

FreeBSDでMBRパーティションのHDDからGPTパーティションのHDDへフルコピーする

2015/07/26

/tmp のコピーもやったほうがよいも知れない□/tmp のスティッキービットが落ちてしまうから。自分でスティッキービットを設定できるなら/tmpコピーしなくてもかまわない。

2015/07/23

作成完了

エラーが出始めた 320GB HDD の環境を新品の 4TB HDD に移動(コピー)する方法のメモ書き。

注意事項等々

今回の処理はシングルユーザーモードで行っていないので、たとえばWebサーバを運用したままだとログが完全に引き継ぎできません。またメールスプールなんかも、タイミングで一部引継ぎできなかったりすることも考えられます。完全引継ぎをしたければ、シングルユーザーモードで実行するべきでしょう。

手順概要

大雑把には以下の手順を踏みます。

- 4TB HDD にGPTパーティションを作成
- 320GB HDD から 4TB HDD へパーティション毎のコピー
- 4TB HDD の /etc/fstab 書き換え
- HDD を入れ替えして再起動

4TB HDD にGPTパーティションを作成

- [2.6. ディスク領域の割り当て - FreeBSD ハンドブック](#)
- [第15回 FreeBSD勉強会へ行ってきました - 完熟トマト](#)
- [FreeBSDでGPTを使う \(UNIX \) - アルファのブログ - Yahoo!ブログ](#)

上記ページを参考に 4TB HDD にGPTパーティションを作成します。

現在のパーティションの確認

作成する前に、現在の 320GB HDD のパーティション構成を持っている /etc/fstab を確認します。

[fstab](#)

#	Device	Mountpoint	FStype	Options	Dump
Pass#					

/dev/ada0s1b 0	none	swap	sw	0
/dev/ada0s1a 1	/	ufs	rw	1
/dev/ada0s1e 2	/tmp	ufs	rw	2
/dev/ada0s1f 2	/usr	ufs	rw	2
/dev/ada0s1g 2	/var	ufs	rw	2
/dev/ada1p1 2	/home	ufs	rw	2
/dev/acd0 0	/cdrom	cd9660	ro,noauto	0
proc 0	/proc	procfs	rw	0

デバイス ada0 は環境移動元の 320GB HDDです。また、デバイス ada0 の後ろに sXX が続くことから MBRパーティションで作成されているらしいことがわかるかとおもいます。

デバイス ada1 は /home 専用のGPTパーティションを作成した 2TB HDD となっています。

お次はサイズ。以下は gpart show ada0 の結果。

```
root@amanda:~ # gpart show ada0
=>      63 625142381  ada0  MBR  (298G)
        63 625137282   1  freebsd [active] (298G)
        625137345      5099   - free - (2.5M)

root@amanda:~ # gpart show ada0s1
=>      0 625137282  ada0s1  BSD  (298G)
        0 2097152   1  freebsd-ufs (1.0G)
        2097152 16777216   2  freebsd-swap (8.0G)
        18874368 8388608   4  freebsd-ufs (4.0G)
        27262976 16777216   5  freebsd-ufs (8.0G)
        44040192 314572800  6  freebsd-ufs (150G)
        358612992 266524290  7  freebsd-ufs (127G)

root@amanda:~ #
```

まとめると以下のような対応になっています。

fstab	gpart表示	サイズ	マウントポイント
ada0s1b	ada0s2	8.0GB	-(swap)
ada0s1a	ada0s1	1.0GB	/
ada0s1d	ada0s4	4.0GB	-(旧 /var)
ada0s1e	ada0s5	8.0GB	/tmp
ada0s1f	ada0s6	150GB	/usr
ada0s1g	ada0s7	127GB	/var (旧 /usr/ports)

「/var でかくな？」...はい、その通り。実は /dev/ada0s1d を /var にあてていたのですがHDDエラーがこのスライス(パーティション)で出ているようなので /usr/ports に使っていた /dev/ada0s1g を /var 用に割り当てしなおしました。

この構成に従って 4TB HDD にパーティションを作成します。

外付け接続機器で 4TB HDD を接続

私はSATA-USBの変換ケーブルを使い、USB外付けディスクとして接続しました。ギリギリまでPC筐体を開けたくなかったからです。使ったのは [Groovy HDD簡単接続セット UD-505SA](#) です。

もちろんPC筐体を開けてHDDをマザーボードの空きコネクタに接続してもかまいません。

接続したら `dmesg` コマンドで 4TB HDD のデバイス名を確認します。

```
da2 at umass-sim2 bus 2 scbus9 target 0 lun 0
da2: <WDC WD40 EZRX-00SPEB0 0016> Fixed Direct Access SCSI-4 device
da2: Serial Number 0000000000033
da2: 40.000MB/s transfers
da2: 3815447MB (7814037164 512 byte sectors: 255H 63S/T 486401C)
da2: quirks=0xa<NO_6_BYTE,4K>
```

