ZFS Day 2011.10

FreeBSDさんとZFSさん

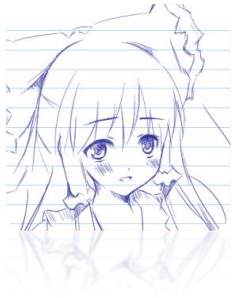
2011/10/15

@team_eririn

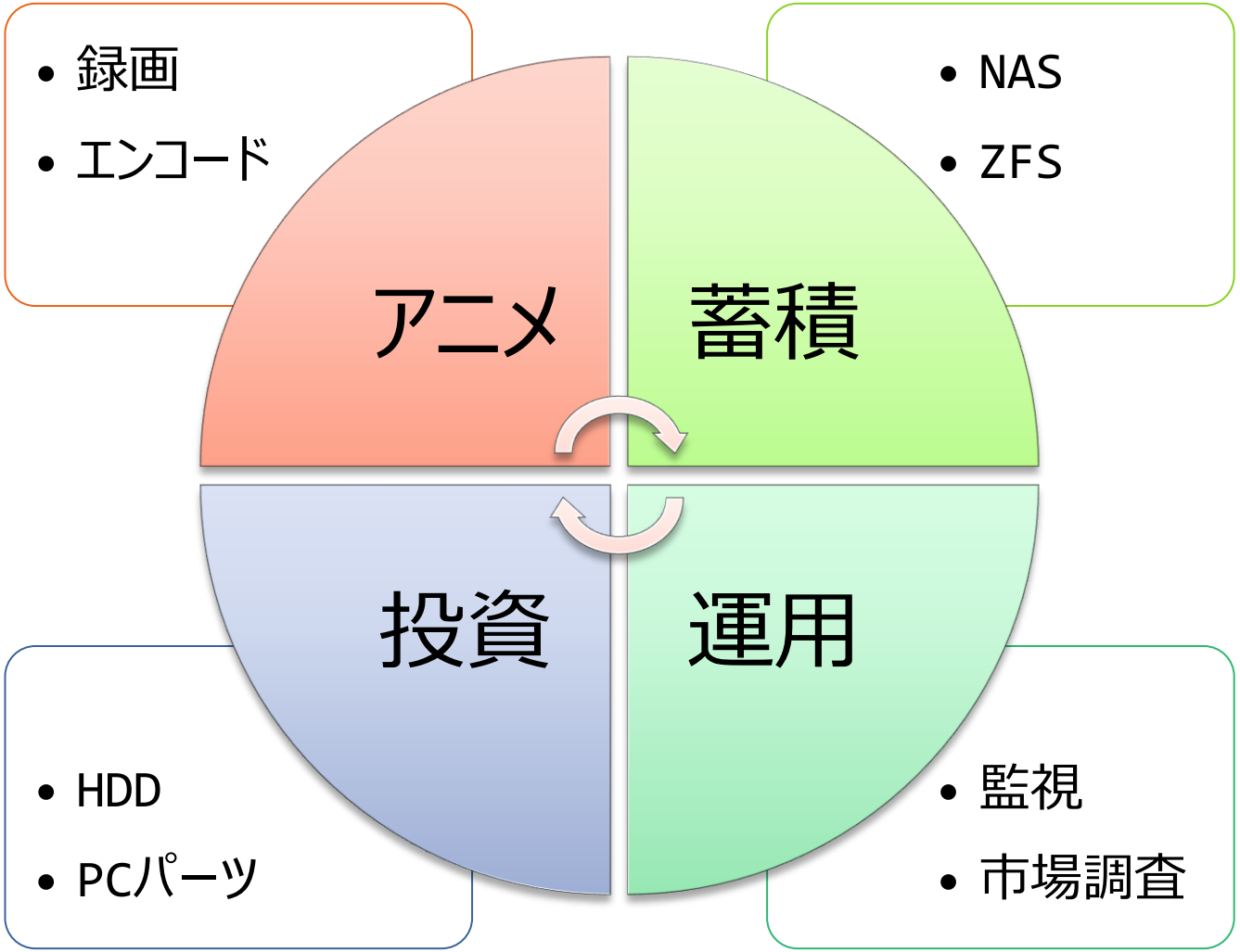
こっそり手直し

- RAID周りの表現の訂正
- 4KiBセクタの記述を検証結果に訂正

自己紹介？

- どちらかというとネットワーク屋さん
- FreeBSDと出会って5年くらい
- Twitter :  team_eririn
- Web : http://www.ainoniwa.net/ssp/

生活スタイル？



あじぇんだー

- 近況報告
- ZFSとUFS
- ZFSとGEOM
- ちょっとだけ運用話

近況報告

FreeBSDから見たZFS

- 標準で搭載されて、普通(*)に使えます
- 初搭載はFreeBSD 7.0-RELEASE(2008/02/26)
- 正式搭載から3年半。今の状況は？

(*)カーネルの再構築とかパッケージのインストールが不要というレベルの意味で

近況報告

- FreeBSD 9.0-RELEASEが登場！
  - ZFSv28までバージョンアップ！
  - システムインストーラがZFSに対応！

近況報告

と、なる予定でした。

が、
FreeBSD 9.0-RELEASEの提供は
少々遅れております。

のんびり待ちましょう。

現状確認

- じゃあ今どうなってるの？
- チラ見してみます。

FreeBSDのZFSバージョン - 1

- 正式リリース編

FreeBSD	ZFS	zpool	備考
7.0-RELEASE	1	6	初搭載
7.1-RELEASE	1	6	
7.2-RELEASE	1	6	
7.3-RELEASE	3	13	8.0-RELEASEからのバックポート
7.4-RELEASE	3	13	
8.0-RELEASE	3	13	
8.1-RELEASE	3	14	
8.2-RELEASE	**4**	**15**	**最新リリース**

使う人はここだけ知ってればいいよ

FreeBSDのZFSバージョン - 2

- 開発版でも良ければ

FreeBSD	ZFS	zpool	備考
8-STABLE	5	28	
9.0-BETA3	5	28	STABLEも可
10-CURRENT	5	28	

- 今後の予定

FreeBSD	ZFS	zpool	備考
8.3-RELEASE	5	28	リリース未定
9.0-RELEASE	5	28	2011/09 リリース予定（でした）

RELEASE ： 正式リリース
STABLE ： 開発リリース版
CURRENT ： 開発途上版

FreeBSDのZFSバージョン - 3

- 最新のZFSを使いたいのは山々だけど、怖いバグもある
  - scrubで発見できないデータ不整合
  - zpool add/removeでほぼ確実にPanic
  - 詳しくはallbsd.orgで
    - http://www.allbsd.org/~hrs/diary/201109.html

- 9.0-BETA2も、試しに読み書きしてたらプール壊れた。
  9.0-BETA3は大丈夫。

現状の選択肢

- 安定ならFreeBSD 8.2-RELEASE
- リリースに備えるならFreeBSD 9.0-BETA3
- ただし、どちらもインストーラはZFS未対応
  - ZFSのみにしたい時は手動インストールか、PC-BSDで
  - FreeBSD Wikiはとても参考になる
    http://wiki.freebsd.org/RootOnZFS

FreeBSDがベースのものは？

- FreeBSDがベースになってるディストリビューションもいくつかある
- それもチラ見。

デスクトップOSの選択肢

PC-BSD 8.2

- ZFS v4, zpool v15
- インストーラがZFS対応（PureZFS可）
- PC-BSD/FreeBSDのインストールが可能
- PC-BSD 9.0-BETA2という手も

