Journées Mathrice, Caen
Mercredi 27 mars 2013

# DragonflyBSD

David DELAVENNAT
Centre de Mathématiques Laurent Schwartz
Ecole Polytechnique

# Agenda de la présentation

1 I Contexte : DragonflyBSD / Mathrice

2 I Stockage : Pilotes RAID, Systèmes de fichier Hammer, Quota VFS

3 I Identification / Authentification : nss_ldap / kerberos

4 I Userland : pkgsrc / dports

5 I Performances : 10Gbe, scheduler & swapcache, tmpfs

6 I Conclusion : perspectives

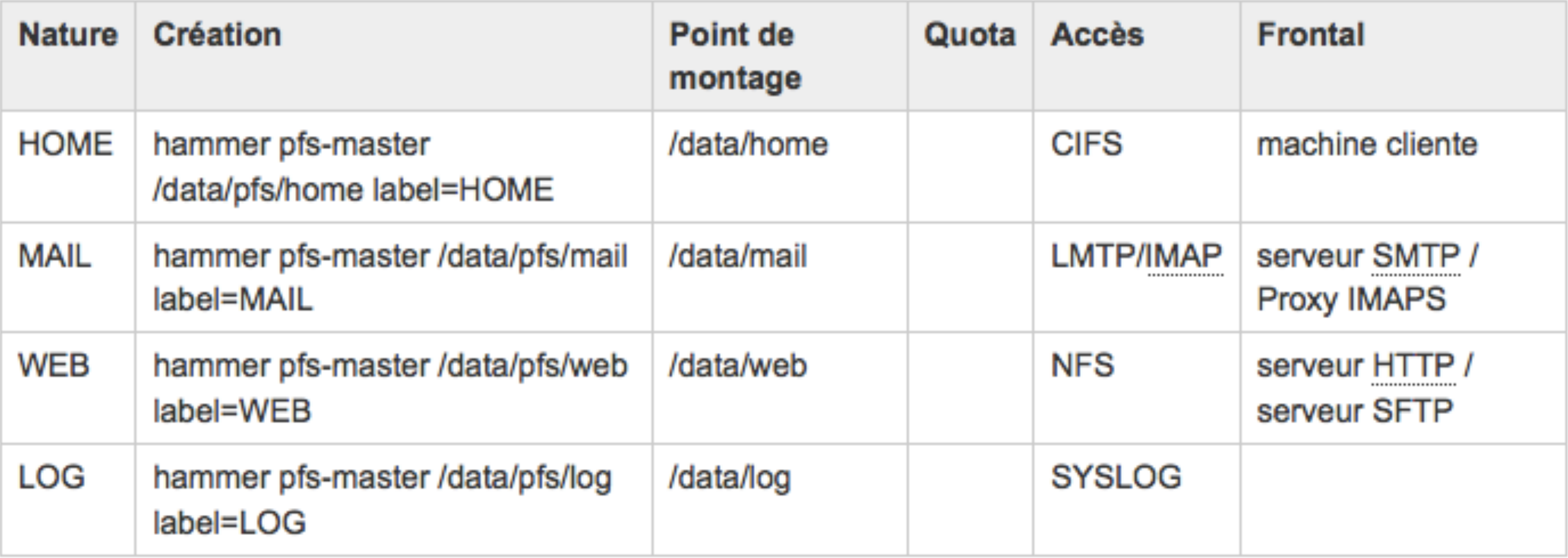
# 1 I Contexte : DragonflyBSD / Mathrice

- ANF Mathrice Mai 2012 ➔ ? Sécurisation des filers ?
  - Ext4 + DRBD ?
  - HammerFS + mirror-stream ?
    - Pilotes matériels DELL ?
    - Réactivité des développeurs, support ?
    - « Entreprise ready NAS »?

# 2 I Pilote RAID, Système de fichier Hammer, Quota VFS

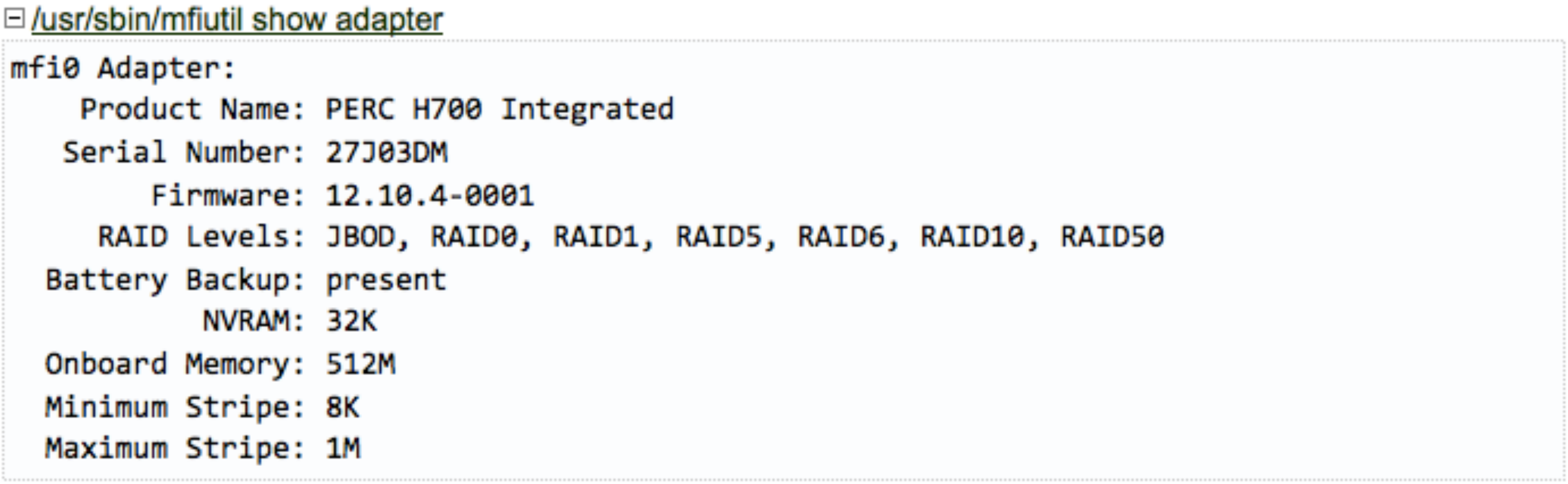
1 DELL R510

- 1 carte réseau Intel X-520 10Gb/s cuivre
- 1 carte RAID PERC H700
- 2 disques 2,5'', 300Go, SAS 6Gb/s
- 12 disques 3,5'', 300Go, SAS 6Gb/s



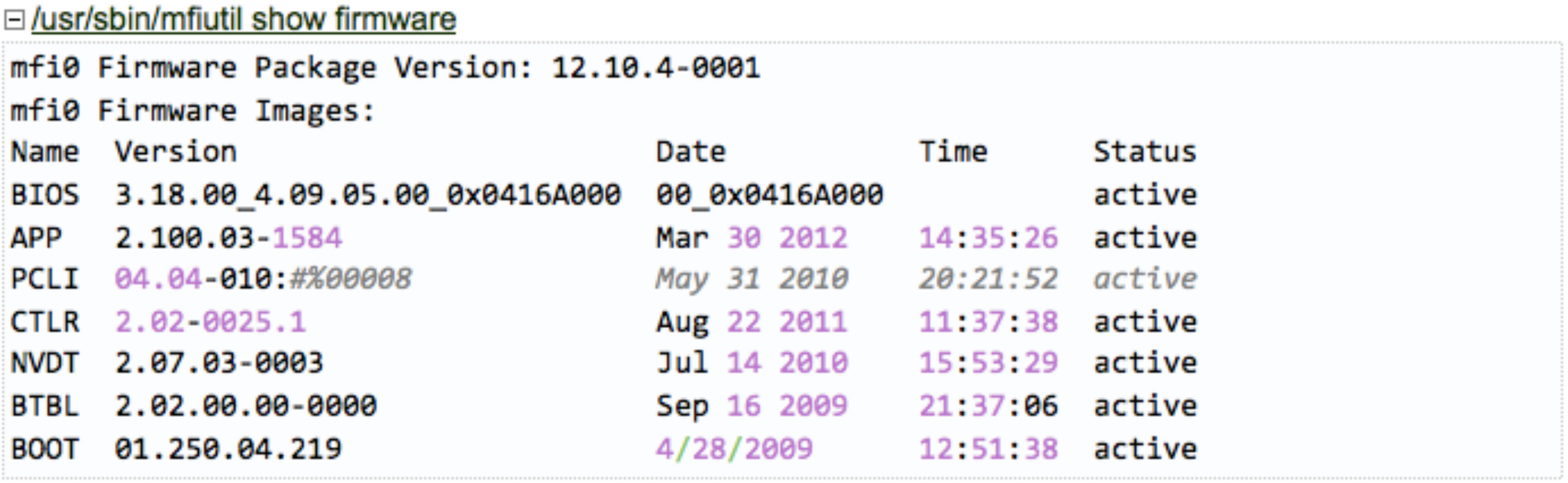
| Nature | Création | Point de montage | Quota | Accès | Frontal |
|---|---|---|---|---|---|
| HOME | hammer pfs-master /data/pfs/home label=HOME | /data/home | | CIFS | machine cliente |
| MAIL | hammer pfs-master /data/pfs/mail label=MAIL | /data/mail | | LMTP/IMAP | serveur SMTP / Proxy IMAPS |
| WEB | hammer pfs-master /data/pfs/web label=WEB | /data/web | | NFS | serveur HTTP / serveur SFTP |
| LOG | hammer pfs-master /data/pfs/log label=LOG | /data/log | | SYSLOG | |

