成を行う

デモでは、これらの設定を、すべてスクリプトで行っています。尚、このデモスクリプトは、エバンジェリストの加藤氏が作ったすばらしいものです。

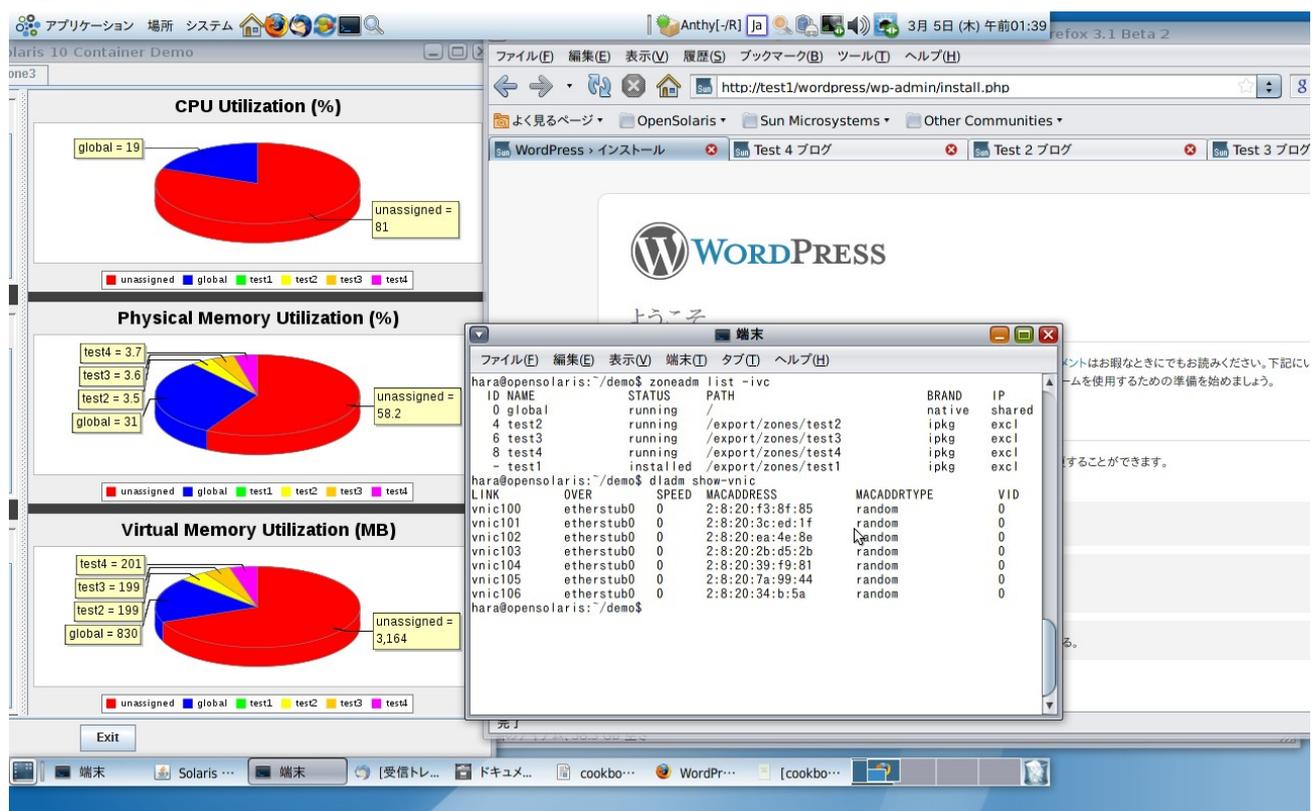
```

hara@opensolaris:~# zfs list
NAME                                USED    AVAIL   REFER  MOUNTPOINT
rpool                                7.07G   570M   74.5K   /rpool
rpool/ROOT                           5.24G   570M    18K   legacy
rpool/ROOT/opensolaris                20.1M   570M   4.00G   /
rpool/ROOT/opensolaris-1              5.22G   570M   4.16G   /
rpool/dump                             256M    570M   256M   -
rpool/export                          1.08G   570M    21K   /export
rpool/export/home                     318M    570M    19K   /export/home
rpool/export/home/hara                 318M    570M   318M   /export/home/hara
rpool/export/zones                    786M    570M    19K   /export/zones
rpool/export/zones/test1              786M    570M    21K   /export/zones/test1
rpool/export/zones/test1/ROOT         786M    570M    18K   legacy
rpool/export/zones/test1/ROOT/zbe     786M    570M   786M   legacy
rpool/swap                             512M   1.05G   6.29M   -

```

ディスクの使用量も見てみましょう。Test 1 構築時に 786MB 使用しましたが、3つ複製を作って走らせても、実際には 30-40MB ぐらいしか増えてません、これは ZFS のスナップショットでファイルシステムの複製を行っているため、複製した各 Zone で新たに使った差分の分しか増えないためです。また Zone の複製をたくさん走らせていても、メモリ使用量は思ったより増えません、資源を有効に使えるという点でもすばらしい機能です。

下図は、デモの様子



作成: 原口