DesktopBSD 1.7

- ZFS is not implemented

アプライアンスOSの選択肢

FreeNAS 8.0

- ZFS v4, zpool v15

FreeNAS 0.7.2

- ZFS v3, zpool v13

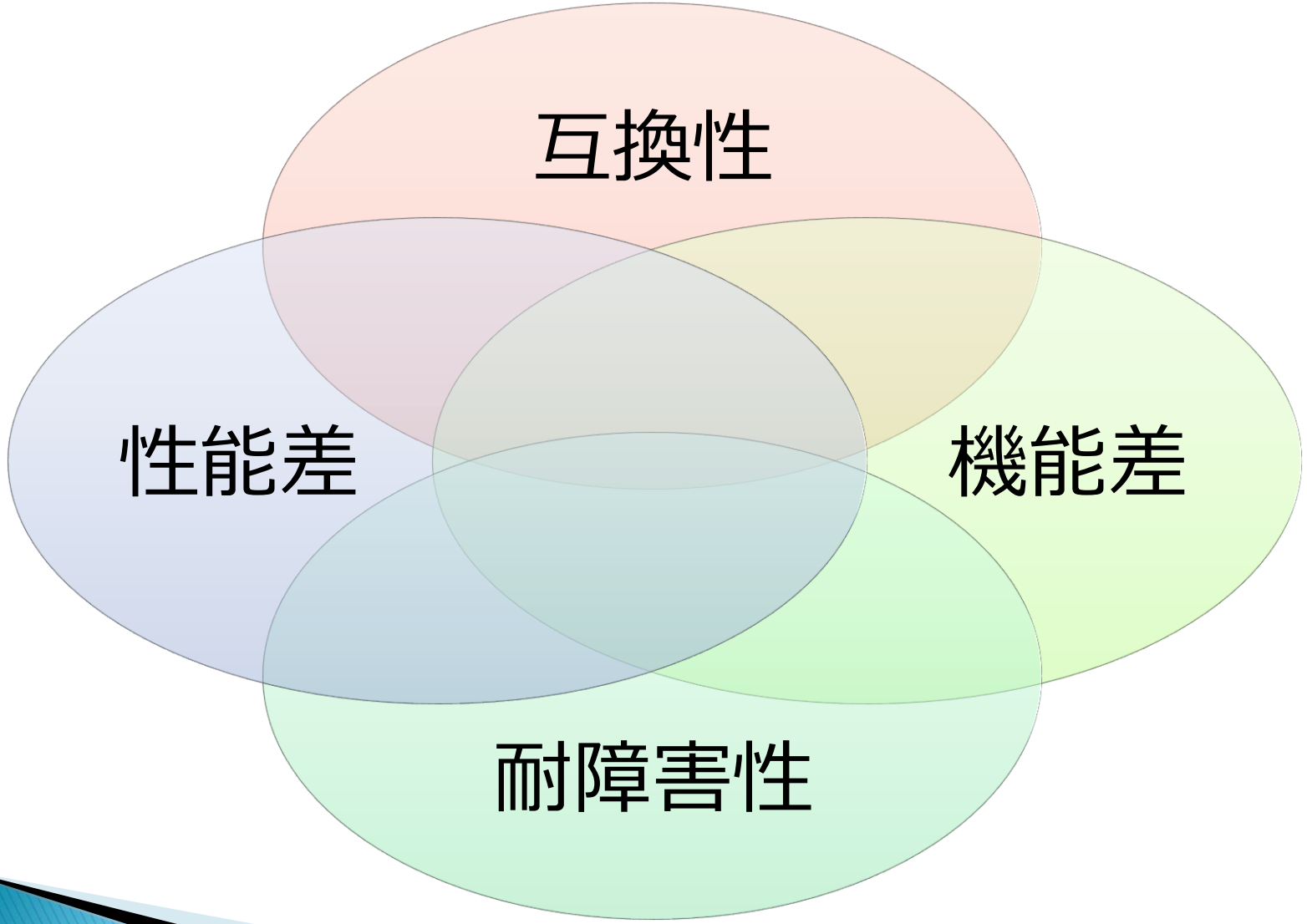
ZFSguru 0.1.8

- ZFS v5, zpool v28

本家以外でのZFSの使用

- ディストリにマージされるのは、まだ少し先かも

なぜバージョンを気にするの？



互換性？

- ZFSは後方互換
- システムのZFSバージョンが上がっても、古いまま使える
- 逆はできない
  - 新しいZFSは、古いZFSシステムでは操作できない
    - FreeNAS 8.0で作ったZFSはFreeNAS 0.7.2で操作できない
  - バックアップ先として古いZFSシステムを指定できない
    - Solaris(v28)からFreeBSD(v15)にsend/recvできない

...etc

機能差？

- バージョンアップで追加機能が山盛り
  - uid/gidベースのquota(v15)
  - RAID-Z3(v17)
  - Logデバイスの取り外し(v19)
  - 重複排除(v21)
  - zfs diff(v28)

    -------- Solarisの壁 --------
  - ZFS暗号化(v30)

    ...etc
- 必要な機能があるバージョンを使いましょう

性能差？

- バージョンアップで性能改善されることも
  - scrubの高速化(v11)
  - snapshot削除の高速化(v26)
  - snapshot作成の高速化(v27)

    -------- Solarisの壁 --------
  - RAID-Z/mirror hybrid allocator(v29)
  - zfs listの高速化(v31)
  - 1MB Block Size(v32)

    ...etc
- とは言え、読み書きの速度自体はそう変わらない

耐障害性？

- 障害からの復帰機能と障害復旧補助ツール
  - zpool import -m（ZIL破損プールの強制import）
  - zpool clear -F（直前のトランザクションのロールバック）
  - zpool import -F（直前のトランザクションのロールバック）
  - zdbの機能差(undocumented)
    ...etc
- データが壊れたら直せないけど、部分的なアクセスや無理矢理なプールの認識はバージョン間で差がある

今後のFreeBSDにおけるZFS

1.v29以降の機能実装

- RAID-Z/mirror hybrid allocator(v29)
  ZFS data set encryption(v30)
  ...etc
- v29以降のソースコードは...？

2.アプライアンスOSへのマージ

- 時間の問題（PC-BSD辺りがやっぱり早そう）

3.バグ潰し

- まだバグ残ってる?

近況報告まとめ

- FreeBSD 9.0-RELEASEを待っててね！
- それからディストリビューションにマージしていくよ！
- v29以降の実装は不透明...
  - あとNetBSDの話できなくてすいません...

ZFSとUFS

ZFSとUFS

1. スタートアップ
2. NFS/Samba/iSCSI
3. スナップショット
4. バックアップ
5. RAID

ZFSとUFS - Startup

- 新しくHDDを買ってきて、まるまる使う

ZFSとUFS - Startup

- UFS(MBR)

```
# dd if=/dev/zero of=/dev/da1 bs=1k count=1
# fdisk -BI da1
# #bsdlabel -Bw da1s1 auto
# #bsdlabel -e da1s1
# newfs -U /dev/da1s1e
# mkdir -p /mnt/ufs-01
# mount /dev/da1s1e /mnt/ufs-01
# echo "/dev/da1s1e /mnt/ufs-01 ufs rw 1 1" >> /etc/fstab
```

- ZFS

```
# zpool create -m /mnt/zfs-01 zfs-01 da2
# echo 'zfs_enable="YES"' >> /etc/rc.conf
```

ZFSとUFS - Start Up

- 出来上がり

```
# df
Filesystem  1K-blocks  Used   Avail Capacity  Mounted on
/dev/da1s1e 1031800       8  949248     0%    /mnt/ufs-01
zfs-01      1007481     316 1007165     0%    /mnt/zfs-01
```

- 明らかにZFSの方がコマンド量は少ないけど...

ZFSとUFS - NFS/Samba/iSCSI

- NAS/SANとして使い始める

ZFSとUFS - NFS/Samba/iSCSI

- UFSの場合
  - NFS　　: /etc/exportsの設定
  - Samba : samba入れる
  - iSCSI : iscsi-targetまたはistgt入れる

- ZFSの場合
  - 全部 : ZFSプロパティで設定

ZFSとUFS - NFS/Samba/iSCSI

- UFSの場合
  - N
  - iS

(FreeBSDの場合は)
全部じゃない

- ZFSの場合
  - 全部 ： ZFSプロパティで設定

ZFSとUFS - NFS/Samba/iSCSI

- SolarisならZFSプロパティ使えば簡単なんだけど...

- FreeBSDは、propertyあるけど使えない
  - zfs set shareiscsi

```
# zfs set shareiscsi=on zfsday-01
property 'shareiscsi' not supported on FreeBSD:
permission denied
```

    - Solaris系もSTMFに統合されているので、状況はほぼ同じ
  - zfs set sharesmb

```
# zfs set sharesmb=on zfsday-01
property 'sharesmb' not supported on FreeBSD:
permission denied
```

ZFSとUFS - NFS/Samba/iSCSI

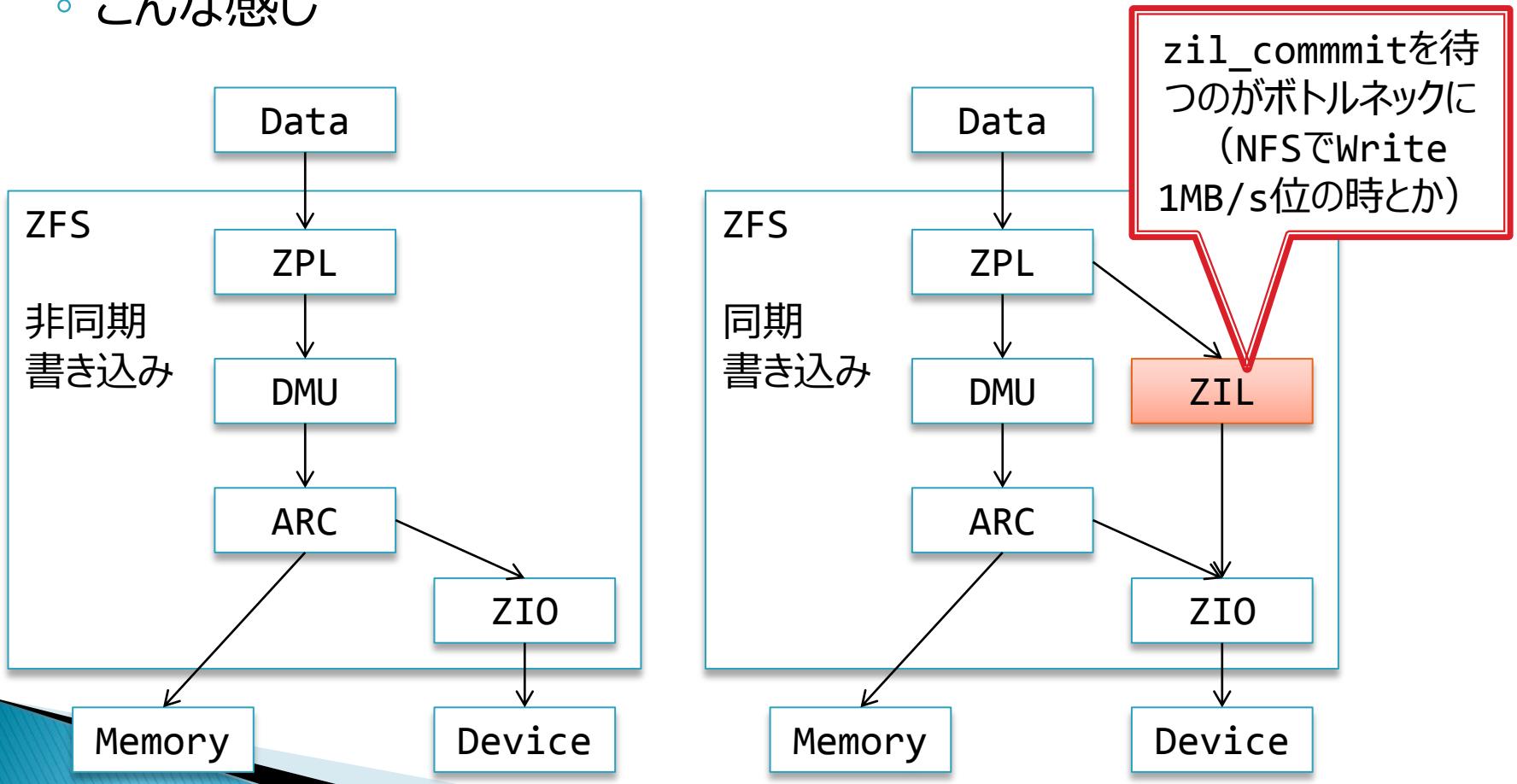
- 使えるのはNFSだけ
  - zfs set sharenfs

```
# zfs set sharenfs='-network 192.168.1 -mask 255.255.255.0' zfsday-01
# showmount -e
Exports list on localhost:
/zfsday-01                 192.168.1.0
```

- rpcbind/nfsd/mountdはもちろん必要
- 書式はFreeBSDのexportsと一緒
  - そこまでするならexports編集した方がいいんじゃないかな...
  - mountd restartしなくても有効になる

ZFSとUFS - NFS/Samba/iSCSI

- 性能がとっても低い時は、同期書き込みになっているかどうか意識した方がいいかもね
  - こんな感じ



ZFSとUFS - NFS/Samba/iSCSI

- ZILがボトルネックとなっていることが明らかな場合
  - ZILの強化
    - ZILとしてランダムI/Oに強いSSDやioDriveを導入する
  - ZILの無効化
    - `zfs set sync=disable pool/partition`
    - `sysctl -w vfs.zfs.zil_disable=1`
  - 上層で同期書き込みの使用をやめる(zvolには意味ない)
    - NFSなら`-o async`
    - Sambaなら`sync always = no`

ZFSとUFS - NFS/Samba/iSCSI

- 結局...