# 2 I Pilote RAID, Système de fichier Hammer, Quota VFS



⊟ /usr/sbin/mfiutil show adapter

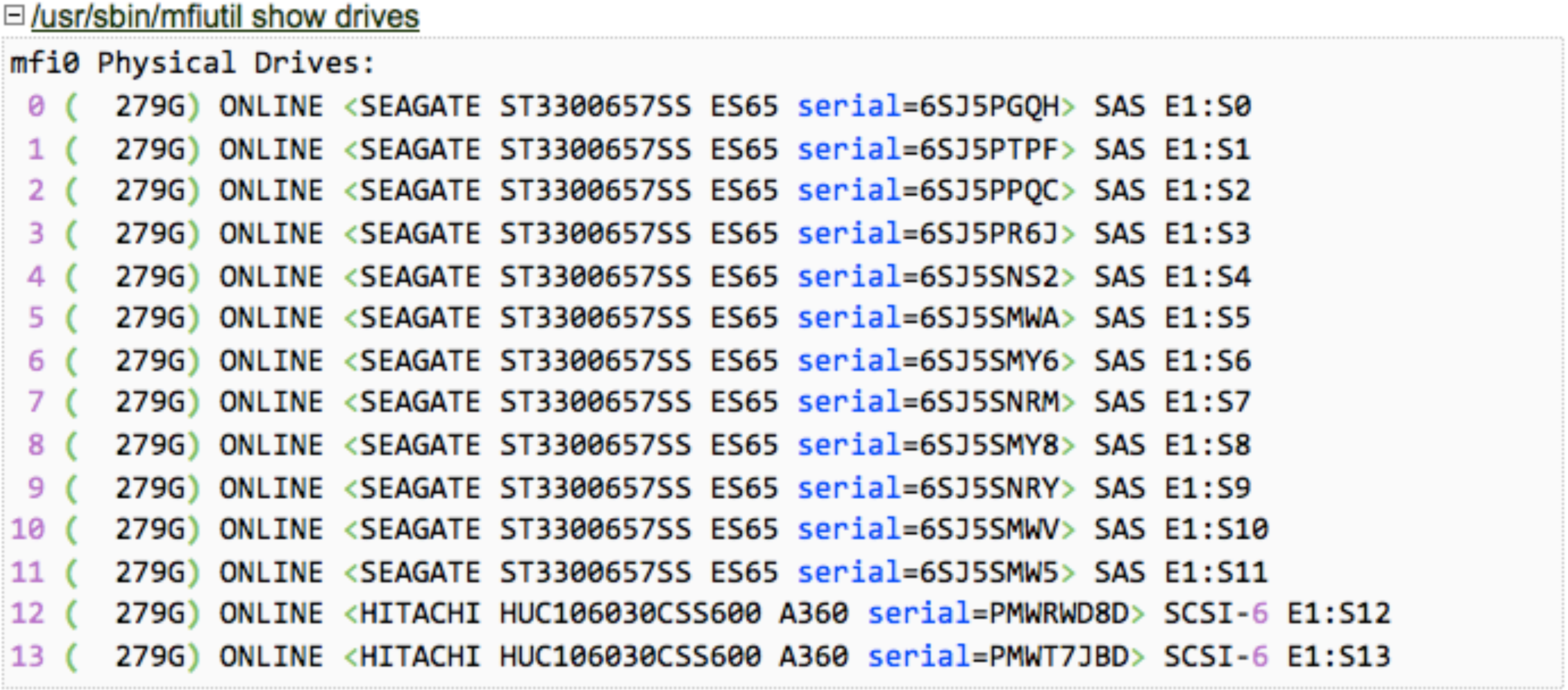
```
mfi0 Adapter:
    Product Name: PERC H700 Integrated
   Serial Number: 27J03DM
        Firmware: 12.10.4-0001
     RAID Levels: JBOD, RAID0, RAID1, RAID5, RAID6, RAID10, RAID50
  Battery Backup: present
           NVRAM: 32K
  Onboard Memory: 512M
  Minimum Stripe: 8K
  Maximum Stripe: 1M
```



⊟ /usr/sbin/mfiutil show firmware

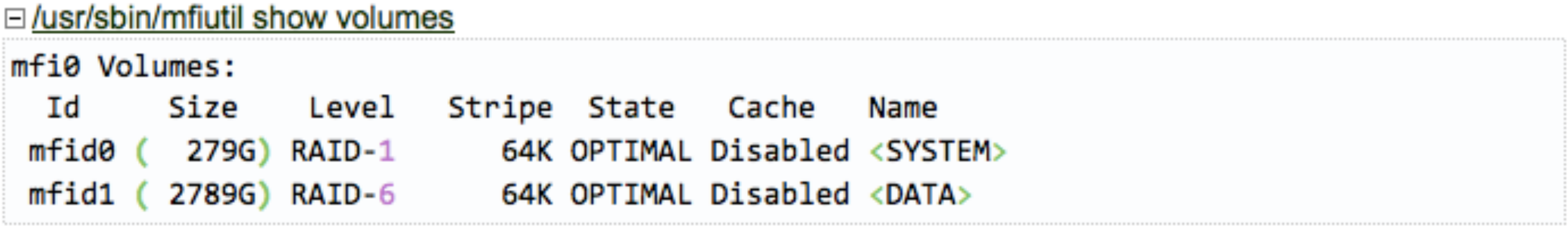
```
mfi0 Firmware Package Version: 12.10.4-0001
mfi0 Firmware Images:
Name  Version                      Date           Time      Status
BIOS  3.18.00_4.09.05.00_0x0416A000 00_0x0416A000           active
APP   2.100.03-1584                Mar 30 2012    14:35:26  active
PCLI  04.04-010:#%00008            May 31 2010    20:21:52  active
CTLR  2.02-0025.1                  Aug 22 2011    11:37:38  active
NVDT  2.07.03-0003                 Jul 14 2010    15:53:29  active
BTBL  2.02.00.00-0000              Sep 16 2009    21:37:06  active
BOOT  01.250.04.219                4/28/2009      12:51:38  active
```

# 2 I Pilote RAID, Système de fichier Hammer, Quota VFS



⊟ /usr/sbin/mfiutil show drives

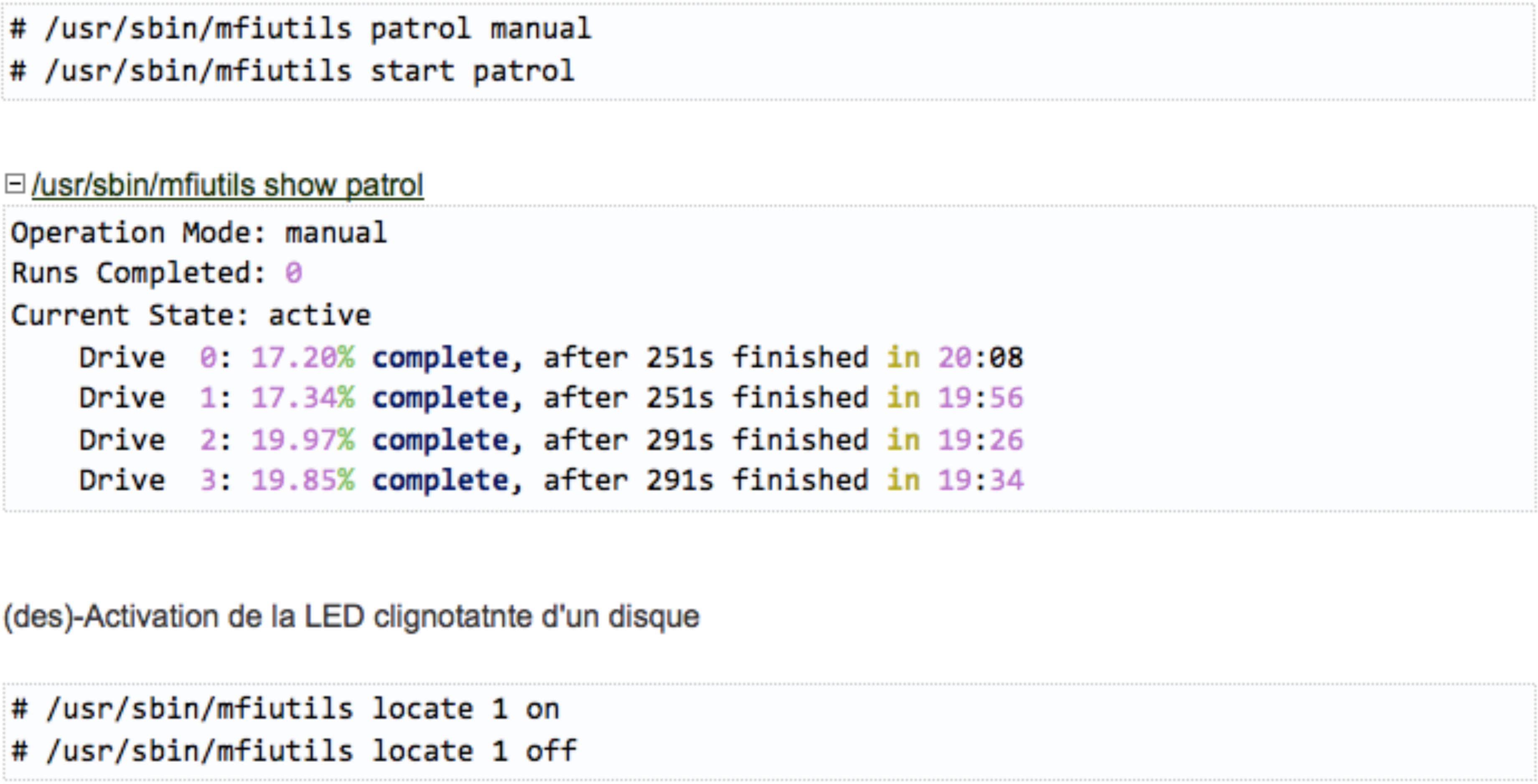
```
mfi0 Physical Drives:
 0 (  279G) ONLINE <SEAGATE ST3300657SS ES65 serial=6SJ5PGQH> SAS E1:S0
 1 (  279G) ONLINE <SEAGATE ST3300657SS ES65 serial=6SJ5PTPF> SAS E1:S1
 2 (  279G) ONLINE <SEAGATE ST3300657SS ES65 serial=6SJ5PPQC> SAS E1:S2
 3 (  279G) ONLINE <SEAGATE ST3300657SS ES65 serial=6SJ5PR6J> SAS E1:S3
 4 (  279G) ONLINE <SEAGATE ST3300657SS ES65 serial=6SJ5SNS2> SAS E1:S4
 5 (  279G) ONLINE <SEAGATE ST3300657SS ES65 serial=6SJ5SMWA> SAS E1:S5
 6 (  279G) ONLINE <SEAGATE ST3300657SS ES65 serial=6SJ5SMY6> SAS E1:S6
 7 (  279G) ONLINE <SEAGATE ST3300657SS ES65 serial=6SJ5SNRM> SAS E1:S7
 8 (  279G) ONLINE <SEAGATE ST3300657SS ES65 serial=6SJ5SMY8> SAS E1:S8
 9 (  279G) ONLINE <SEAGATE ST3300657SS ES65 serial=6SJ5SNRY> SAS E1:S9
10 (  279G) ONLINE <SEAGATE ST3300657SS ES65 serial=6SJ5SMWV> SAS E1:S10
11 (  279G) ONLINE <SEAGATE ST3300657SS ES65 serial=6SJ5SMW5> SAS E1:S11
12 (  279G) ONLINE <HITACHI HUC106030CSS600 A360 serial=PMWRWD8D> SCSI-6 E1:S12
13 (  279G) ONLINE <HITACHI HUC106030CSS600 A360 serial=PMWT7JBD> SCSI-6 E1:S13
```



⊟ /usr/sbin/mfiutil show volumes

```
mfi0 Volumes:
  Id       Size     Level    Stripe  State    Cache    Name
 mfid0 (  279G) RAID-1      64K OPTIMAL Disabled <SYSTEM>
 mfid1 ( 2789G) RAID-6      64K OPTIMAL Disabled <DATA>
```

# 2 I Pilote RAID, Système de fichier Hammer, Quota VFS



```
# /usr/sbin/mfiutils patrol manual
# /usr/sbin/mfiutils start patrol
```

