やってみたNative ZFS on Momonga Linux

2010-09-11

ver0.5

Momonga Project

Takaaki Tabuchi

概要

- ZFSとは
- Native ZFS on Linux
- Momonga Linux 7 での ZFS環境の構築方法

ZFSとは

- Solarisで作成されたファイルシステム

ZFSの利点

- 128 bit FileSystem
- オープンソース
- ストレージプールとしてデバイスを仮想化
- RAID-Z
- iSCSI/CIFS/NFSとの統合

Native ZFS on Linux from LLNL

- ZFS の Linux kernel module としての実装が LLNL(Lawrence Livermore National Laboratory /ローレンス・リバモア国立研究所)から発表された。

- Momonga Linux 7 ではこちらの実装が利用可能。

Native ZFS on Linux from LLNL

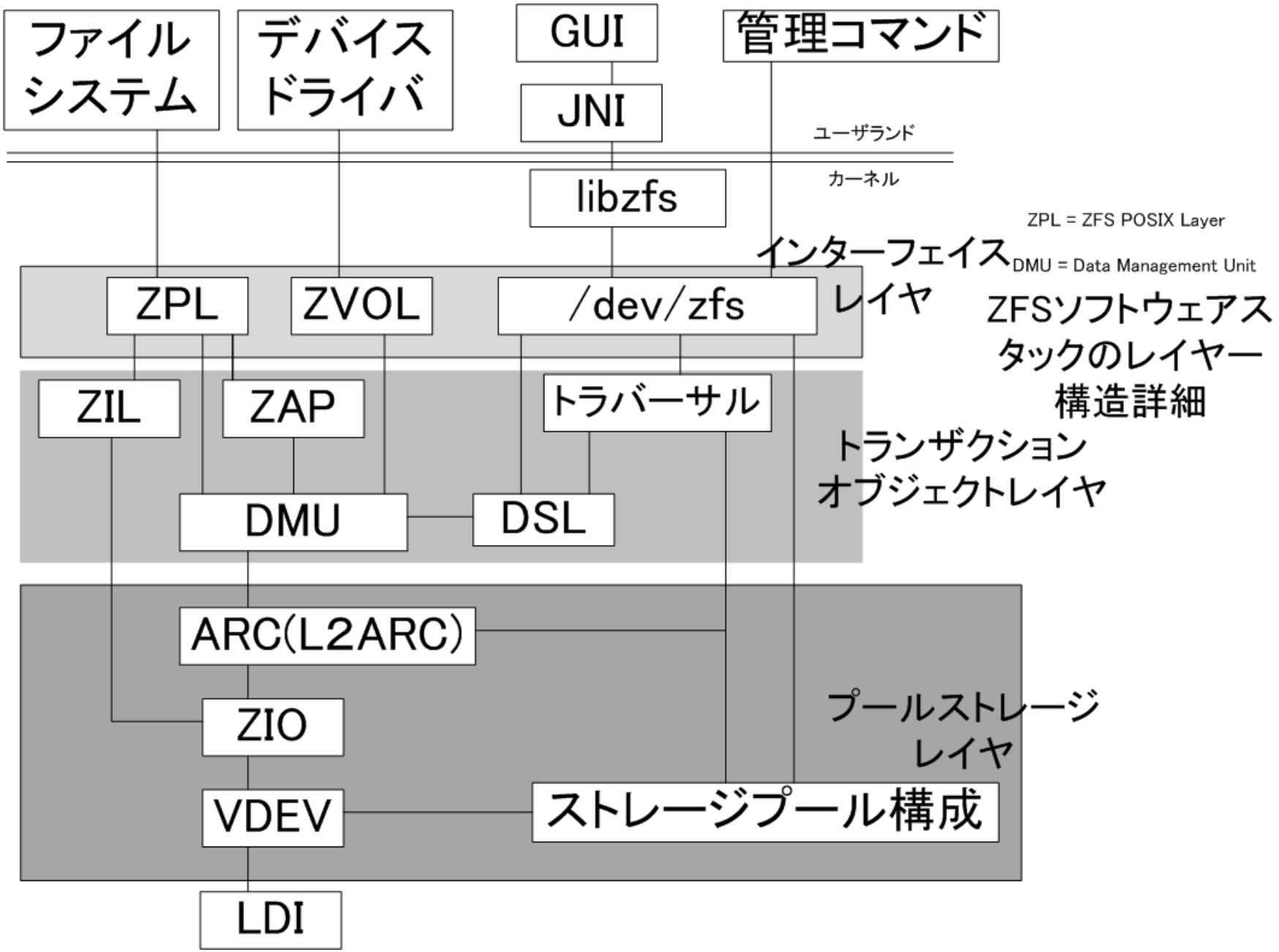
注意点: ZFS としてはmountできない。

理由はZPL(ZFS POSIX Layer)が未実装である為。

ZPLに関しては次ページの図を参照。

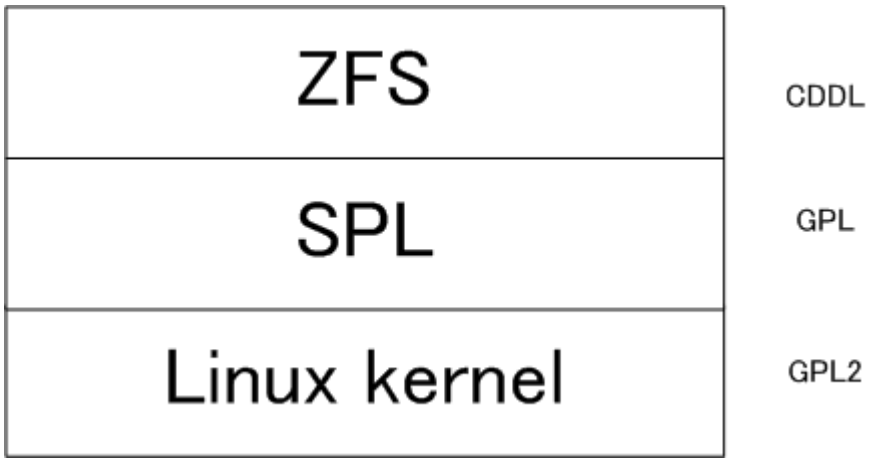
つまり、別途ext4などでformatしてmountする必要
　がある。

ZFSの構造(Solaris / Open Solaris)



Native ZFS on Linux の構造

- SPL(Solaris Portable Layer) を設けた実装となっている。
- SPLを設けることで、移植のためのZFSコードの変更作業を減らせ移植の手間を減らせる。また、ライセンス対策(?)の為、Linux カーネルとZFSモジュールを直接リンクさせない構造である。



CDDLとGNU GPL

- CDDLとは、Common Development and Distribution License のことで、ZFSのコードはこのライセンス下でリリースされている。
- CDDLはGNU GPLと矛盾するため、CDDLで保護されたモジュールはGPLのコードとは合法的に一緒にリンクすることができない。

（参考:さまざまなライセンスとそれらについての解説 - GNU プロジェクト http://www.gnu.org/licenses/license-list.ja.html）

実際にやってみる

前提

- Linux のインストールができる
- UNIXの知識はそれなりにある
- UNIX/Linuxのコマンドが使える
- # は root 権限のshell prompt
- $ は一般ユーザのshell prompt

実際にやってみる

手順

- Momonga Linux 7 をインストール
- zfs moduleパッケージをインストール
- (fdisk) + zpool + zfs + mkfs + mount