- UFSの場合
  - NFS　: /etc/exportsの設定
  - Samba : samba入れる
  - iSCSI : iscsi-targetまたはistgt入れる

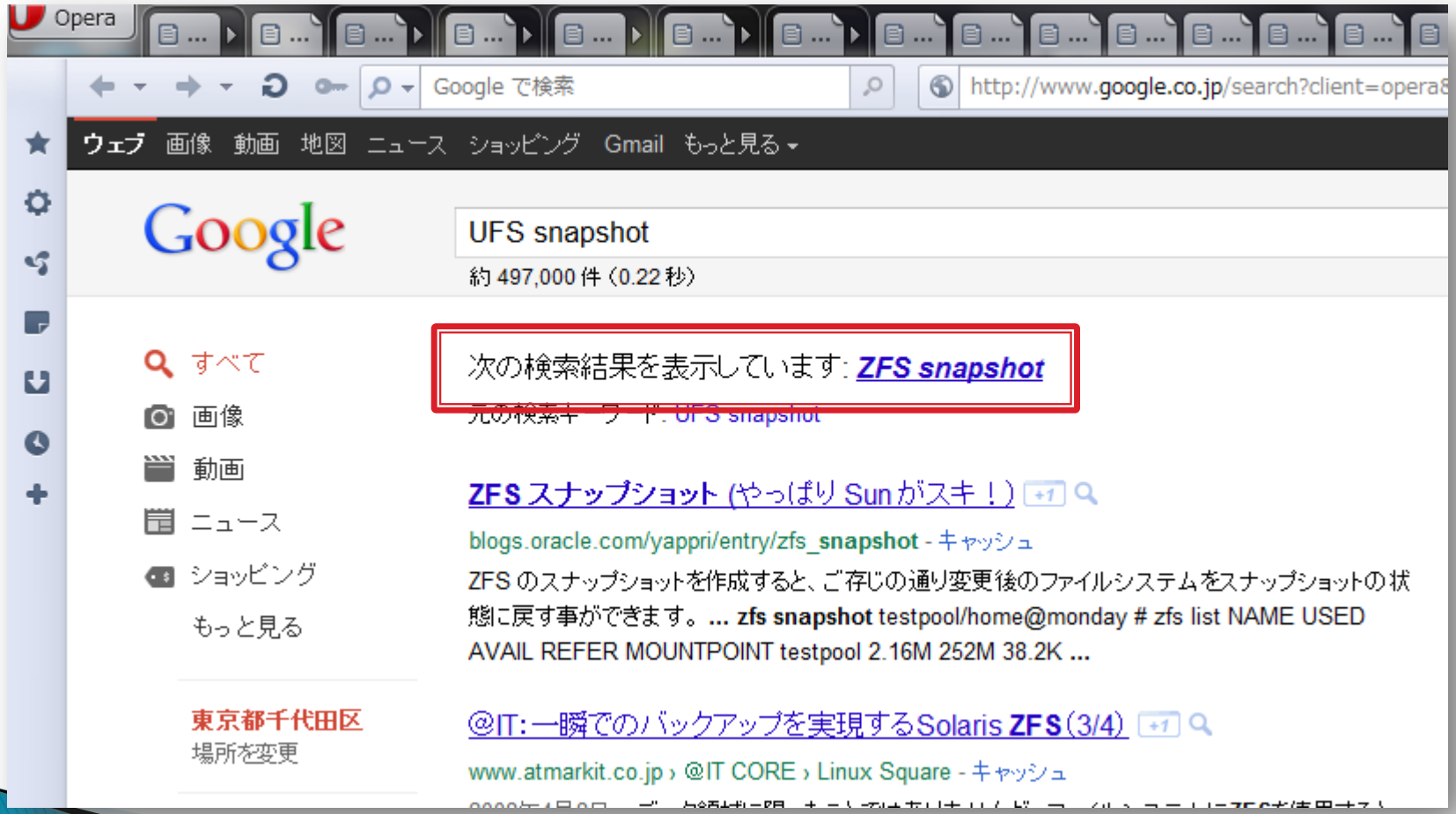
- ZFSの場合
  - NFS　: ZFSのプロパティで設定
  - Samba : samba入れる
  - iSCSI : iscsi-targetとかistgt入れる

ZFSとUFS - snapshot

- 間違えてファイルを消した時のために、snapshotを取る

ZFSとUFS - snapshot

さて、UFSのsnapshotでも...ん？



ZFSとUFS - snapshot

- Google「"UFS snapshot"...？」
  mksnap_ffs「おい」

- 一応ユーティリティもある
  - UFS/ZFS snapshot management utility

- ~~でも、みんな定期的なrsyncの方が好きだよね~~

ZFSとUFS - snapshot

- 確かにUFS(FFS)にもsnapshotはある...が、

	UFS(FFS) snapshot	ZFS snapshot
最大取得数	20個/FS	2^64個/プール
中身確認	mdconfig & mount	cd .zfs/snapshot/snap
ロールバック	不可	可
クローン	不可	可
速度低下	容量に伴い増加	ほとんど無い

- snapshotに関してはZFSの方が上のように感じる。
  - rollbackとかcloneを挙げてるのは少し作為的だけど...

ZFSとUFS - Backup

- データ壊して悲しい想いをしたくないから、Backupする

ZFSとUFS - Backup

- UFSのバックアップは主にdump/restore
  - zfsもdump/restoreできる。以下両方可能
    - UFS(dump) -> ZFS(restore)
    - ZFS(dump) -> UFS(restore)
- ZFSのバックアップは主にsend/recv
  - UFSはできないので、ZFS同士で使う

方式	UFS	ZFS
dump	○	○
restore	○	○
send	×	○
recv	×	○
その他ファイル単位バックアップ	○	○

ZFSとUFS - RAID

- HDDが壊れて悲しい想いをしたくないから、RAIDする

ZFSとUFS - RAID

- UFS...RAIDは？

- UFS自身にRAID機能は無い

- おしまい -

ZFSとUFS - RAID

- GEOMの機能を使ってRAIDできるよ？

	UFS(GEOM)	ZFS
JBOD	○	×
RAID0	○ （3台以上も可）	○ （3台以上も可）
RAID1	○ （3台以上も可）	○ （3台以上も可）
RAID10	○	○
RAID3	○	×
RAID5	△	RAIDZ
RAID50	△	RAIDZ + Stripe
RAID6	-	RAIDZ2
RAIDZ3	-	RAIDZ3
同期	ブロック全部	使用ブロックのみ

ZFSとUFS - その他機能

- その他機能もさらっと

	UFS	ZFS
圧縮	-	lzjb gzip[1-9] zle
データ複製配置	スーパーブロックのみ	メタデータ ユーザデータ
重複排除	-	ハッシュ一致 データ完全一致
自動修復	-	有
ブロックCheckSUM	-	有

ZFSとUFS - 性能

- うそっ・・・ZFSの性能低すぎ・・・？ （※）

- うん、「また」なんだ。済まない。
  - FreeBSDにおけるZFSのあれこれ–FFS/ZFSのベンチマーク
    http://gihyo.jp/event/01/freebsd/2010/0716

（※）体感速度は人と用途によって異なります

ZFSとUFS - まとめ

- ZFSはコマンド少なくて楽。けど、透明さには欠けるかも
- NAS/SANのやることは変わらない。でも性能に落とし穴
- snapshotはZFSの方が優秀。速いし簡単に増やせる
- バックアップは少し楽。復旧事例が少ないのが怖い
- RAIDは多機能上位互換。使いたいだけ使おう

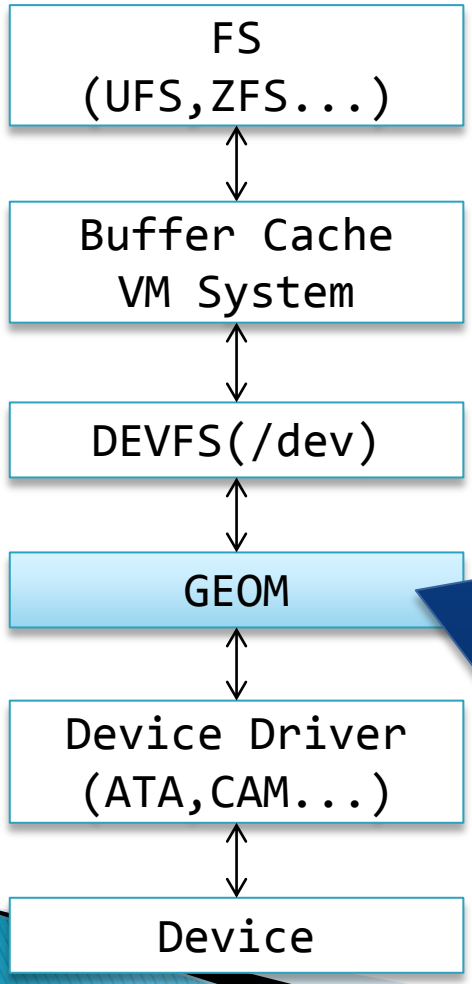
ZFSとGEOM

ZFSと...GEOM？

- ZFSはいいけど、GEOM？
- GEOM...modular disk I/O request transformation frameworkの（ｒｙ
- 簡単に言うと、デバイスとFSの中間層