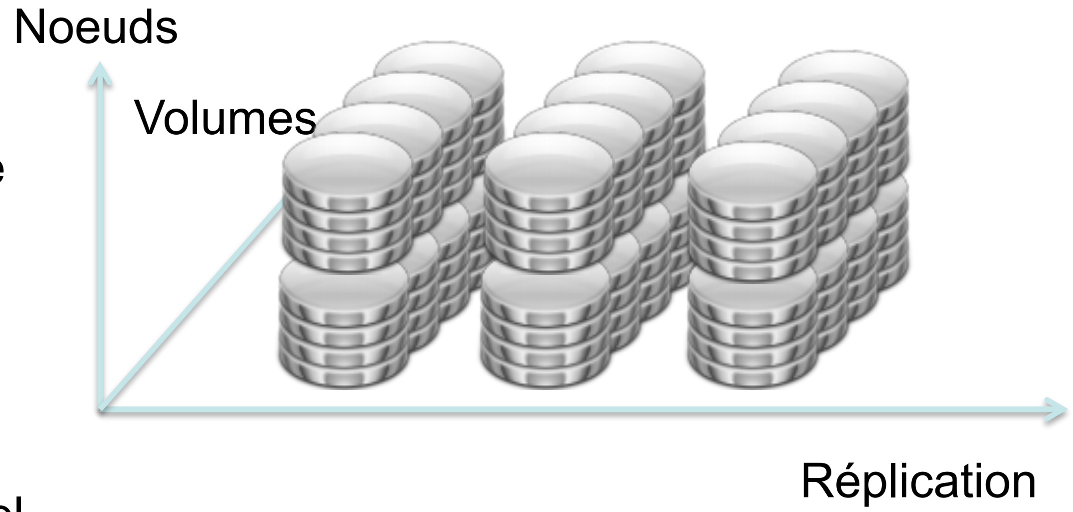
⊟ /usr/sbin/mfiutils show patrol

```
Operation Mode: manual
Runs Completed: 0
Current State: active
    Drive  0: 17.20% complete, after 251s finished in 20:08
    Drive  1: 17.34% complete, after 251s finished in 19:56
    Drive  2: 19.97% complete, after 291s finished in 19:26
    Drive  3: 19.85% complete, after 291s finished in 19:34
```

(des)-Activation de la LED clignotatnte d'un disque

```
# /usr/sbin/mfiutils locate 1 on
# /usr/sbin/mfiutils locate 1 off
```

# 2 | Pilote RAID, Système de fichier Hammer, Quota VFS

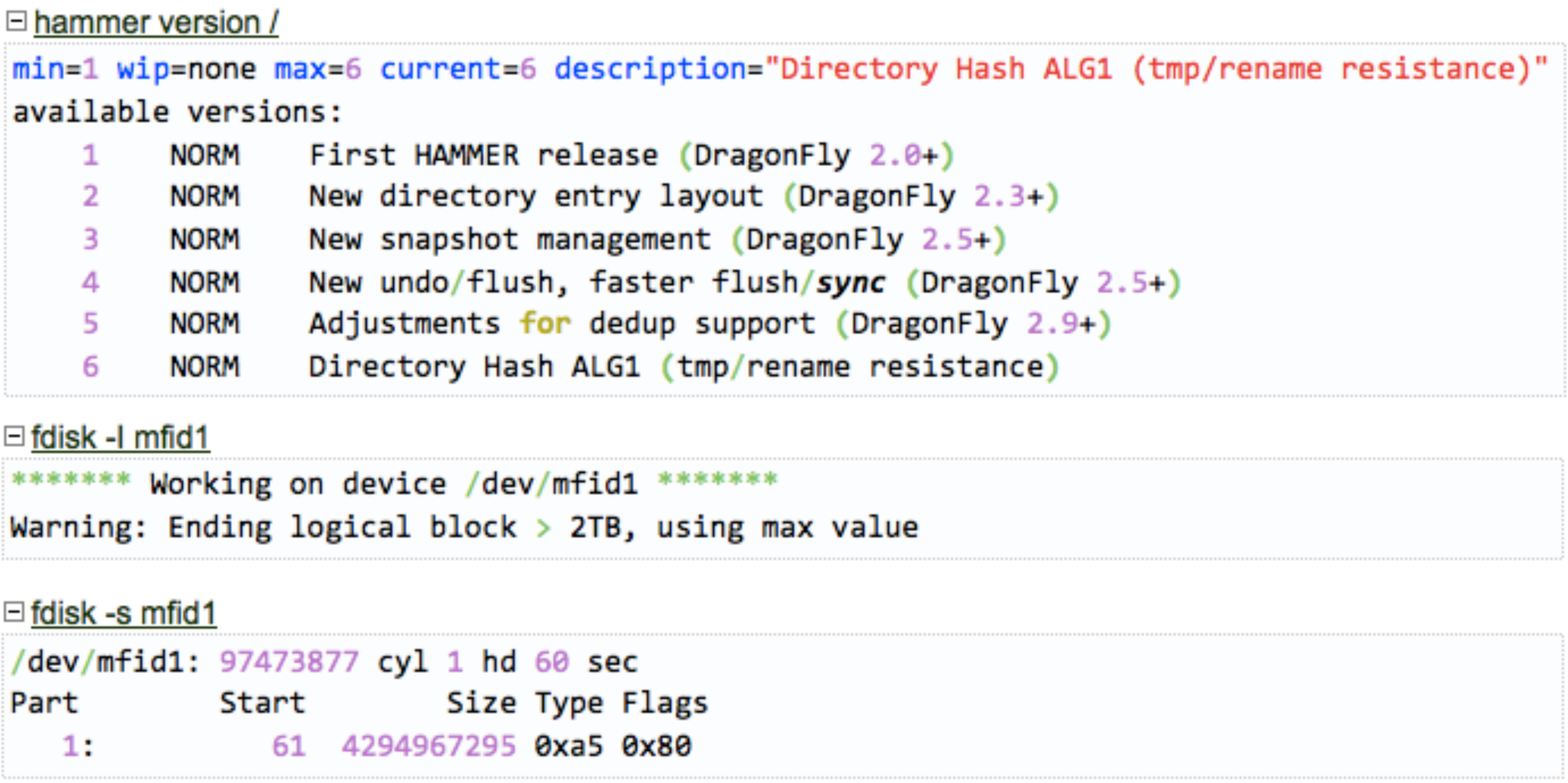


- Multi-volumes
  - 1 slice ou partition / Volume
  - 256 volumes
  - 4096To/Volume
- ➚ 1 Exaoctet
- Système de fichier transactionnel
  - Chaque sync est une transaction
- Snapshots « modernes »
  - Snapshot ➔ lien explicite vers une transaction

```
filesystem@transaction_N-1
filesystem@transaction_N    ➔   /snapshots/2013-01-22_22h10
filesystem@transaction_N+1
filesystem@transaction_N+2
```

- Déduplication
- Réplication
  - Master / Multi-Slaves

# 2 I Pilote RAID, Système de fichier Hammer, Quota VFS



⊟ hammer version /

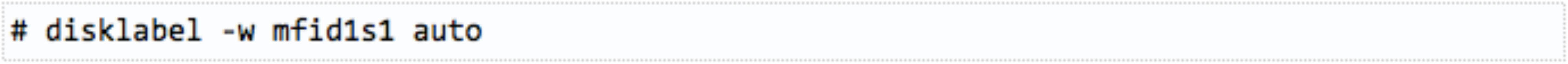
```
min=1 wip=none max=6 current=6 description="Directory Hash ALG1 (tmp/rename resistance)"
available versions:
    1      NORM    First HAMMER release (DragonFly 2.0+)
    2      NORM    New directory entry layout (DragonFly 2.3+)
    3      NORM    New snapshot management (DragonFly 2.5+)
    4      NORM    New undo/flush, faster flush/sync (DragonFly 2.5+)
    5      NORM    Adjustments for dedup support (DragonFly 2.9+)
    6      NORM    Directory Hash ALG1 (tmp/rename resistance)
```

⊟ fdisk -l mfid1

```
******* Working on device /dev/mfid1 *******
Warning: Ending logical block > 2TB, using max value
```

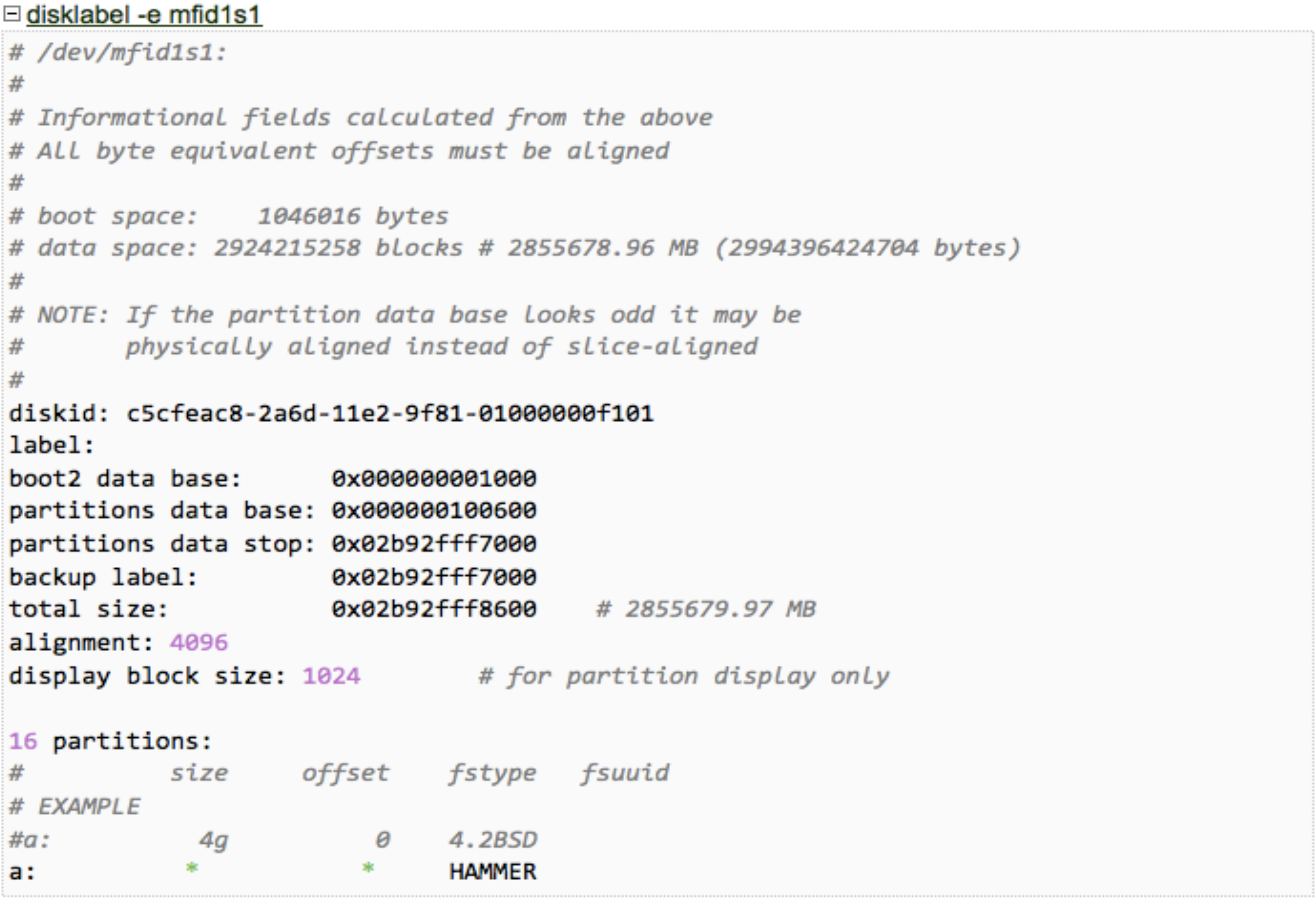
⊟ fdisk -s mfid1

```
/dev/mfid1: 97473877 cyl 1 hd 60 sec
Part        Start        Size Type Flags
   1:          61  4294967295 0xa5 0x80
```



```
# disklabel -w mfid1s1 auto
```