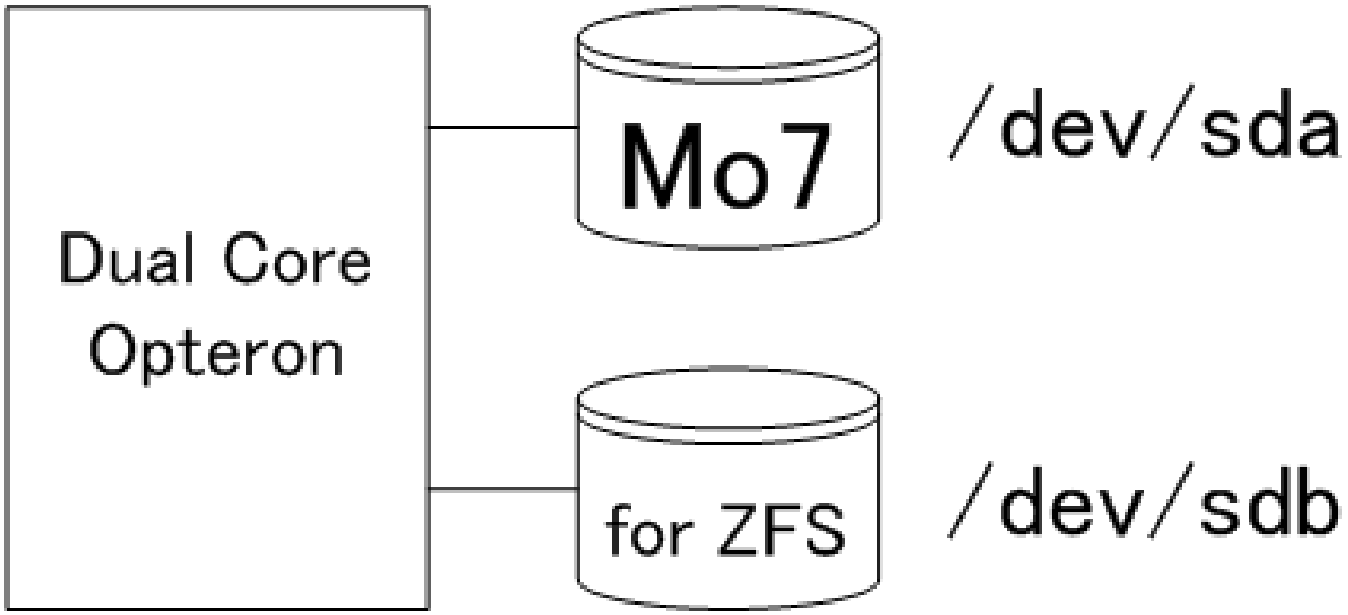
テストしたハードウェア構成

CPU: Dual-Core AMD Opteron(tm) Processor 1216

Memory: 2GB

HDD(1): ST3250823AS (250GB) : Mo7 OS用

HDD(2): ST3500320AS (500GB) : ZFS用



テストしたハードウェア構成

HDD1本での構成はテストしたが成功せず。

i686マシンでは試していない。

Momonga Linux 7 のインストール

手順は省略

テスト時はMomonga Linux 7 beta 3 を使用。

パッケージをインストール

kernel は最新のものを使うのが吉

yum install zfs-modules

念のため
yum install zfs
も実行しておく。

前者がZFSカーネルモジュール、
後者がzpool/zfs コマンドのインストールである。

Momonga Linux 7でのZFS関連パッケージの構成

1. splパッケージ：SPLコマンド(spl, splat) を提供
2. zfsパッケージ：ZFSコマンド(zfs, zpoolなど)を提供
3. zfs-modulesパッケージ: ZFSカーネルモジュールを提供

SPL モジュールは kernel パッケージに同梱

ZFSモジュールパッケージは、kernelのアップデートと同期する必要がある。

更新を忘れると、zfs モジュールを modprobe できない。(次ページ参照)

modprobe zfs

```
# modprobe zfs
#
```

確認方法は、lsmod | grep zfs

zfs モジュールをロードすると /dev/zfs が作成される。

fdisk

fdisk を用いて /dev/sdb の partitionを消す。
消さないと zpool コマンドがうまく動かない。

うまく動かない例:

```
# zpool create dpool /dev/sdb
cannot open '/dev/sdb': Device or resource busy
#
```

fdisk

うまく動かない理由: zpoolが自動的にGPTパーティションを作る為

```
# fdisk -l /dev/sdb

WARNING: GPT (GUID Partition Table) detected on '/dev/sdb'! The util fdisk
    doesn't support GPT. Use GNU Parted.


Disk /dev/sdb: 250.1 GB, 250059350016 bytes
256 heads, 63 sectors/track, 30282 cylinders, total 488397168 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x1a5d1a5c

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1               1   488397167   244198583+  ee  GPT
```

zpool + zfs + mkfs + mount

ちゃんと動く例:

```
# zpool create dpool /dev/sdb
# zfs create -V 400G dpool/fs
# mkfs.ext4 /dev/dpool/fs

# mkdir /zfs
# mount /dev/dpool/fs /zfs
#
```

df の表示例

```
$ df –HT
Filesystem    Type     Size   Used  Avail Use% Mounted on
/dev/mapper/vg_72-lv_root  ext4      53G   7.4G    43G  15% /
tmpfs         tmpfs     2.1G      0   2.1G   0% /dev/shm
/dev/sda1     ext4     508M    30M   452M   7% /boot
/dev/mapper/vg_72-lv_home   ext4     101G   199M    95G   1% /home
/dev/dpool/fs ext4     423G   208M   402G   1% /zfs
$
```

```
/dev/dpool/fs ext4     423G   208M   402G   1% /zfs
```