GEOMの立ち位置

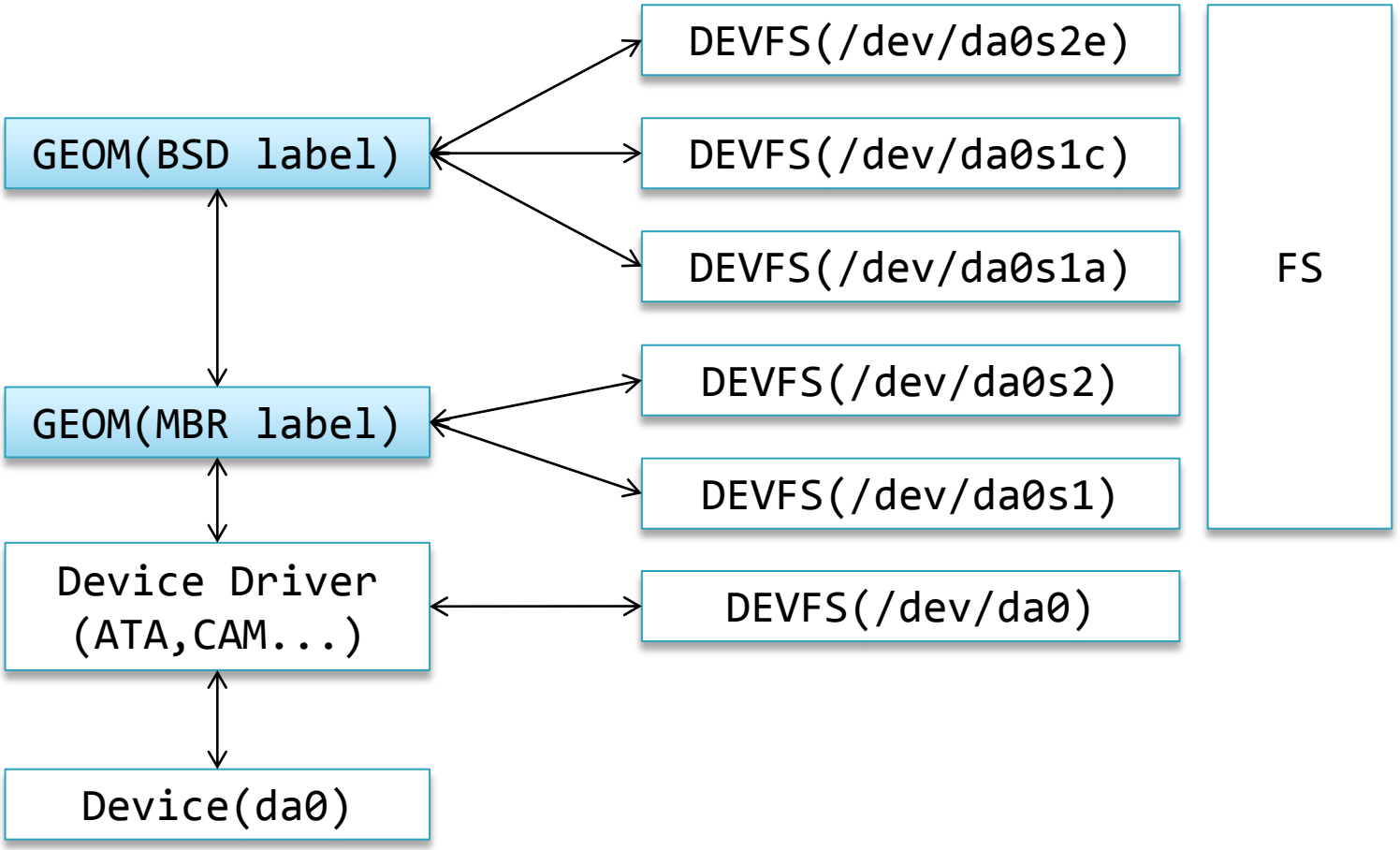
- 暗号化やRAID、FS拡張、統計情報などの機能を提供




```
geom_mirror : RAID1
geom_stripe : RAID0
...
geom_eli : Disk Encryption
...
geom_journal : UFSにjournal機能を追加
geom_cache : バッファキャッシュとしてRAMを使用
...
geom_nop : エラーテストやデバッグ用途
geom_gate : Netowork越しにデバイスを見せる
...
```

GEOMの仕事

- /devにスライス切ったデバイス名が見えるのも

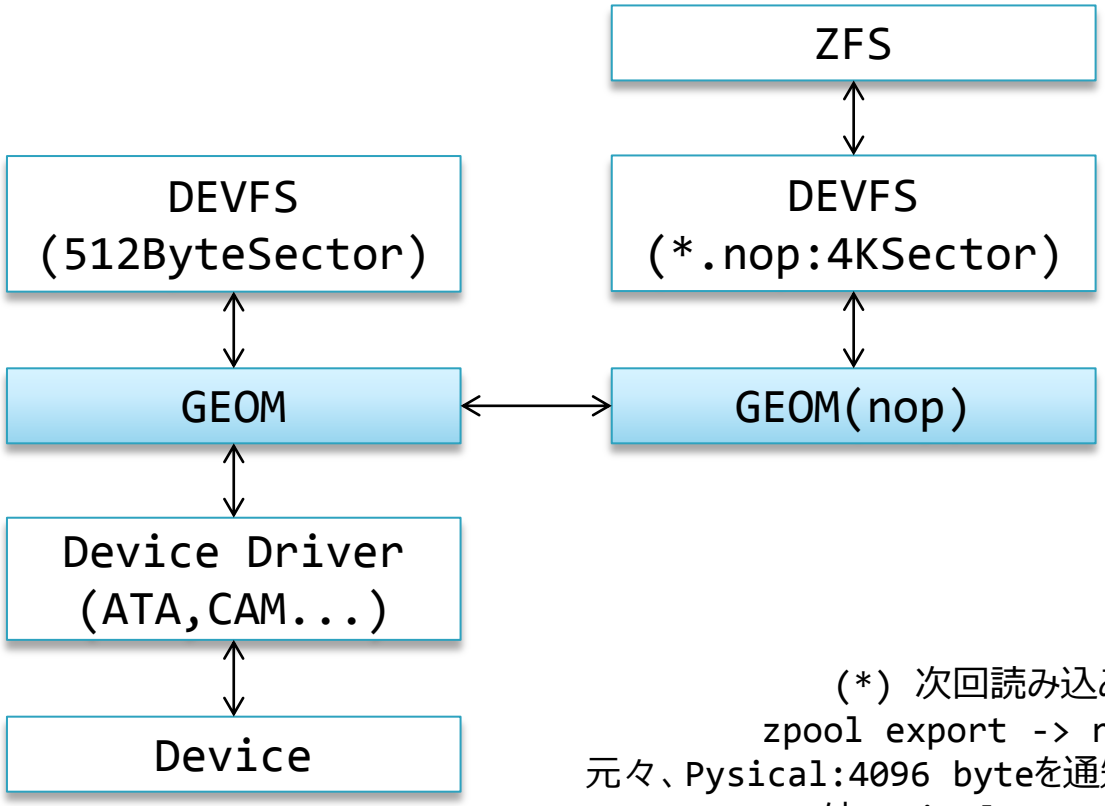


ZFSとGEOM

- ZFSと何の関係が？
- 多分、主な関心は以下の2つ
  - 4KiBセクタHDD(Pseudo-512Byte)対応
  - 暗号化

ZFSとGEOM - geom_nop

- geom_nopに4KiBセクタ通知をさせて対応
  - zpool create後、ashift=12になったら*.nop不要(*)



(*) 次回読み込みのために、以下の作業を推奨
zpool export -> nop削除 -> zpool import
元々、Pysical:4096 byteを通知するHDDには無用な作業で、
DEV_BSIZEはPysical sector sizeで上書きされるはず。
→ Physical sector sizeが4KiB通知しても、
zpoolはashift=9になりました。

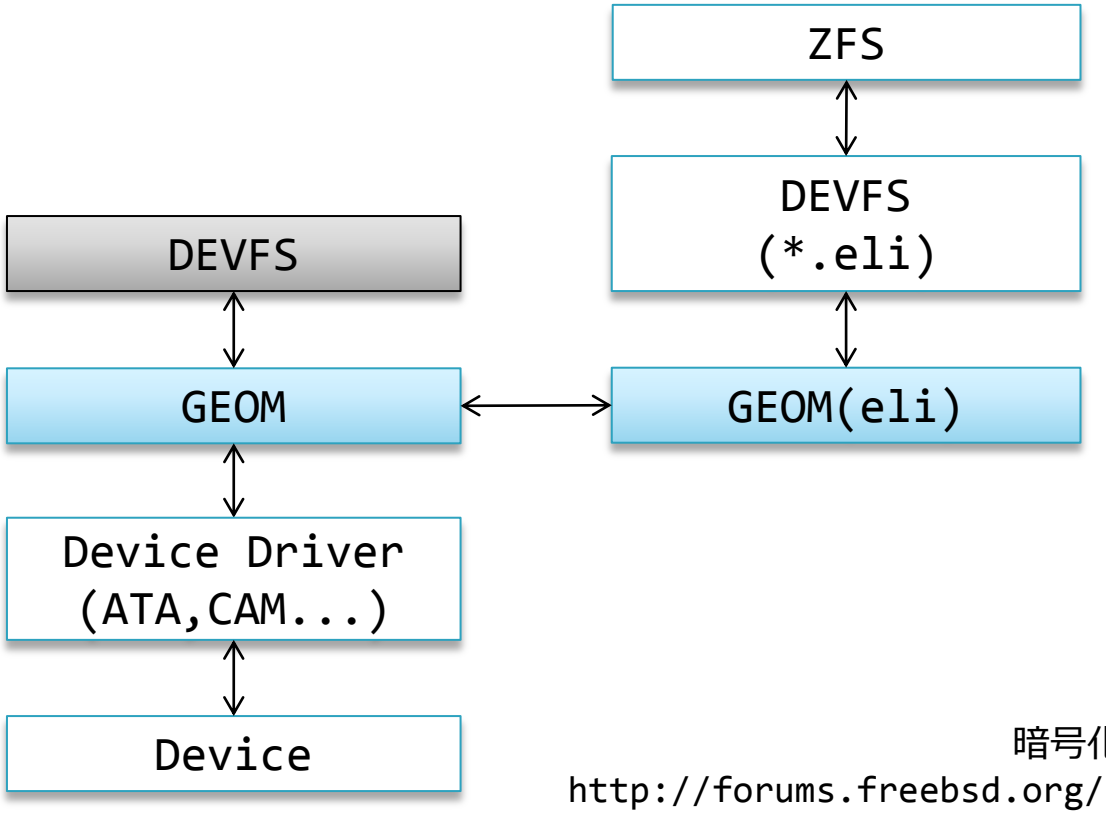
ZFSとGEOM

- お試しコマンド

```
# zpool create tank-4KiB da1
# zdb -C tank-4KiB | grep shift
                metaslab_shift: 23
                ashift: 9
# gnop create -S 4096 da1
# zpool create tank-4KiB da1.nop
# zdb -C tank-4KiB | grep shift
                metaslab_shift: 23
                ashift: 12
# zpool export tank-4KiB
# gnop destroy da1.nop
# zpool import tank-4KiB
# zdb -C tank-4KiB | grep shift
                metaslab_shift: 23
                ashift: 12
```