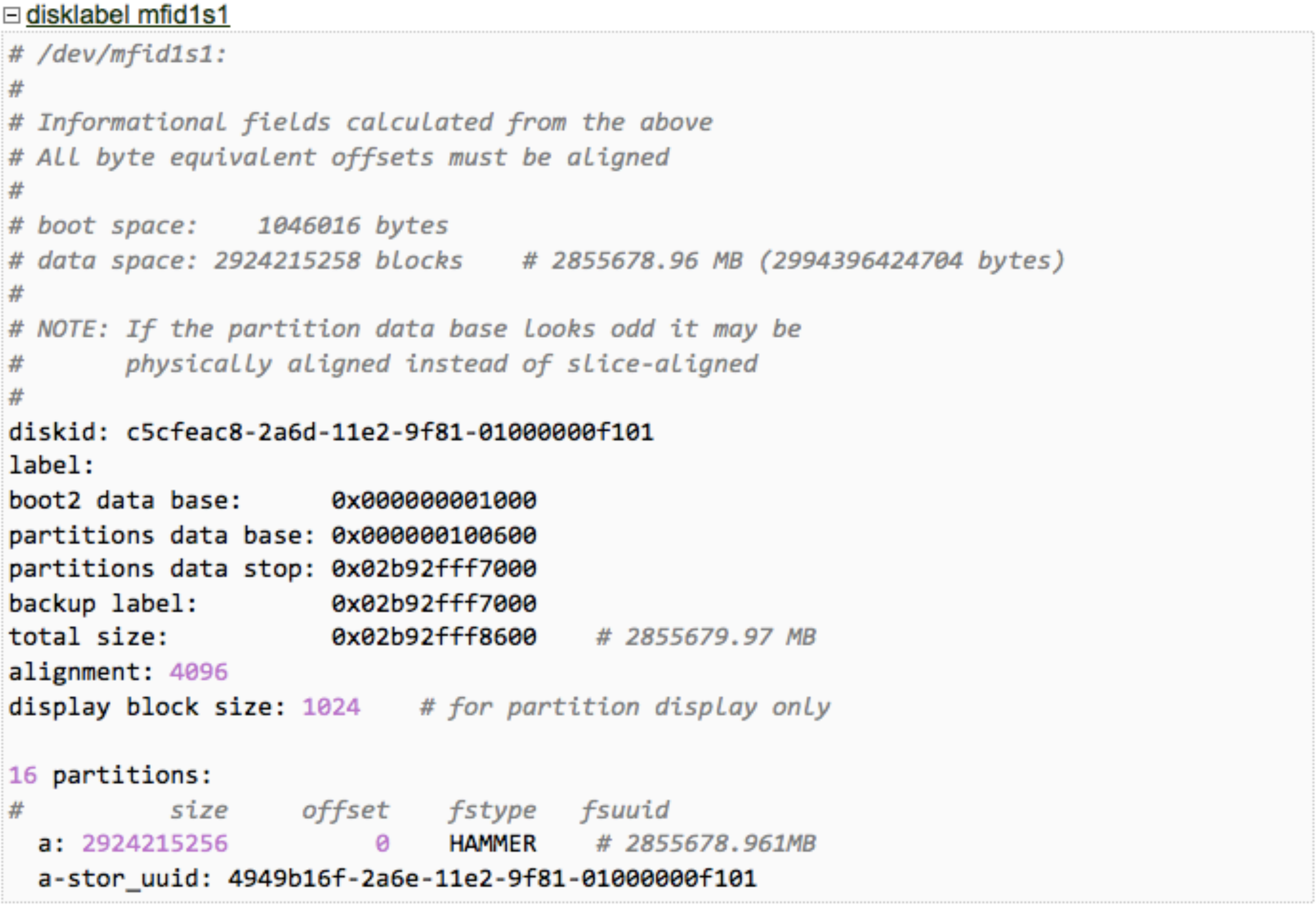
# 2 I Pilote RAID, Système de fichier Hammer, Quota VFS



⊟ disklabel -e mfid1s1

```
# /dev/mfid1s1:
#
# Informational fields calculated from the above
# All byte equivalent offsets must be aligned
#
# boot space:    1046016 bytes
# data space: 2924215258 blocks # 2855678.96 MB (2994396424704 bytes)
#
# NOTE: If the partition data base looks odd it may be
#       physically aligned instead of slice-aligned
#
diskid: c5cfeac8-2a6d-11e2-9f81-01000000f101
label:
boot2 data base:      0x000000001000
partitions data base: 0x000000100600
partitions data stop: 0x02b92fff7000
backup label:         0x02b92fff7000
total size:           0x02b92fff8600    # 2855679.97 MB
alignment: 4096
display block size: 1024        # for partition display only

16 partitions:
#          size     offset    fstype   fsuuid
# EXAMPLE
#a:          4g          0    4.2BSD
a:          *           *     HAMMER
```

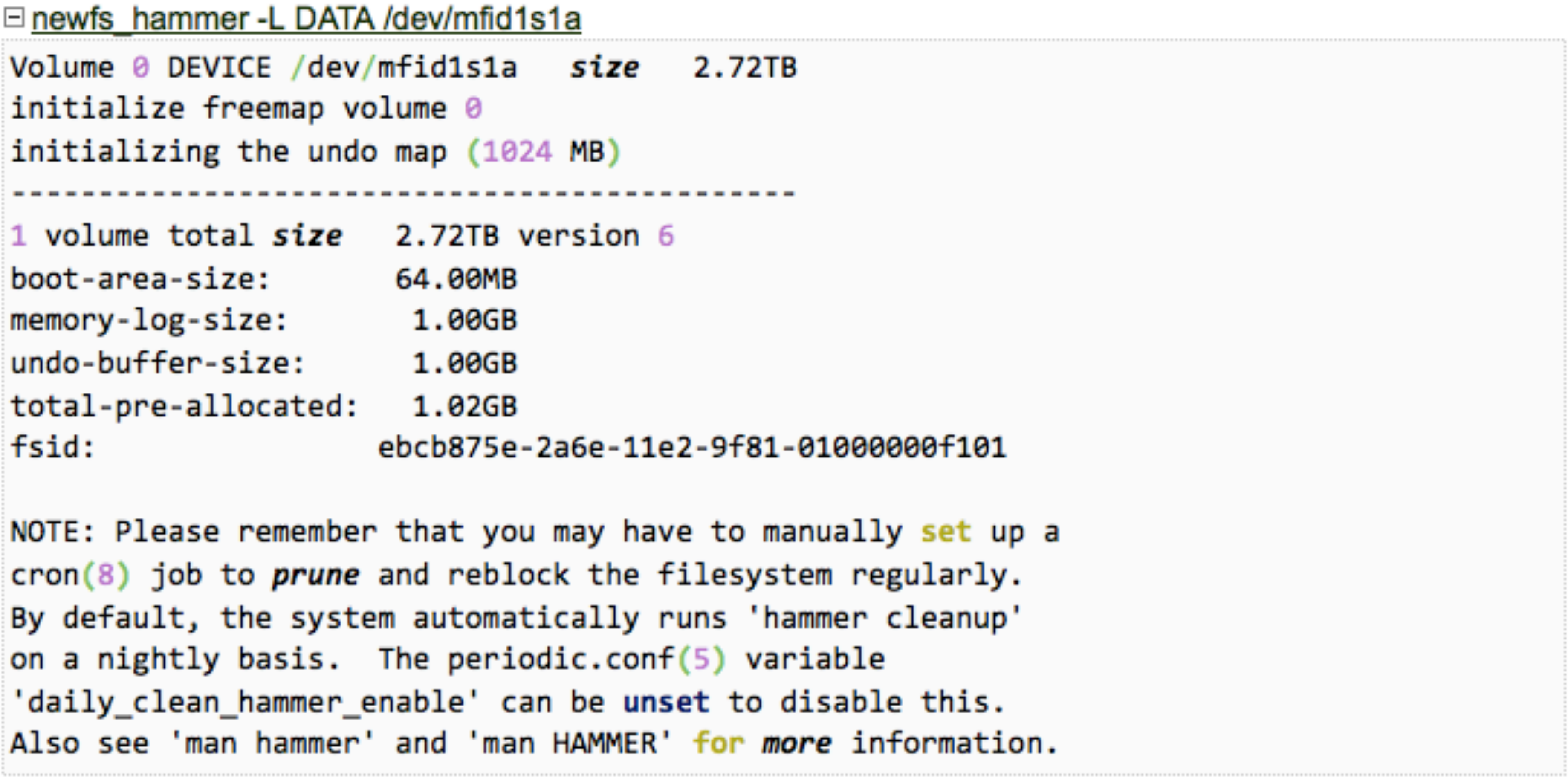
# 2 | Pilote RAID, Système de fichier Hammer, Quota VFS



⊟ disklabel mfid1s1

```
# /dev/mfid1s1:
#
# Informational fields calculated from the above
# All byte equivalent offsets must be aligned
#
# boot space:    1046016 bytes
# data space: 2924215258 blocks    # 2855678.96 MB (2994396424704 bytes)
#
# NOTE: If the partition data base looks odd it may be
#       physically aligned instead of slice-aligned
#
diskid: c5cfeac8-2a6d-11e2-9f81-01000000f101
label:
boot2 data base:      0x000000001000
partitions data base: 0x000000100600
partitions data stop: 0x02b92fff7000
backup label:         0x02b92fff7000
total size:           0x02b92fff8600    # 2855679.97 MB
alignment: 4096
display block size: 1024    # for partition display only

16 partitions:
#          size     offset    fstype   fsuuid
  a: 2924215256          0    HAMMER    # 2855678.961MB
  a-stor_uuid: 4949b16f-2a6e-11e2-9f81-01000000f101
```