表示最後に接続したHDDの表示が出ていると思います。

この例だとデバイス名 da2 と認識されています。マザーボードのSATAコネクタに繋いだ人は ada1とかada2とかになるかと。

GPTパーティションの作成

以下のコマンドラインで 4TB HDD にGPTパーティションを作成します。MBRパーティションでは2TBを超える容量のHDDに対応できません。

```
gpart create -s GPT da2
gpart bootcode -b /boot/pmbr da2
gpart add -s 64K -b 34 -t freebsd-boot da2 ( または gpart add -s 64K -t
freebsd-boot da2 )
gpart bootcode -p /boot/gptboot -i 1 da2
gpart add -s 10GB -t freebsd-ufs -l amandaRootfs da2
gpart add -s 32GB -t freebsd-swap -l amandaSwap da2
gpart add -s 400GB -t freebsd-ufs -l amandaTemp da2
gpart add -s 400GB -t freebsd-ufs -l amandaVar da2
gpart add -t freebsd-ufs -l amandaUsr da2
```

1行目でエラーが出るなら `gpart destroy da2` で既存のスキーマを消してやり直してください。デバイス間違えないように。

2行目でGPTパーティションのHDDから起動するためのブートコードインストール指示BIOSが `freebsd-boot` パーティションを読むためのコードになります。

3行目で freebsd-boot パーティションを定義します。オプション -b 34 を付けてるのは、参照したサイトではブロック番号34番から freebsd-boot パーティションを作成しているようなのでこれに従った結果です。ただ、コマンド実行時に4096バイトのアライメントが云々の警告が出ます[AFTで4KセクタのHDDだとどこかで認識しているためかもしれません。 オプション -b 34 を付けないとブロック番号40番から freebsd-boot パーティションが作成されます。

4行目で freebsd-boot パーティションに /boot/gptboot を書き込みます。これが実行されルートファイルシステムがマウントされるようになります。

5行目で ルート(/)にマウントする freebsd-ufs パーティションを作成します。FreeBSDの /boot/gptboot は freebsd-boot パーティションの次に見つかる freebsd-ufs パーティションを読み出し対象とするので freebsd-boot パーティションの次に作成しました。

6行目で スワップ、7行目で /tmp、8行目で /var、9行目で /usr のパーティションを作成しました。-s オプションでサイズを指定しないと、「割り当て可能な全サイズ」の意となります。

もし /boot/gptboot や /boot/pmbr が見つからなければ LiveCD 辺りから抜き出してくるのがお手軽かも。

結果はこんな感じ。

```
root@amanda:~ # gpart show da2
=>      34  7814037101  da2  GPT  (3.6T)
        34           128    1  freebsd-boot  (64K)
        162           6     - free -  (3.0K)
        168    20971520    2  freebsd-ufs  (10G)
        20971688    67108864  3  freebsd-swap (32G)
        88080552    838860800  4  freebsd-ufs  (400G)
        926941352    838860800  5  freebsd-ufs  (400G)
        1765802152  6048234976  6  freebsd-ufs  (2.8T)
        7814037128           7     - free -  (3.5K)

root@amanda:~ # ls /dev/da2*
/dev/da2      /dev/da2p1    /dev/da2p2    /dev/da2p3    /dev/da2p4
/dev/da2p5    /dev/da2p6
root@amanda:~ #
```

MBRパーティションの構成にあわせて作成し、各パーティションのサイズも増えています。あとは各 freebsd-ufs パーティションを newfs コマンドでフォーマットしておきます。

```
newfs /dev/da2p2
newfs /dev/da2p4
newfs /dev/da2p5
newfs /dev/da2p6
```

320GB HDD から 4TB HDD へパーティション毎のコピー

- [FreeBSD QandA 1859](#)
- [FreeBSDサーバ設定手順 » バックアップ&復元\[dump & restore\]](#)
- [FreeBSD 日本語マニュアル検索 of "dump"](#)
- [FreeBSD 日本語マニュアル検索 of "restore"](#)

上記ページを参考にデバイス `ada0` からデバイス `da2` にコピーを行います。

マウントポイントの作成とパーティションマウント

以下のディレクトリを作成します。

```
mkdir /mnt/root
mkdir /mnt/var
mkdir /mnt/usr
```

作成したディレクトリに、デバイス `da2` のパーティションをマウントします。

```
mount /dev/da2p2 /mnt/root
mount /dev/da2p5 /mnt/var
mount /dev/da2p6 /mnt/usr
```

`/tmp` はコピーしません。意味無いし。

フルコピー

`dump` コマンドと `restore` コマンドをパイプで繋いで `ada0` のディレクトリ(パーティション)ダンプ結果を `da2` パーティションへリストアします。

```
dump -0aLC 32 -f - / | ( cd /mnt/root && restore -rf -)
dump -0aLC 32 -f - /var | ( cd /mnt/var && restore -rf -)
dump -0aLC 32 -f - /usr | ( cd /mnt/usr && restore -rf -)
```