ZFSとGEOM - geom_eli

- geom_eliを挟んで暗号化
  - zpool v30が無いFreeBSDの暗号化方式



暗号化ディスクへの定期バックアップ例
http://forums.freebsd.org/showthread.php?t=18326
gihyoでも記事にされてるよ
http://gihyo.jp/admin/clip/01/fdt/201010/26

ZFSとGEOM - geom_eli

- お試しコマンド

```
# geli init -s 4096 da2
Enter new passphrase:
Reenter new passphrase:

Metadata backup can be found in /var/backups/da2.eli
and can be restored with the following command:

        # geli restore /var/backups/da2.eli da2

# geli attach da2
Enter passphrase:
# zpool create eli-tank da2.eli
```

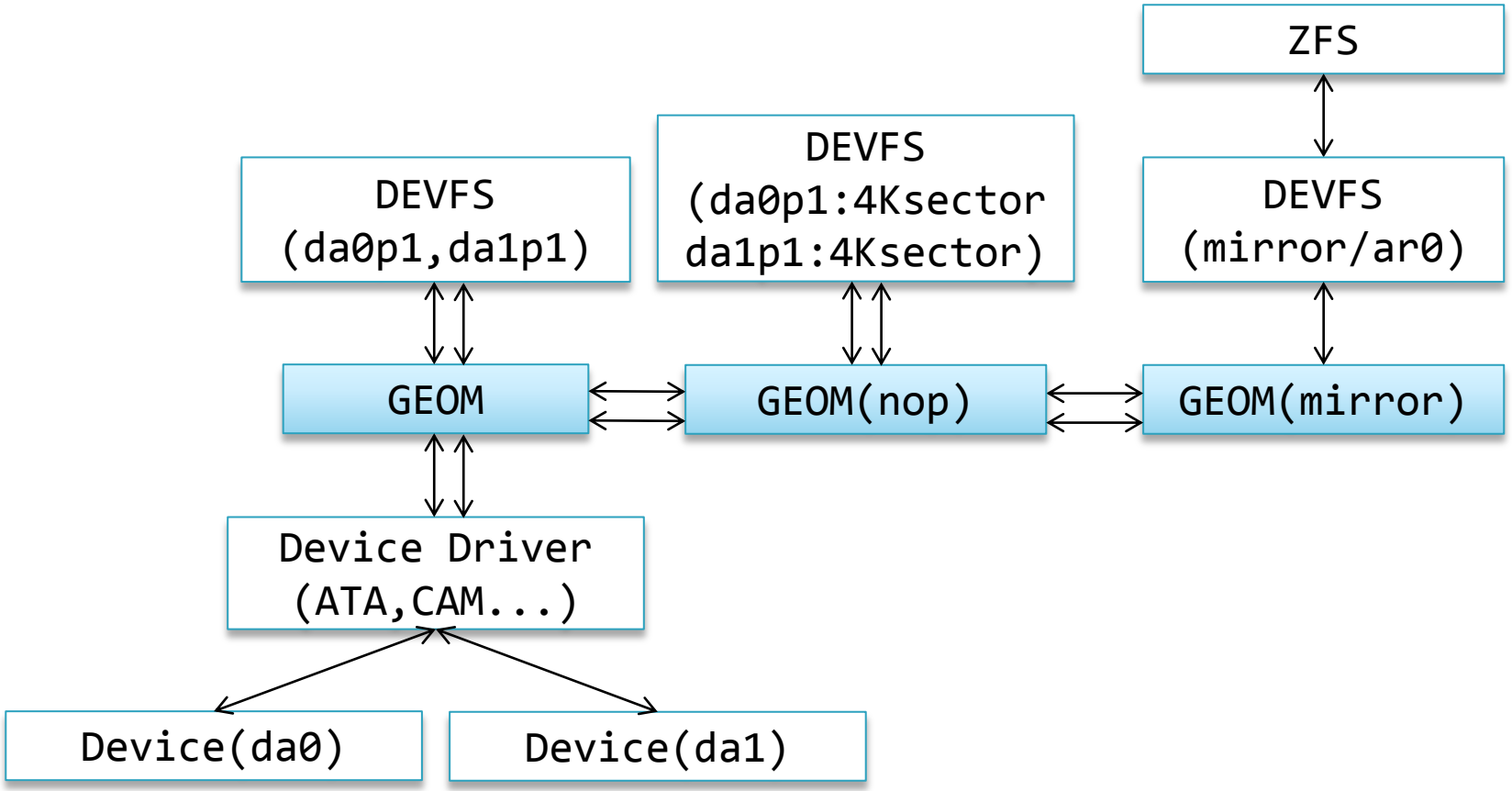
ZFSとGEOM

- そもそも、FreeBSDにおけるZFSの実装は、一部GEOMとして実装されている（直接のコマンドは無いみたい）
  - VDEV : geom_vdev
  - ZVOL : geom_zvol

- そして、GEOMはループしないなら組み合わせが可能
  - なので、スライス切った後でも、ZFSを上に作れる
    geom label -> geom vdevの順に組み合わさる
    GPTならgeom part -> geom vdev

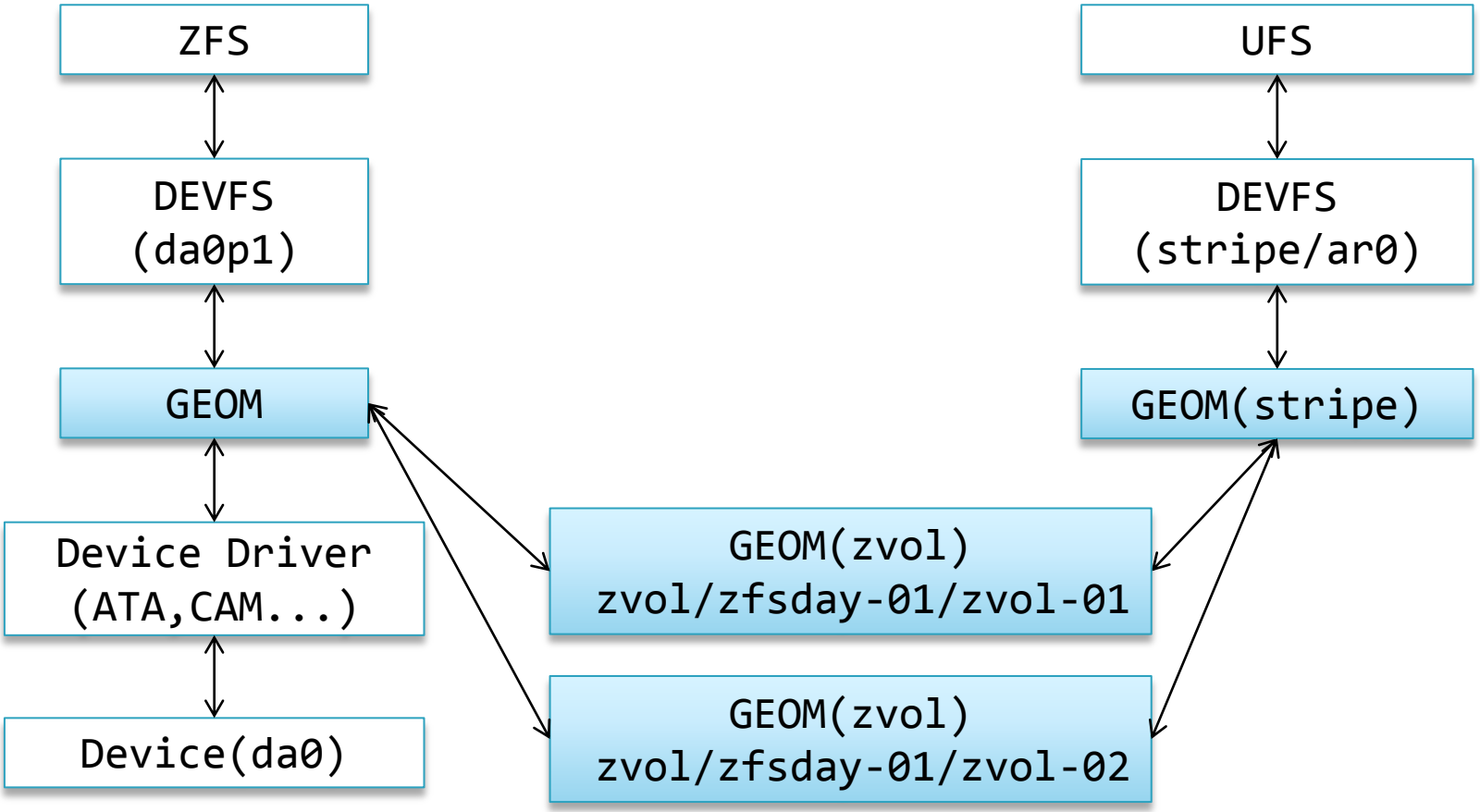
ZFSとGEOM

- 例えば、2-HDD on 4KiB通知 on RAID1 on ZFS



ZFSとGEOM

- 例えば、1-HDD on 2-ZVOLs on RAID0 on UFS



UFSの代わりにzpool createしたらKernel Panicしたけどね...