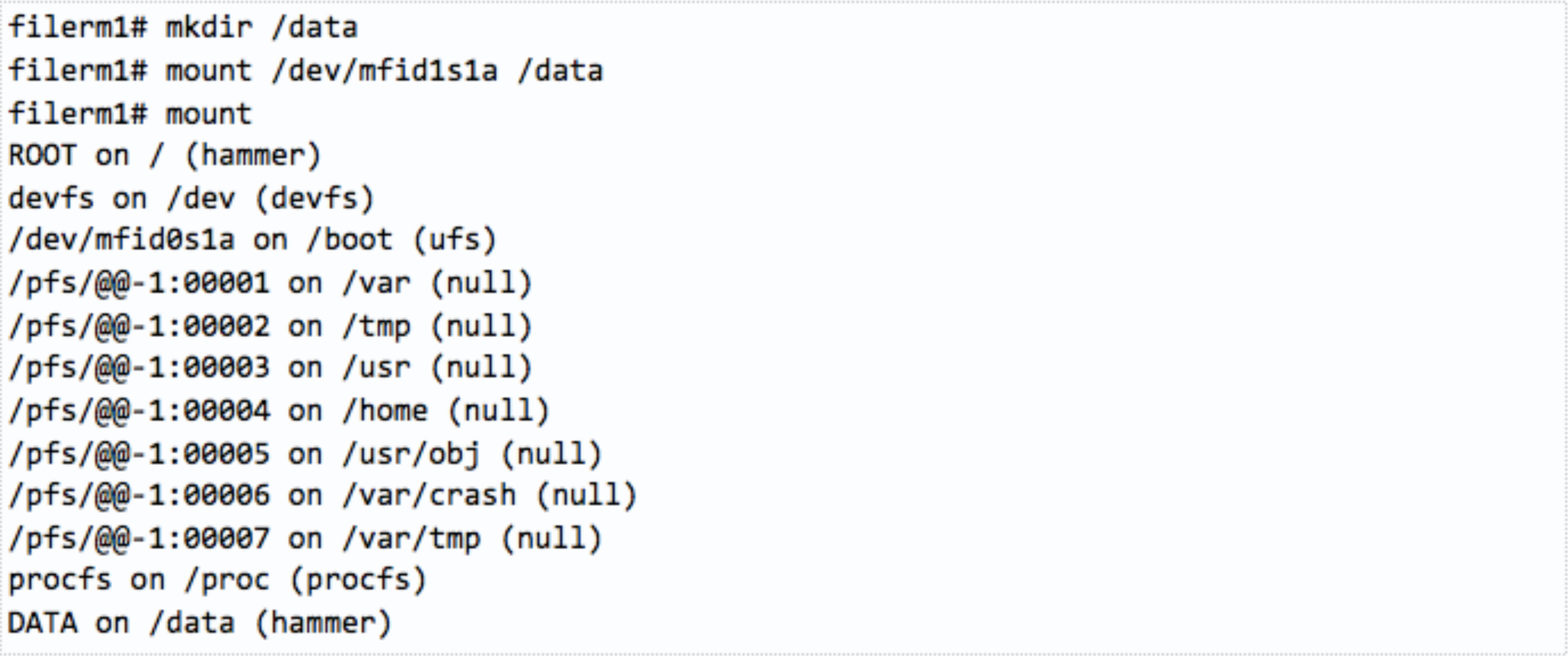
# 2 I Pilote RAID, Système de fichier Hammer, Quota VFS



⊟ newfs_hammer -L DATA /dev/mfid1s1a

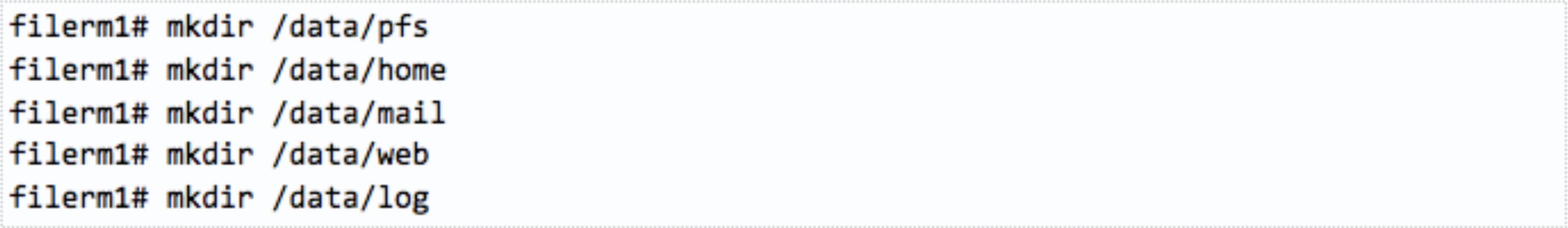
```
Volume 0 DEVICE /dev/mfid1s1a   size   2.72TB
initialize freemap volume 0
initializing the undo map (1024 MB)
---------------------------------------------
1 volume total size   2.72TB version 6
boot-area-size:       64.00MB
memory-log-size:       1.00GB
undo-buffer-size:      1.00GB
total-pre-allocated:   1.02GB
fsid:                ebcb875e-2a6e-11e2-9f81-01000000f101

NOTE: Please remember that you may have to manually set up a
cron(8) job to prune and reblock the filesystem regularly.
By default, the system automatically runs 'hammer cleanup'
on a nightly basis.  The periodic.conf(5) variable
'daily_clean_hammer_enable' can be unset to disable this.
Also see 'man hammer' and 'man HAMMER' for more information.
```

# 2 I Pilote RAID, Système de fichier Hammer, Quota VFS

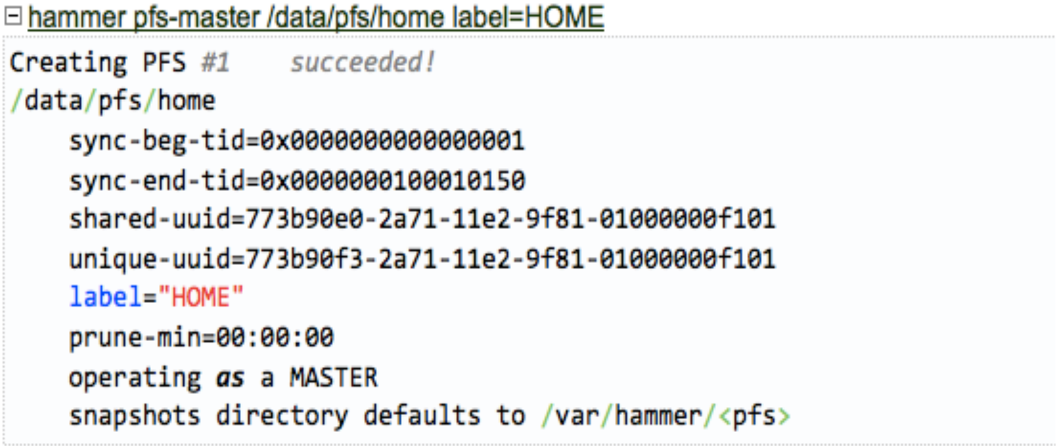


```
filerm1# mkdir /data
filerm1# mount /dev/mfid1s1a /data
filerm1# mount
ROOT on / (hammer)
devfs on /dev (devfs)
/dev/mfid0s1a on /boot (ufs)
/pfs/@@-1:00001 on /var (null)
/pfs/@@-1:00002 on /tmp (null)
/pfs/@@-1:00003 on /usr (null)
/pfs/@@-1:00004 on /home (null)
/pfs/@@-1:00005 on /usr/obj (null)
/pfs/@@-1:00006 on /var/crash (null)
/pfs/@@-1:00007 on /var/tmp (null)
procfs on /proc (procfs)
DATA on /data (hammer)
```

# 2 I Pilote RAID, Système de fichier Hammer, Quota VFS

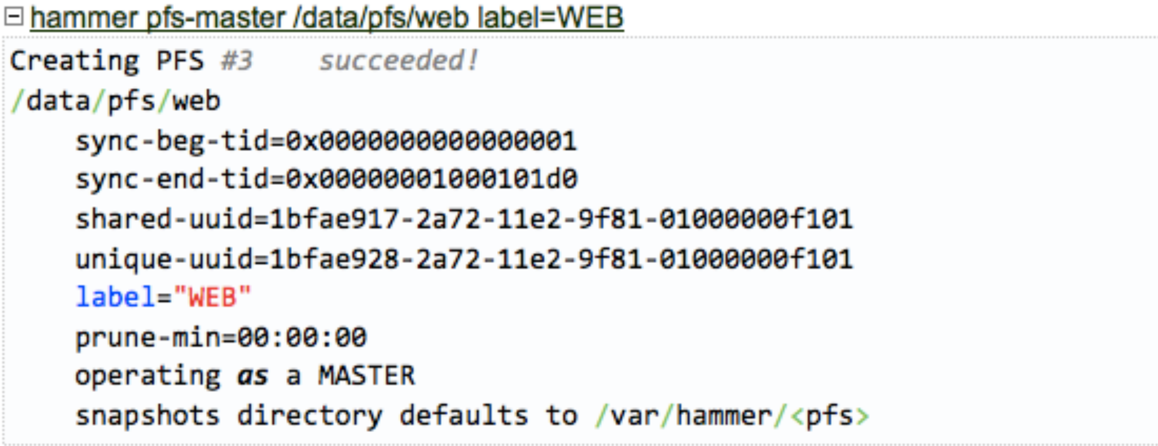


```
filerm1# mkdir /data/pfs
filerm1# mkdir /data/home
filerm1# mkdir /data/mail
filerm1# mkdir /data/web
filerm1# mkdir /data/log
```



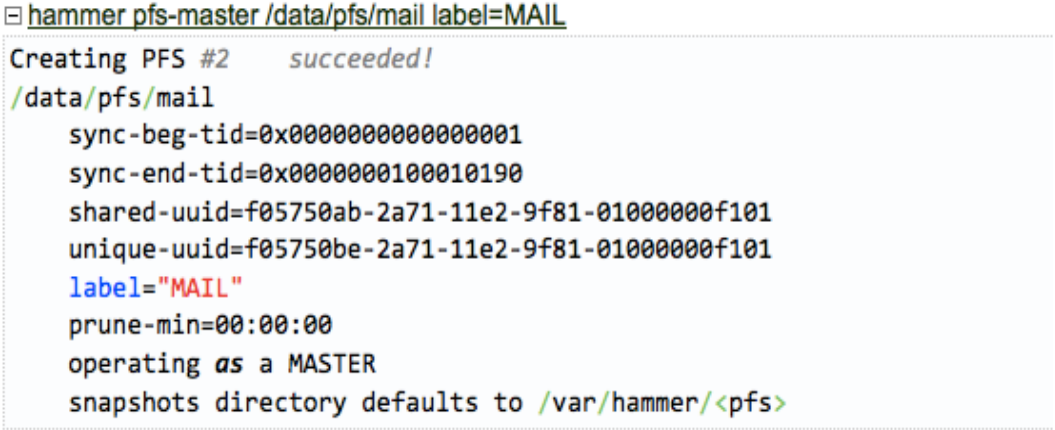
⊟ hammer pfs-master /data/pfs/home label=HOME

```
Creating PFS #1     succeeded!
/data/pfs/home
    sync-beg-tid=0x0000000000000001
    sync-end-tid=0x0000000100010150
    shared-uuid=773b90e0-2a71-11e2-9f81-01000000f101
    unique-uuid=773b90f3-2a71-11e2-9f81-01000000f101
    label="HOME"
    prune-min=00:00:00
    operating as a MASTER
    snapshots directory defaults to /var/hammer/<pfs>
```