- `dump` コマンドの `0` オプションはダンプレベルをフルにする指示です。
- `dump` コマンドの `L` オプションはスナップショット機能を使うことを指示します。スナップショットをダンプするのでダンプ対象がアクセス中(たとえばログファイルで書き込みが行われている途中)でも安全に処理できることとなります。もちろん、スナップショットが取られた後に発生した更新は反映されません。それが嫌な場合はシングルユーザーモードで(サーバやサービスが停止してログを書き出ししていない状態で)実行する必要があります。
- `dump` コマンドの `f` オプションでファイルに `-` (標準出力)を指定します。
- `restore` コマンドの `r` オプションでカレントディレクトリにリストアを行います。
- `restore` コマンドの `f` オプションでファイルに `-` (標準入力)を指定します。

エラーがなければ `restore` コマンドにより作成されたファイル `restoresymtable` を削除します。

```
rm /mnt/root/restoresymtable
rm /mnt/var/restoresymtable
rm /mnt/usr/restoresymtable
```

まとめると、こんなコピーを実施したことになります。

コピー元 <code>ada0</code>	コピー先 <code>da2</code>
<code>ada0s1a</code>	<code>/</code> <code>da2p2</code> <code>/mnt/root</code>
<code>ada0s1g</code>	<code>/var</code> <code>da2p5</code> <code>/mnt/var</code>

コピー元 ada0	コピー先 da2
ada0s1f /usr	da2p6 /mnt/usr

4TB HDD の /etc/fstab 書き換え

ファイル /mnt/root/etc/fstab は現在の /etc/fstab と同じ内容になっています。この /mnt/root/etc/fstab を以下のように書き換えました。

fstab

```
# Device          Mountpoint      FStype  Options      Dump
Pass#
/dev/ada0p3       none            swap    sw           0
0
/dev/ada0p2       /               ufs     rw           1
1
/dev/ada0p4       /var            ufs     rw           2
2
/dev/ada0p5       /tmp            ufs     rw           2
2
/dev/ada0p6       /usr            ufs     rw           2
2
/dev/ada1p1       /home           ufs     rw           2
2
/dev/acd0         /cdrom          cd9660  ro,noauto    0
0
proc              /proc           procfs  rw           0
0
```

現在のデバイス ada0 はマザーボードのコネクタから除去され、デバイス da2 のHDDが接続されます。このデバイス da2 はおそらくPC再起動で ada0 と認識されるでしょう。そして /mnt/root/etc/fstab は /etc/fstab として参照されることになります。

Device/Slice (Old)	Device/Partision (New)	Mountpoint
/dev/ada0s1b	/dev/ada0p3	none (swap)
/dev/ada0s1a	/dev/ada0p2	/
/dev/ada0s1g	/dev/ada0p4	/var
/dev/ada0s1e	/dev/ada0p5	/tmp
/dev/ada0s1f	/dev/ada0p6	/usr

HDD を入れ替えして再起動

ファイル fstab の書き換えが済んだら、マシンをシャットダウンし、HDDの入れ替えて再起動します。デバイス da2 だったものが デバイス ada0 として認識されました。

```
ada0 at ahcich0 bus 0 scbus0 target 0 lun 0
ada0: <WDC WD40EZR-00SPEB0 80.00A80> ATA-9 SATA 3.x device
ada0: Serial Number WD-WCC4E3CHZ501
```

```
ada0: 600.000MB/s transfers (SATA 3.x, UDMA6, PIO 8192bytes)
ada0: Command Queueing enabled
ada0: 3815447MB (7814037168 512 byte sectors: 16H 63S/T 16383C)
ada0: quirks=0x1<4K>
```

転送速度が40MB/sから600MB/sになってる□SATA速いですねー。

再起動時、もしかするとこんなメッセージが出ているかもしれません。

```
GEOM: ada0: the secondary GPT header is not in the last LBA.
GEOM: diskid/DISK-WD-WCC4E3CHZ501: the secondary GPT header is not in the
last LBA.
```

こちらについては[alaif Laboratory: FreeBSD: ディスクサイズ増加に対応する](#)を参照し、コマンド `gpart recover ada0` で対処できました。

追記

/tmp のフルコピーをやっていなかったせいで問題があるかもしれない。

```
drwxr-xr-x  7 root  wheel          512  7月 26 13:00 /tmp
```

この状態ではroot以外が /tmp に書き込みできないので、たとえばfetchmailなんかがおかしな振る舞いになる□`chmod go+w /tmp` 辺りで書き込み権限を与えておく。

```
drwxrwxrwx  7 root  wheel          512  7月 26 13:06 /tmp
```

ただ、このままだと、誰でも /tmp を削除できてしまうので□`chmod +t /tmp` でスティッキービットを立てる。

```
drwxrwxrwt  7 root  wheel          512  7月 26 13:12 /tmp
```

[技術資料, FreeBSD, GPT, gpart, HDD](#)

From:
<https://wiki.hgotoh.jp/> - 努力したWiki

Permanent link:
<https://wiki.hgotoh.jp/documents/freebsd/freebsd-018>

Last update: **2024/11/01 16:25**