ZFSとGEOM

- GEOMは機能満載
- ZFSと組み合わせて使えるよ

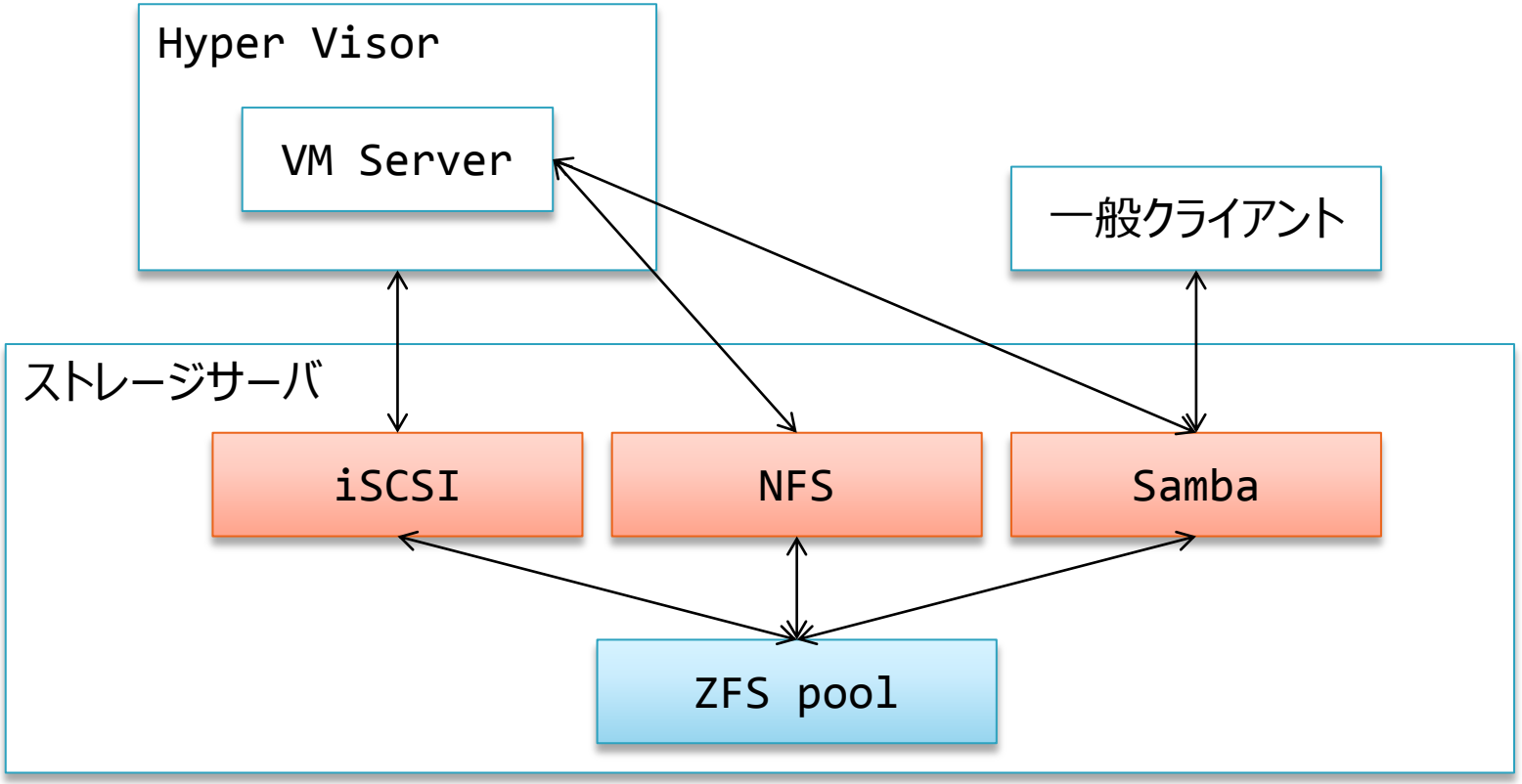
ちょっとだけ運用話

ちょっとだけ運用話

- ご紹介
- 使い方
- 悩み事

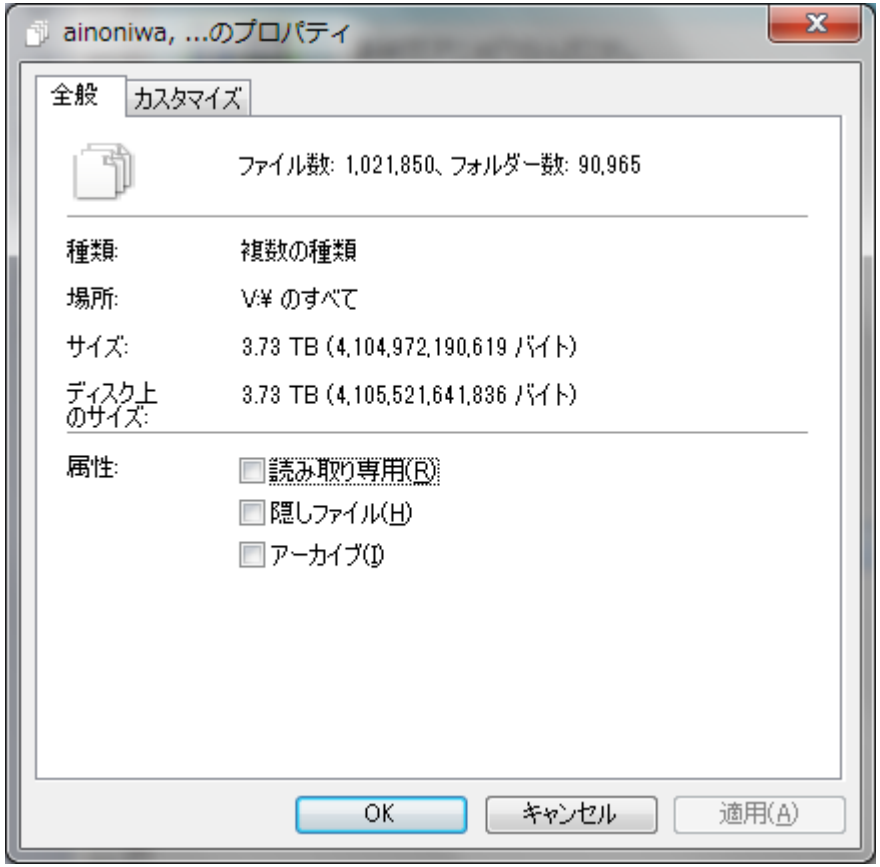
ちょっとだけ運用話 - 用途

- 簡単構成図



ちょっとだけ運用話 - 蓄積規模

- ファイルサイズ合計 ： 3.7TB強
- ファイル数 ： 100万個位、 フォルダ数 ： 9万個位



ちょっとだけ運用話 - 材料

部位	機材	備考
ケース	Antec NineHundred	
電源	EarthWatts EA-650	
M/B	GA-MA780G-UD3H	
CPU	Athron X4 605e	
メモリ	Pulsar DCDDR2-4GB-800 W2U800CQ-2GL5J	
HDD	HTS545016B9A300 *2(system) WD10EADS *5 HDS722020ALA330 *2 WD15EADS HTS545016B9A300	
SASカード	DELL SAS 6i/R	
NIC	EXPI9301CT	OnboardとLAG
エンクロージャ	DIR-2221-SATA CSE-M35T-1B CRS-3056SS	2.5inch *2 3.5inch *5 3.5inch *5

ちょっとだけ運用話 - 年表

時期	出来事
2009-06-29	データプールが組まれる （この時、システムはUFSだった）
2009-11-28	データプールのHDD(ad4)の交換
2010-03-11	データプールのHDD(ad12)の交換
2010-06-28	システム(UFS)が壊れた。 システム再インストール後、データプールをインポート
2011-03-13	データプールのHDD(ad8)の交換
2011-10-15	今に至る

ちょっとだけ運用話 - 使い方

- 仮想マシン用のLU増やすときは、zvolではなくFile
  - 仮想ホスト用のZFSパーティション切って、その中にファイル
  - 仮想ホストが停電とかネットワーク断とか余計なアップデートとかオペミスで壊れたら、対象パーティションをrollback

```
# zfs create onechan/iSCSI/youmu
# vi /usr/local/etc/istgt/istgt.conf
[LogicalUnit9]
  Comment "For Observer"
  TargetName youmu-01
  TargetAlias "Youmu Disk1"
  Mapping PortalGroup1 InitiatorGroup1
  AuthMethod Auto
  AuthGroup AuthGroup1
  UseDigest Auto
  UnitType Disk
  QueueDepth 32
  LUN0 Storage /onechan/iSCSI/youmu/youmu-01 30GB
# zfs rollback onechan/iSCSI/youmu@20111013-17
```

ちょっとだけ運用話 - 使い方

- 仮想マシン用にファイルを使うと、サイズ変更が簡単
  - istgtの設定で、LUサイズを変更してサービス再起動すると、iSCSI Initiator側でセッション再確立した時にサイズ増えてる
  - 気になった時、vi -bして%!xxdできるかもしれないので
  - zvol使うとZILを強制されるので、zil_disable=1するよりはファイルベースの非同期書き込みにしちゃおうかな、と。（v28ならsync=disableで使うかも）
    - SSD欲しい。
- 性能はちょっと低いかもしれない（ちゃんと覚えてない

ちょっとだけ運用話 - 使い方

- 物理ディスクの入れ替え作業
  - ATAのとき

```
# atacontrol list
# atacontrol detach ata2
# atacontrol attach ata2
# zpool replace ad4
```

  - CAM(SCSI)のとき

```
# camcontrol devlist
# camcontrol eject 0:1:0
# camcontrol rescan all
# zpool replace da1
```