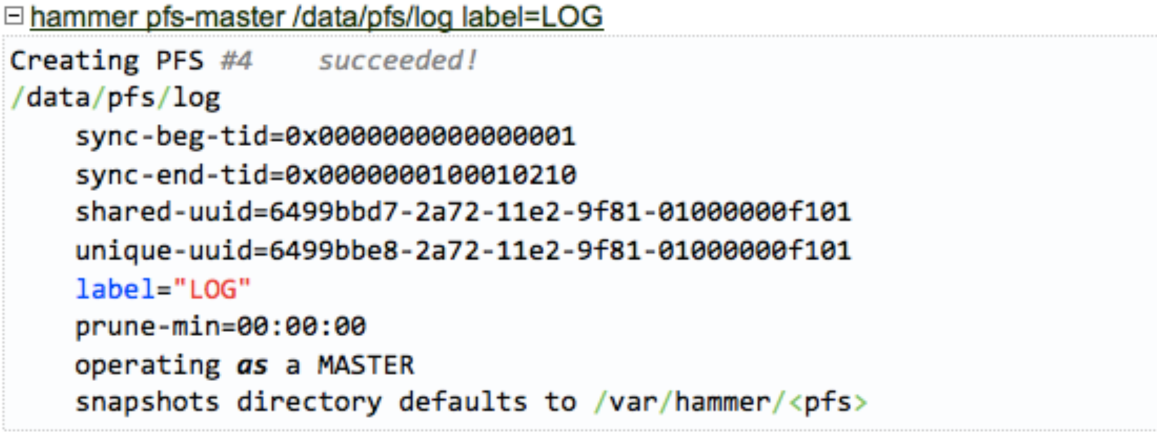
⊟ hammer pfs-master /data/pfs/web label=WEB

```
Creating PFS #3     succeeded!
/data/pfs/web
    sync-beg-tid=0x0000000000000001
    sync-end-tid=0x00000001000101d0
    shared-uuid=1bfae917-2a72-11e2-9f81-01000000f101
    unique-uuid=1bfae928-2a72-11e2-9f81-01000000f101
    label="WEB"
    prune-min=00:00:00
    operating as a MASTER
    snapshots directory defaults to /var/hammer/<pfs>
```



⊟ hammer pfs-master /data/pfs/mail label=MAIL

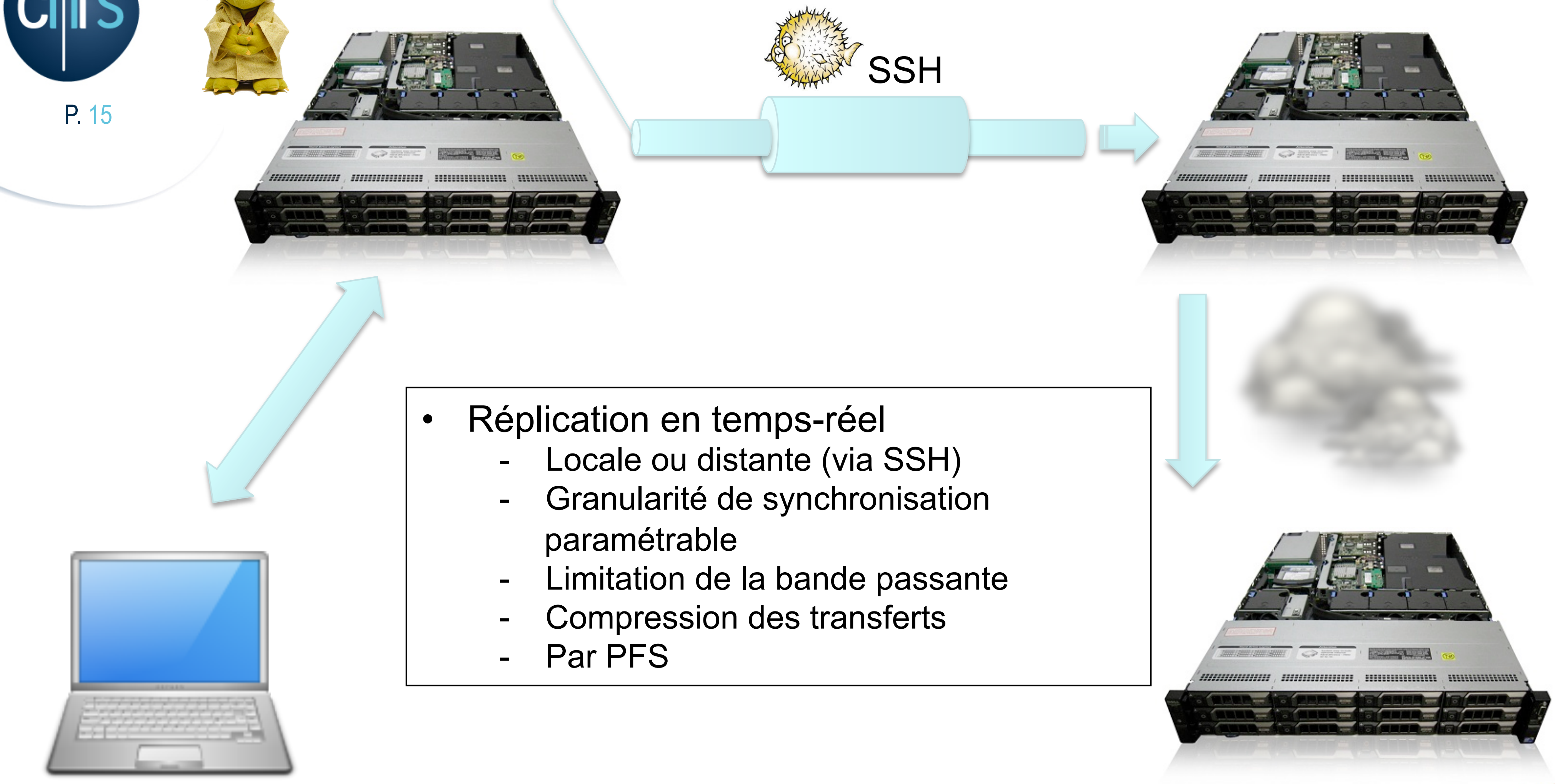
```
Creating PFS #2     succeeded!
/data/pfs/mail
    sync-beg-tid=0x0000000000000001
    sync-end-tid=0x0000000100010190
    shared-uuid=f05750ab-2a71-11e2-9f81-01000000f101
    unique-uuid=f05750be-2a71-11e2-9f81-01000000f101
    label="MAIL"
    prune-min=00:00:00
    operating as a MASTER
    snapshots directory defaults to /var/hammer/<pfs>
```



⊟ hammer pfs-master /data/pfs/log label=LOG

```
Creating PFS #4     succeeded!
/data/pfs/log
    sync-beg-tid=0x0000000000000001
    sync-end-tid=0x0000000100010210
    shared-uuid=6499bbd7-2a72-11e2-9f81-01000000f101
    unique-uuid=6499bbe8-2a72-11e2-9f81-01000000f101
    label="LOG"
    prune-min=00:00:00
    operating as a MASTER
    snapshots directory defaults to /var/hammer/<pfs>
```

# 2 I Pilote RAID, Système de fichier Hammer, Quota VFS

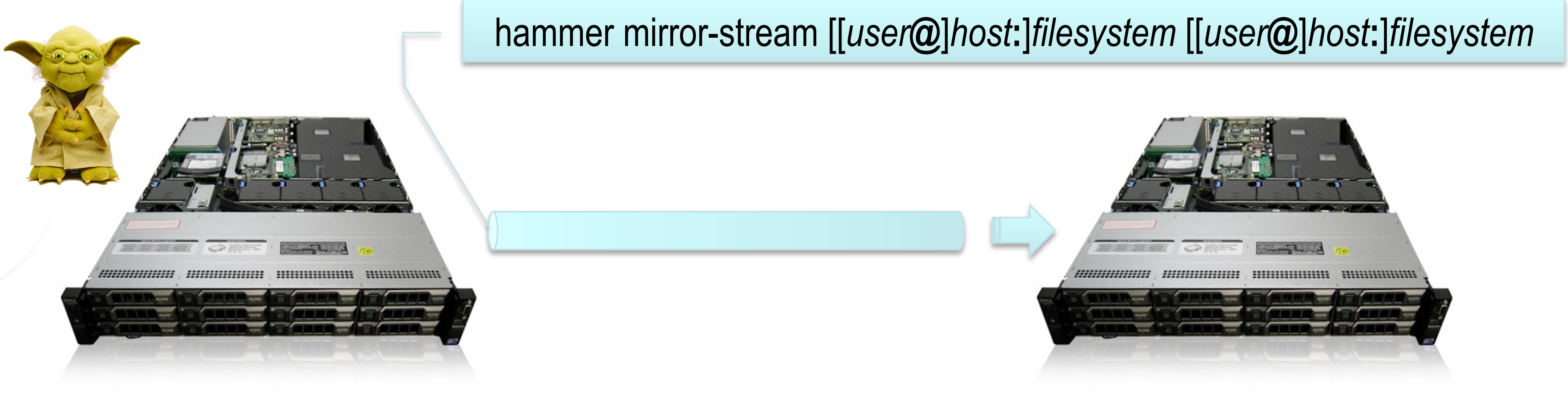


```
hammer mirror-stream [[user@]host:]filesystem [[user@]host:]filesystem
```