という、最下行が zpool で作成したパーティション の mount の内容。

ベンチマーク1

hdparmで試してみた。(3回の中間値)

```
# hdparm -Tt /dev/sdb ; hdparm -Tt /dev/dpool/fs

/dev/sdb:
 Timing cached reads:   2078 MB in  2.00 seconds = 1039.52 MB/sec
 Timing buffered disk reads: 202 MB in  3.01 seconds =  67.09 MB/sec

/dev/dpool/fs:
 Timing cached reads:   2104 MB in  2.00 seconds = 1052.68 MB/sec
 Timing buffered disk reads: 210 MB in  3.04 seconds =  69.19 MB/sec
#
```

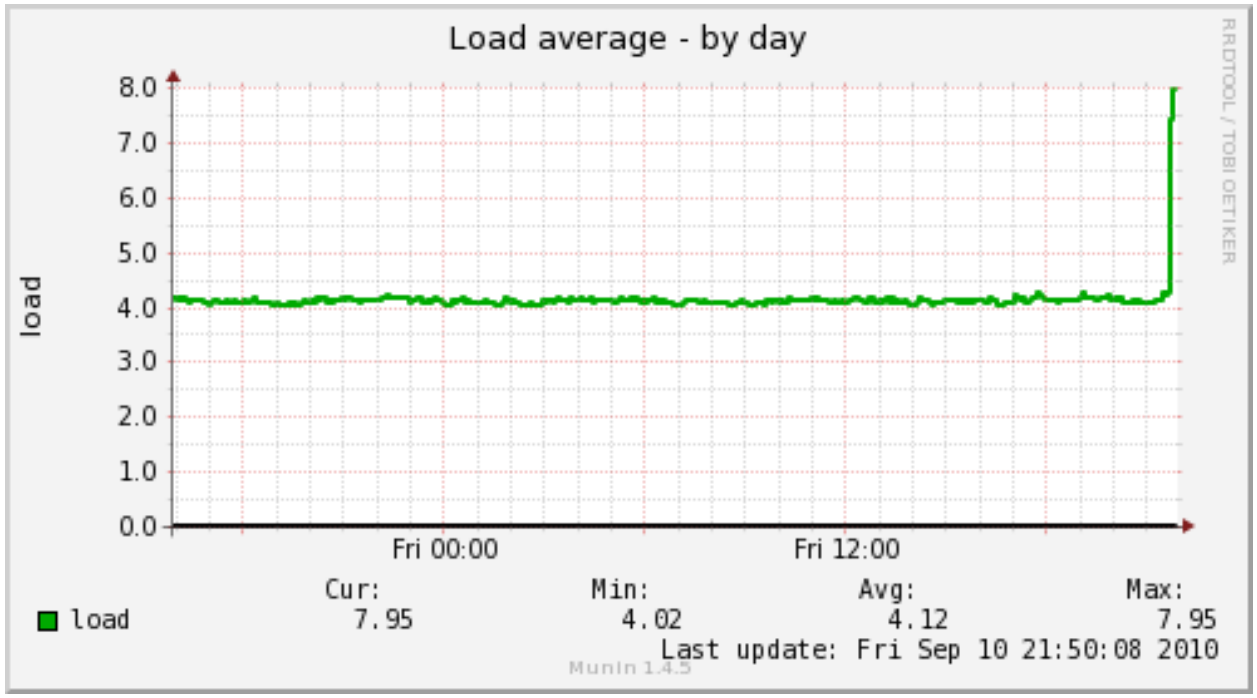
ベンチマーク2

bonnie++で試してみた。(3回の中間値)