ちょっとだけ運用話 - 使い方

- 定期スナップショット
  - cronでsnapshotを取るように記載

```
# cat zfs_snapshot.sh
#! /bin/sh
/sbin/zfs snapshot zfsday-01@`/bin/date '+%Y%m%d-%H'`
/sbin/zfs destroy zfsday-01@`/bin/date -v "-6d" '+%Y%m%d-%H'`
/bin/ln -s /mnt/zfsday-01/.zfs/snapshot/`/bin/date -v "-6d"
'+%Y%m%d-%H'` /mnt/zfsday-01/backup/`/bin/date -v "-6d"
'+%Y%m%d-%H'`

# tail -2 /etc/crontab
### zfs snapshot
5  17  *  *  *  root  /root/zfs_snapshot.sh
```

  - パーティションが増えたら、大体同じように増やす。
  - お金も空間も余裕無いので、バックアップ用の別ディスクはない。

ちょっとだけ運用話 - 使い方

- 削除ポリシーは人それぞれなので、適当に決めます
  - うちではこうなる

```
# zfs list -t snapshot | head -8
NAME                      USED  AVAIL  REFER  MOUNTPOINT
komakan@20110826-17          0      -    19K  -
komakan@20110827-17          0      -    19K  -
komakan@20110828-17          0      -    19K  -
komakan@20110829-17          0      -    19K  -
komakan@20110830-17          0      -    19K  -
komakan@20110831-17          0      -    19K  -
```

ちょっとだけ運用話 - 使い方

- ついでに、.zfs/snapshot/*にシンボリックリンク張っておくと、共有フォルダの操作中に間違って消した後、すぐに復帰できる。

```
# cat zfs_snapshot.sh
#! /bin/sh
/sbin/zfs snapshot zfsday-01@`/bin/date '+%Y%m%d-%H'`
/sbin/zfs destroy zfsday-01@`/bin/date -v "-6d" '+%Y%m%d-%H'`
/bin/ln -s /mnt/zfsday-01/.zfs/snapshot/`/bin/date -v "-6d"
'+%Y%m%d-%H'` /mnt/zfsday-01/backup/`/bin/date -v "-6d"
'+%Y%m%d-%H'`

# tail -2 /etc/crontab
### zfs snapshot
5  17  *  *  *  root  /root/zfs_snapshot.sh
```

```
# ls -l
total 102433
lrwxr-xr-x   1 root  wheel   23 Oct 11 23:57 20110826-17 -
> .zfs/snapshot/20110826-17
```

ちょっとだけ運用話 - 使い方

- snapshotを取得すると（というよりも、zfsの操作全般は）ログに残るので、後から追える
  これは、今稼働中のシステムディスクのもの

```
~# zpool history | head -100 | tail -4
2011-04-13.17:05:01 zfs snapshot komakan@20110413-17
2011-04-13.17:05:01 zfs destroy komakan@20110407-17
2011-04-14.17:05:01 zfs snapshot komakan@20110414-17
2011-04-14.17:05:02 zfs destroy komakan@20110408-17
```

- たまにスクリプト間違えてる時があるので、軌道に乗るまではログを見たりする
- create時のログとか、まだ残ってる

```
# zpool history | head -5
History for 'komakan':
2010-06-28.23:05:42 zpool create komakan /dev/ad4p2
2010-06-28.23:05:56 zfs create komakan/system
2010-06-28.23:05:59 zfs create komakan/system/usr
2010-06-28.23:06:01 zfs create komakan/system/var
```

ちょっとだけ運用話 - 使い方

- あんまり使わないけど、オプションもある。でもマジで使わない
- ユーザ/ホスト名を見れるオプション(-l)

```
# zpool history -l | head -100 | tail -4
2011-04-15.17:05:01 zfs snapshot komakan@20110415-17
[user root on remilia.ainoniwa.net:global]
2011-04-15.17:05:01 zfs destroy komakan@20110409-17
[user root on remilia.ainoniwa.net:global]
2011-04-16.17:05:04 zfs snapshot komakan@20110416-17
[user root on remilia.ainoniwa.net:global]
2011-04-16.17:05:05 zfs destroy komakan@20110410-17
[user root on remilia.ainoniwa.net:global]
```

- 内部イベントを見るオプション(-i)

```
root@remilia ~# zpool history -i | head -100 | tail -4
2011-03-19.17:05:01 [internal snapshot txg:3291862] dataset = 79
2011-03-19.17:05:01 zfs snapshot komakan@20110319-17
2011-03-19.17:05:01 [internal destroy txg:3291864] dataset = 102
2011-03-19.17:05:01 zfs destroy komakan@20110313-17
```

ちょっとだけ運用話 - 使い方

- Historyログは、ディスク上に保存される
  - 最大サイズは32MiB
  - 最小サイズは128KiB
  - デフォルトはプールサイズの1%
- 32MiBあるので、512Byte/Logと仮定すると64,000ログ保存できる。リングバッファなので、古いログから消失していく。
  - サイズの話はこのへん

```
# cd /usr/src/sys/cddl/contrib/opensolaris/uts/common/fs/zfs
# cat -n spa_history.c | head -110 | tail -7
104          /*
105           * Figure out maximum size of history log.  We set it at
106           * 1% of pool size, with a max of 32MB and min of 128KB.
107           */
108          shpp->sh_phys_max_off = spa_get_dspace(spa) / 100;
109          shpp->sh_phys_max_off = MIN(shpp->sh_phys_max_off, 32<<20);
110          shpp->sh_phys_max_off = MAX(shpp->sh_phys_max_off, 128<<10);
```

ちょっとだけ運用話 - 使い方

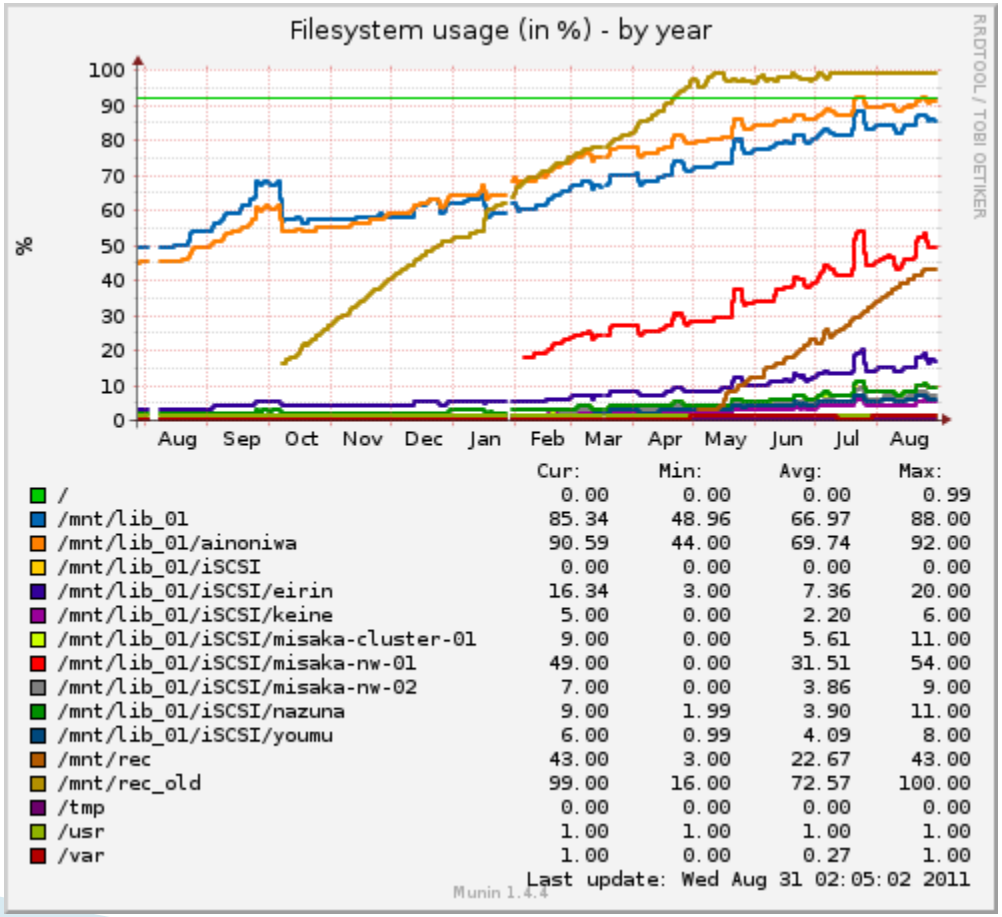
- そこそこファイル数があっても、元気に動いてます
- 使い方はNFS/Samba/iSCSIくらい
  - LDAPとかも動いてるけど、それは無視
- snapshotはマジで便利
- ホットスワップが使えるので、故障交換を前提に運用

ちょっとだけ運用話 - 監視

- 監視
  - と言っても大したことはしてない
  - グラフ書いてるだけ
  - HDDの温度、ディスク消費量位
  - ARCの監視をしようかなぁ、とは思ってても動いてない
  - ZILは分離していないからSSD監視項目とかは無い