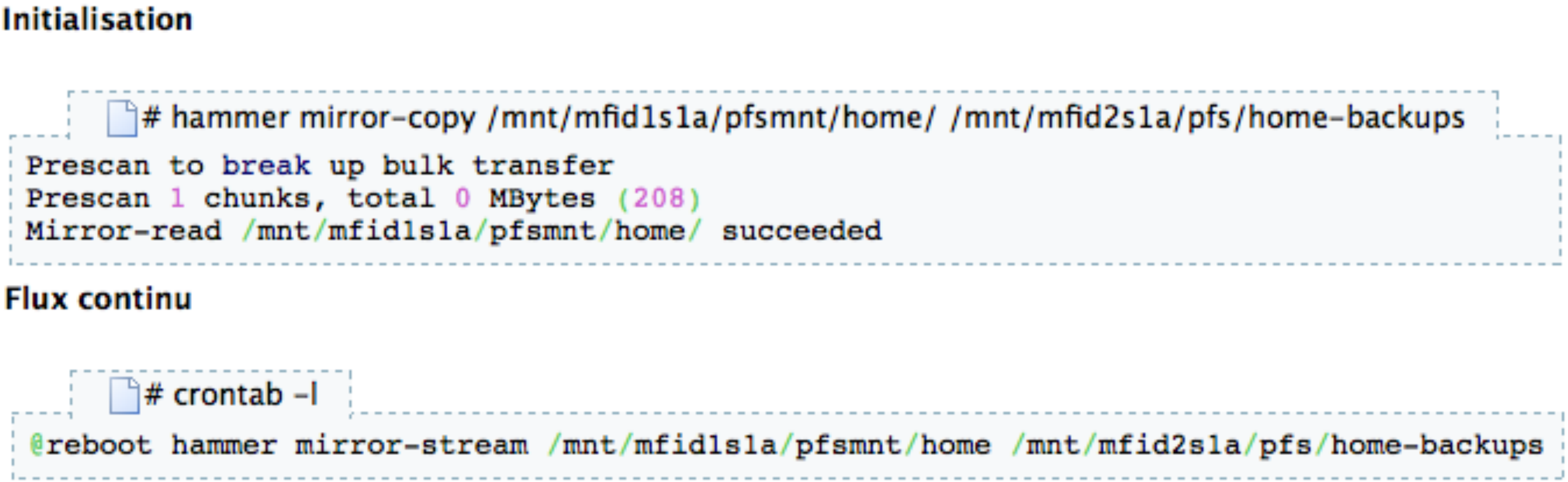
- Réplication en temps-réel
  - Locale ou distante (via SSH)
  - Granularité de synchronisation paramétrable
  - Limitation de la bande passante
  - Compression des transferts
  - Par PFS

http://leaf.dragonflybsd.org/cgi/web-man?command=hammer

# 2 I Pilote RAID, Système de fichier Hammer, Quota VFS



hammer mirror-stream [[*user*@]*host*:]*filesystem* [[*user*@]*host*:]*filesystem*



**Initialisation**

```
# hammer mirror-copy /mnt/mfid1s1a/pfsmnt/home/ /mnt/mfid2s1a/pfs/home-backups
Prescan to break up bulk transfer
Prescan 1 chunks, total 0 MBytes (208)
Mirror-read /mnt/mfid1s1a/pfsmnt/home/ succeeded
```

**Flux continu**

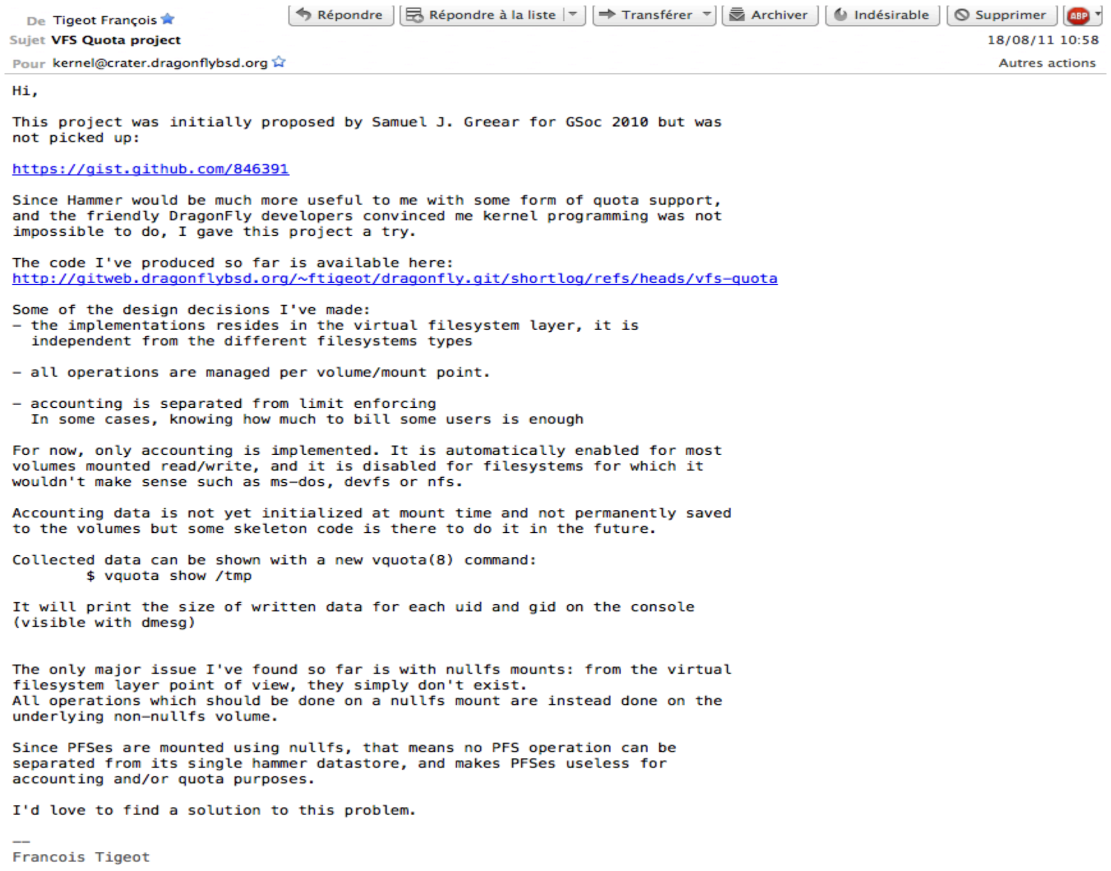
```
# crontab -l
@reboot hammer mirror-stream /mnt/mfid1s1a/pfsmnt/home /mnt/mfid2s1a/pfs/home-backups
```

http://leaf.dragonflybsd.org/cgi/web-man?command=hammer

# 2 I Pilote RAID, Système de fichier Hammer, Quota VFS

Aout 2011

**Objectif : fournir un mécanisme « similaire » à ce qui existe sous XFS ou ZFS**



De Tigeot François
Sujet **VFS Quota project**
Pour kernel@crater.dragonflybsd.org
18/08/11 10:58

Hi,

This project was initially proposed by Samuel J. Greear for GSoc 2010 but was not picked up:

https://gist.github.com/846391

Since Hammer would be much more useful to me with some form of quota support, and the friendly DragonFly developers convinced me kernel programming was not impossible to do, I gave this project a try.

The code I've produced so far is available here:
http://gitweb.dragonflybsd.org/~ftigeot/dragonfly.git/shortlog/refs/heads/vfs-quota

Some of the design decisions I've made:
- the implementations resides in the virtual filesystem layer, it is independent from the different filesystems types

- all operations are managed per volume/mount point.

- accounting is separated from limit enforcing
  In some cases, knowing how much to bill some users is enough

For now, only accounting is implemented. It is automatically enabled for most volumes mounted read/write, and it is disabled for filesystems for which it wouldn't make sense such as ms-dos, devfs or nfs.

Accounting data is not yet initialized at mount time and not permanently saved to the volumes but some skeleton code is there to do it in the future.

Collected data can be shown with a new vquota(8) command:

```
        $ vquota show /tmp
```

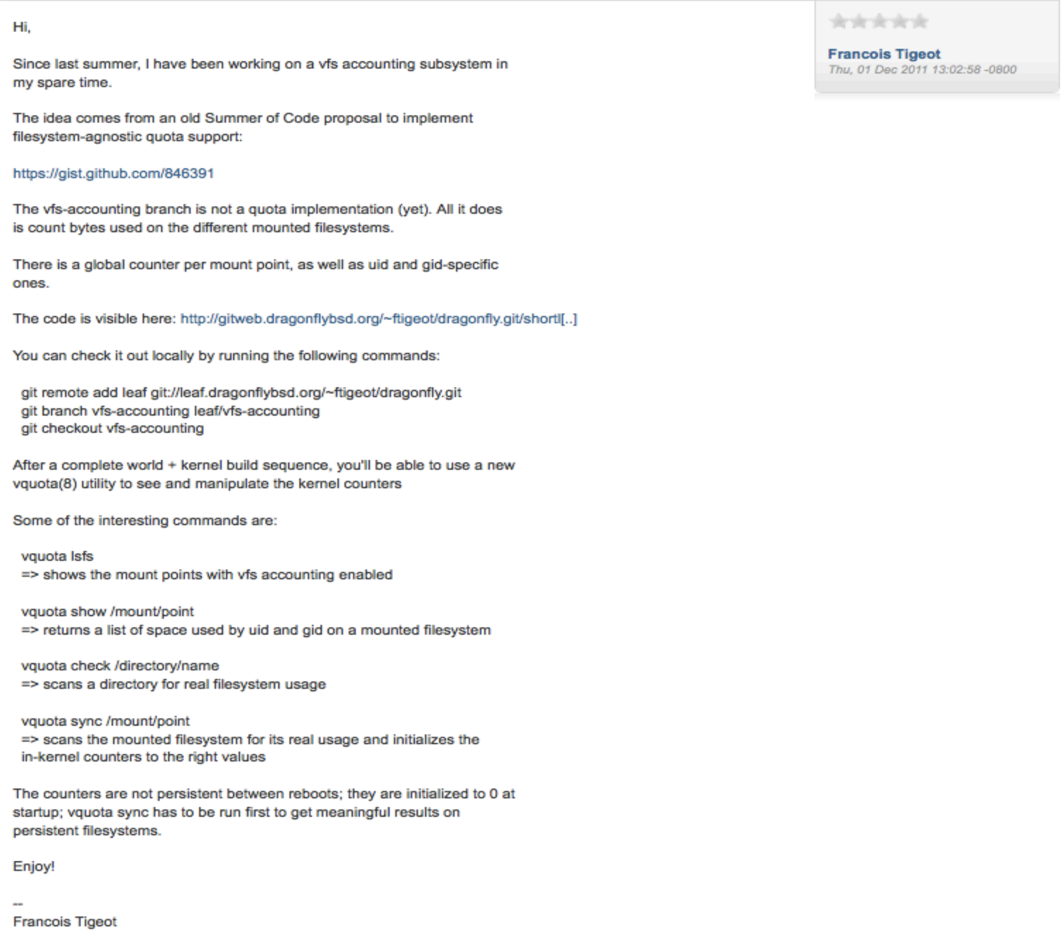
It will print the size of written data for each uid and gid on the console (visible with dmesg)

The only major issue I've found so far is with nullfs mounts: from the virtual filesystem layer point of view, they simply don't exist.
All operations which should be done on a nullfs mount are instead done on the underlying non-nullfs volume.

Since PFSes are mounted using nullfs, that means no PFS operation can be separated from its single hammer datastore, and makes PFSes useless for accounting and/or quota purposes.

I'd love to find a solution to this problem.

--
Francois Tigeot



Francois Tigeot
*Thu, 01 Dec 2011 13:02:58 -0800*

Hi,

Since last summer, I have been working on a vfs accounting subsystem in my spare time.

The idea comes from an old Summer of Code proposal to implement filesystem-agnostic quota support:

https://gist.github.com/846391

The vfs-accounting branch is not a quota implementation (yet). All it does is count bytes used on the different mounted filesystems.

There is a global counter per mount point, as well as uid and gid-specific ones.

The code is visible here: http://gitweb.dragonflybsd.org/~ftigeot/dragonfly.git/shortl[..]