cd /zfs ; bonnie++ -d `pwd`

結果は取得できず。

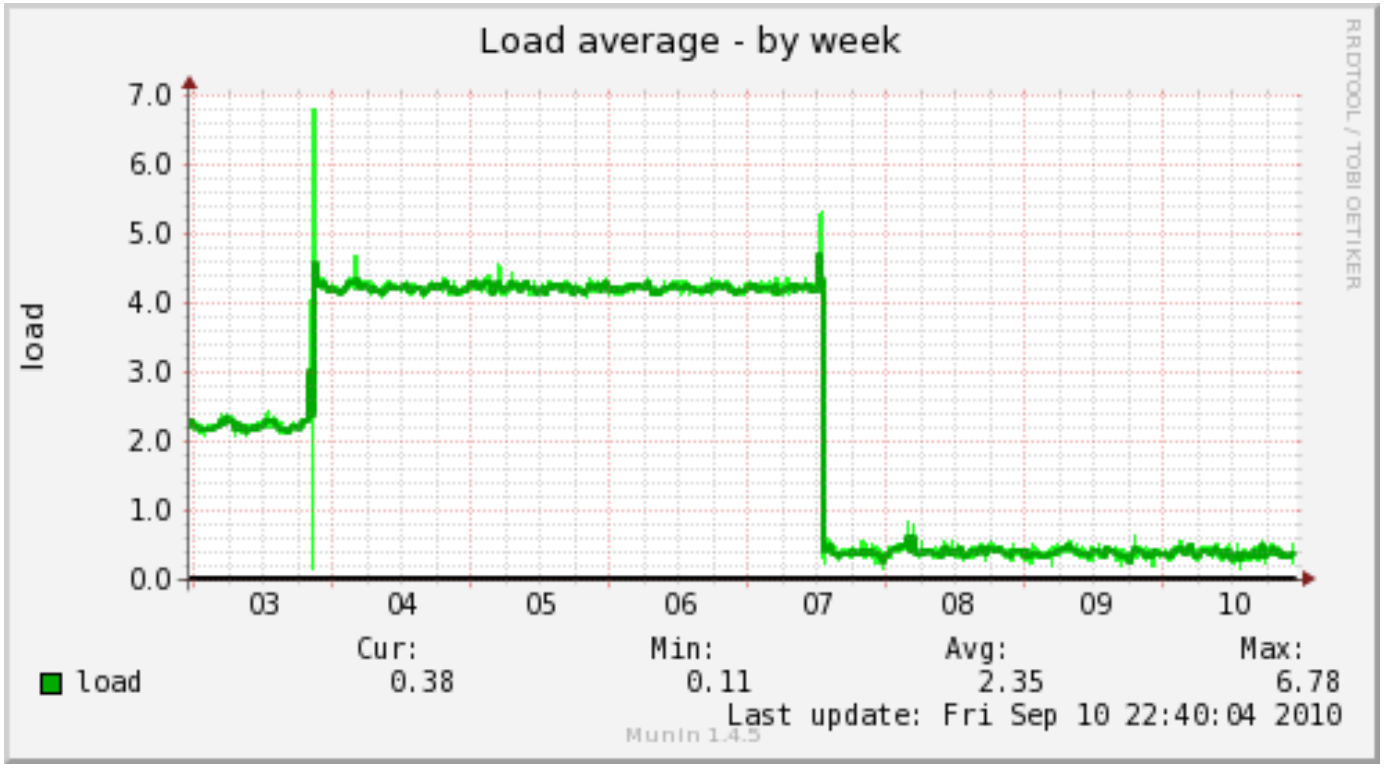
loadが定常時の 4から8を超えた所で、ssh/httpdが反応せず。これが断末魔のグラフ。



問題点1

load 4問題: zfsモジュールを読み込むとload4に貼りつく問題。

グラフ中で、load の高い部分がload 4 を超えている部分。



問題点1(cont.)

load 4の原因。

topで調べると以下の4つのZFSのdaemonが原因

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
24426	root	0	-20	0	0	0	D	0.0	0.0	0:00.00	arc_reclaim
24427	root	0	-20	0	0	0	D	0.0	0.0	0:00.00	l2arc_feed
24708	root	0	-20	0	0	0	D	0.0	0.0	0:00.00	txg_quiesce
24709	root	0	-20	0	0	0	D	0.0	0.0	0:00.00	txg_sync

問題点2

VMWareで試したところ、kernel panic でお亡くなりになった。

実機ではkernel panic は起きなかった。

ZFS on Fuse

Fuseを利用したZFSの実装

Fuse(Filesystem in Userspace) を使用することで
　CDDL/GPLのライセンス問題を解決している。

fuse：http://fuse.sourceforge.net/

zfs on fuse：http://zfs-fuse.net/

Native ZFS on Linux by KQ infotech

kernel module としての実装を、KQ infotechという企業がリリースすると発表された。

[Phoronix] Native ZFS Is Coming To Linux Next Month

http://www.phoronix.com/scan.php?page=article&item=zfs_linux_coming&num=1

Native ZFS on Linux by KQ infotech

- バイナリrpm と Debian 用のビルドできるソースをクローズドbetaとしてリリースする。
- Oracleは法的措置はとらない、と考えている。
- betaリリース後の動向は不明。
- 9/15ごろリリース予定。

参考文献

ZFS 仮想化されたファイルシステムの徹底活用
ASCII ISBN978-4-04-867654-0 C3004