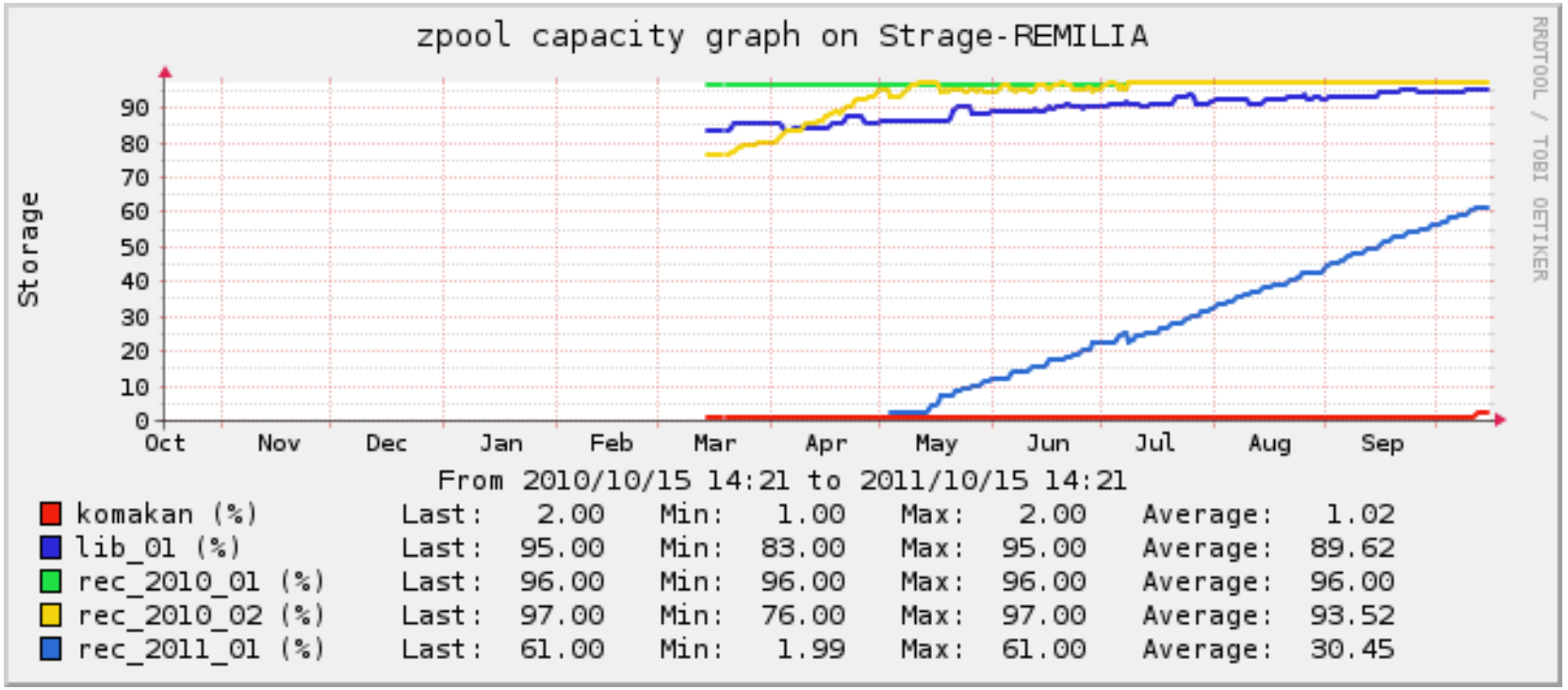
ちょっとだけ運用話 - 監視

- 容量はdfベースで見ると、どのパーティションが圧迫してるのか分かりにくいし、残量も見えにくい



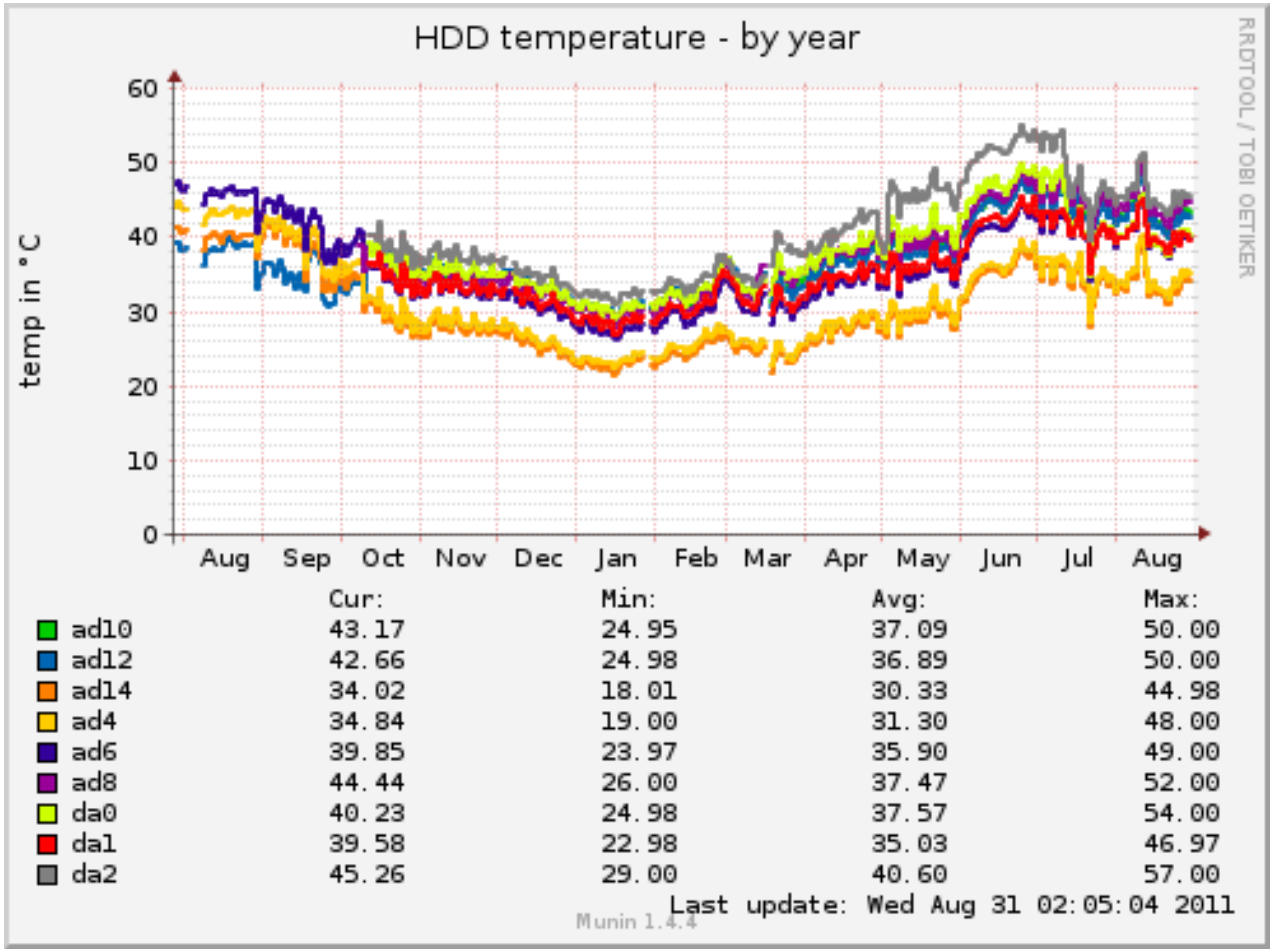
ちょっとだけ運用話 - 監視

- zpool1ベースで見ると、HDDの残り領域が見やすい



ちょっとだけ運用話 - 監視

- ZFS関係無いけど、年間のHDD温度とか



ちょっとだけ運用話 - 悩み事

- 機材選定はFreeBSD関係なく課題？
  - SATA/SASポートが足りない
  - 丁度いいケースがない
  - HBAのドライバがない
  - M/BにPCI-Expressが余ってない(レーン数が合わない)
  - S.M.A.R.T取れない
  - 変なセクタ通知HDD選ぶとオペレーションが増える
  - RAID組まないとOSに見えないRAIDカードの存在
  - 活線抜去できないSATA(IDE)ポート
  - HDD抜いてRAIDカードがOFFLINEと判定するまで十数秒

  ...etc

ちょっとだけ運用話 - 悩み事

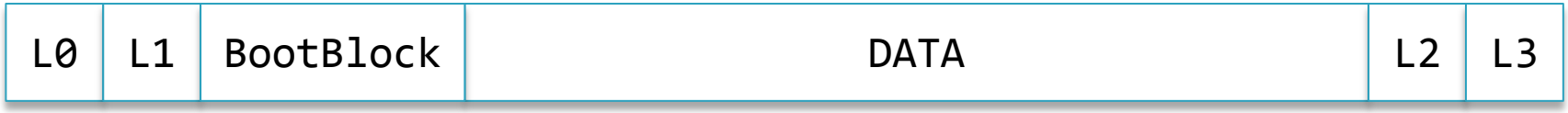
- チープなSATAで組めるZFSだけど
- チープな数のSATAポートじゃ満足できない

ちょっとだけ運用話 - 悩み事（RAIDカード）

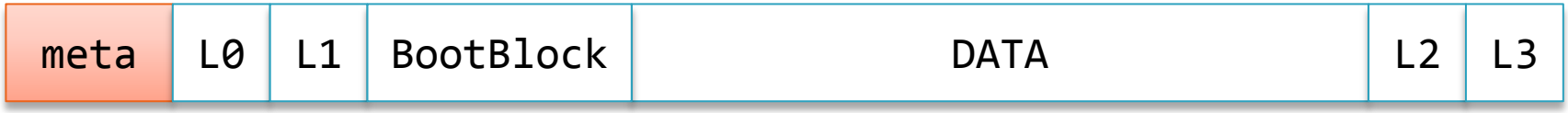
- 例えば。
- 当初、安くSATAポートが欲しくてHigh Point社のRocketRAID2300（PCI-E x1 4ch）を使ってた
- Initializeしないとデバイスが生えてこない。
  仕方ないのでHDD1台でJBODしてみた

ちょっとだけ運用話 - 悩み事（RAIDカード）

- オンボードSATA経由(ex.ad4)



- RAIDカード経由（ex.da1）



- 同じHDDでもRAIDカードでInitializeするとメタデータ分セクタ先頭がずれる。（末尾もずれるかも）
- その状態でzpool createしても、ラベル位置がずれるので繋ぎ変えた先でimportできない。

ちょっとだけ運用話 - 悩み事（RAIDカード）

- 今回の件に限って言えば、以下の手順で回避可能。
  1. non-RAID firmware導入
  2. オンボードSATAに繋いでMBR/GPTでパーティションを作成
  3. RAIDカードに繋ぎ直す
  4. 20秒ほど待つ
  5. regacyとしてOSには直接ディスクが見える

- 正直、こんな運用を強制されるのは謹んでお断りしたい。
  - 他にも、ソース修正+カーネル再構築で回避する手段もあるけど、そっちも正直お断りしたい。

ちょっとだけ運用話 - 悩み事

- 性能
  - どうやって測定すればいい？何の意味がある？
    - bonnie++, raidtest, Iozone...
    - 乗ってる仮想マシン、コンテンツの体感速度測定？
  - 容量的には3年戦えるとして、3年後の性能は？
    - 下がることはあっても、上がることはほとんど無い
    - 一定領域を埋めてsnapshotで保持しておいて、性能低下時にsnapshot消して性能復帰？
  - I/O性能を役立てるサービスが無い :(

ちょっとだけ運用話

- これからも可愛いZFSを使い続けるのでしょう。

FreeBSDさんとZFSさん

ご清聴ありがとうございました