You can check it out locally by running the following commands:

```
 git remote add leaf git://leaf.dragonflybsd.org/~ftigeot/dragonfly.git
 git branch vfs-accounting leaf/vfs-accounting
 git checkout vfs-accounting
```

After a complete world + kernel build sequence, you'll be able to use a new vquota(8) utility to see and manipulate the kernel counters

Some of the interesting commands are:

```
 vquota lsfs
 => shows the mount points with vfs accounting enabled

 vquota show /mount/point
 => returns a list of space used by uid and gid on a mounted filesystem

 vquota check /directory/name
 => scans a directory for real filesystem usage

 vquota sync /mount/point
 => scans the mounted filesystem for its real usage and initializes the
 in-kernel counters to the right values
```

The counters are not persistent between reboots; they are initialized to 0 at startup; vquota sync has to be run first to get meaningful results on persistent filesystems.

Enjoy!

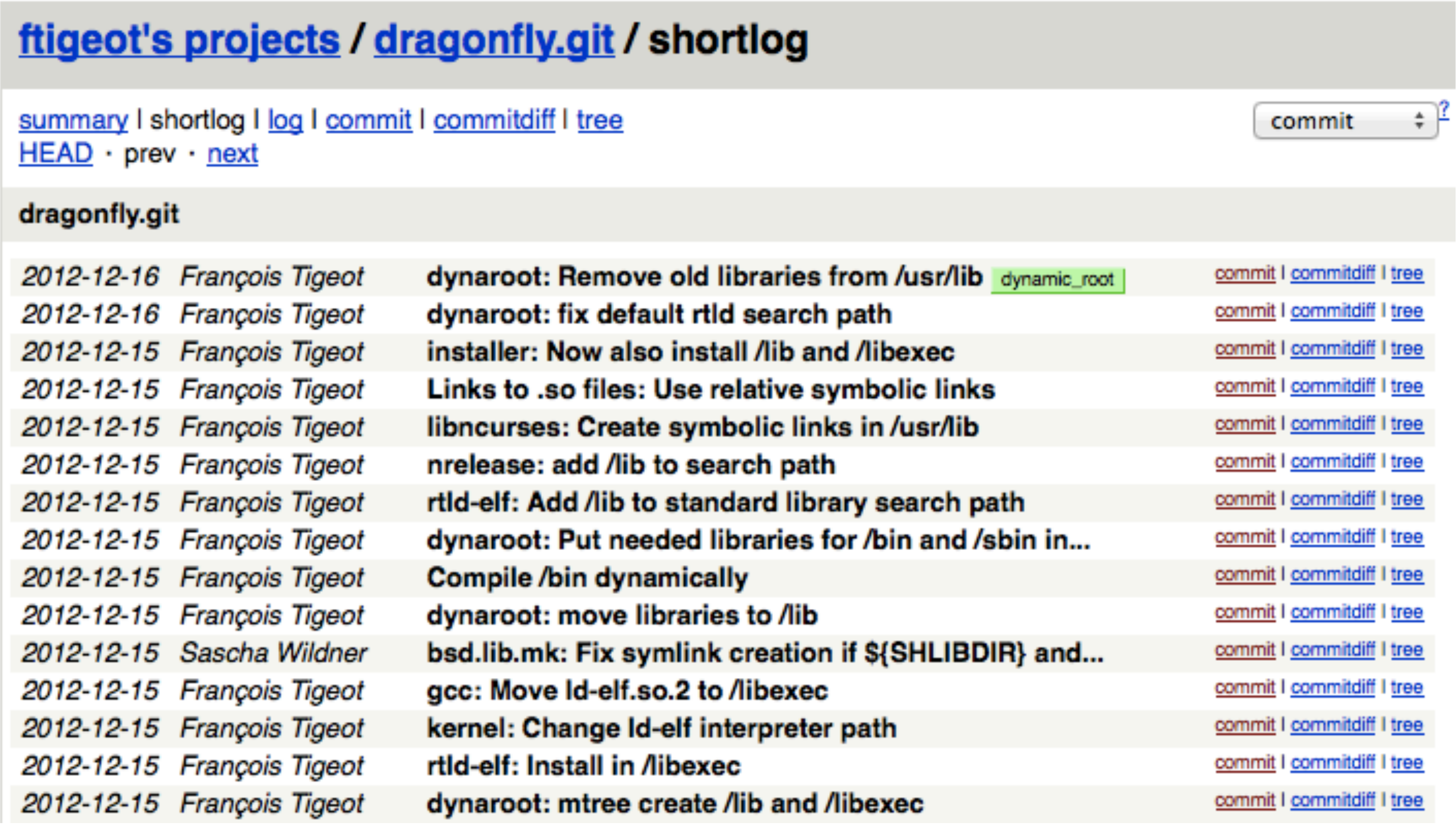
--
Francois Tigeot

# 2 I Pilote RAID, Système de fichier Hammer, Quota VFS

- Sous-système Quota indépendant du système de fichiers (VFS level)
- Les limites sont définies par point de montage et peuvent être positionnées
  - par utilisateur
  - par groupe
  - globalement
- Exemple
  - # vquota limit /tmp 10K
  - # vquota ulim /tmp johndoe 1800
  - Les quotas VFS peuvent être activés pour la plupart des systèmes de fichiers locaux : ext2fs, hammer, hpfs, mfs, ntfs, nullfs, tmpfs et ufs
- Les quotas VFS ne sont pas activés par défaut
  - mettre vfs.quota_enabled="1" dans le fichier /boot/loader.conf
  - rebooter la machine
- cf vquota(8) manpage
  - http://leaf.dragonflybsd.org/cgi/web-man?command=vquota&section=8

# 3 I Identification / Authentification : nss_ldap, kerberos

Décembre 2012 : nsswitch statique ➔ binaires de /bin /sbin dynamiques
➔ nsswitch dynamique



ftigeot's projects / dragonfly.git / shortlog

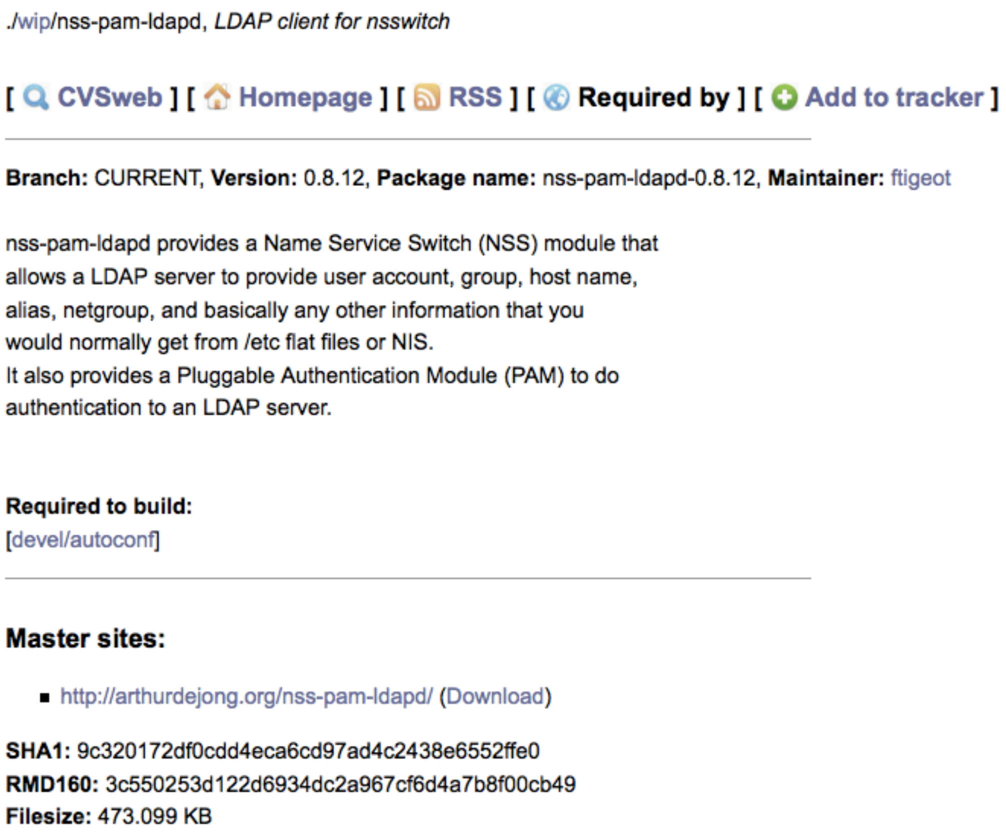
summary | shortlog | log | commit | commitdiff | tree
HEAD · prev · next

dragonfly.git

| | | | |
|---|---|---|---|
| 2012-12-16 | François Tigeot | dynaroot: Remove old libraries from /usr/lib dynamic_root | commit \| commitdiff \| tree |
| 2012-12-16 | François Tigeot | dynaroot: fix default rtld search path | commit \| commitdiff \| tree |
| 2012-12-15 | François Tigeot | installer: Now also install /lib and /libexec | commit \| commitdiff \| tree |
| 2012-12-15 | François Tigeot | Links to .so files: Use relative symbolic links | commit \| commitdiff \| tree |
| 2012-12-15 | François Tigeot | libncurses: Create symbolic links in /usr/lib | commit \| commitdiff \| tree |
| 2012-12-15 | François Tigeot | nrelease: add /lib to search path | commit \| commitdiff \| tree |
| 2012-12-15 | François Tigeot | rtld-elf: Add /lib to standard library search path | commit \| commitdiff \| tree |
| 2012-12-15 | François Tigeot | dynaroot: Put needed libraries for /bin and /sbin in... | commit \| commitdiff \| tree |
| 2012-12-15 | François Tigeot | Compile /bin dynamically | commit \| commitdiff \| tree |
| 2012-12-15 | François Tigeot | dynaroot: move libraries to /lib | commit \| commitdiff \| tree |
| 2012-12-15 | Sascha Wildner | bsd.lib.mk: Fix symlink creation if ${SHLIBDIR} and... | commit \| commitdiff \| tree |
| 2012-12-15 | François Tigeot | gcc: Move ld-elf.so.2 to /libexec | commit \| commitdiff \| tree |
| 2012-12-15 | François Tigeot | kernel: Change ld-elf interpreter path | commit \| commitdiff \| tree |
| 2012-12-15 | François Tigeot | rtld-elf: Install in /libexec | commit \| commitdiff \| tree |
| 2012-12-15 | François Tigeot | dynaroot: mtree create /lib and /libexec | commit \| commitdiff \| tree |

# 3 I Identification / Authentification : nss_ldap, kerberos

Décembre 2012 : port de nss-pam-ldapd



./wip/nss-pam-ldapd, *LDAP client for nsswitch*

[ CVSweb ] [ Homepage ] [ RSS ] [ Required by ] [ Add to tracker ]

**Branch:** CURRENT, **Version:** 0.8.12, **Package name:** nss-pam-ldapd-0.8.12, **Maintainer:** ftigeot

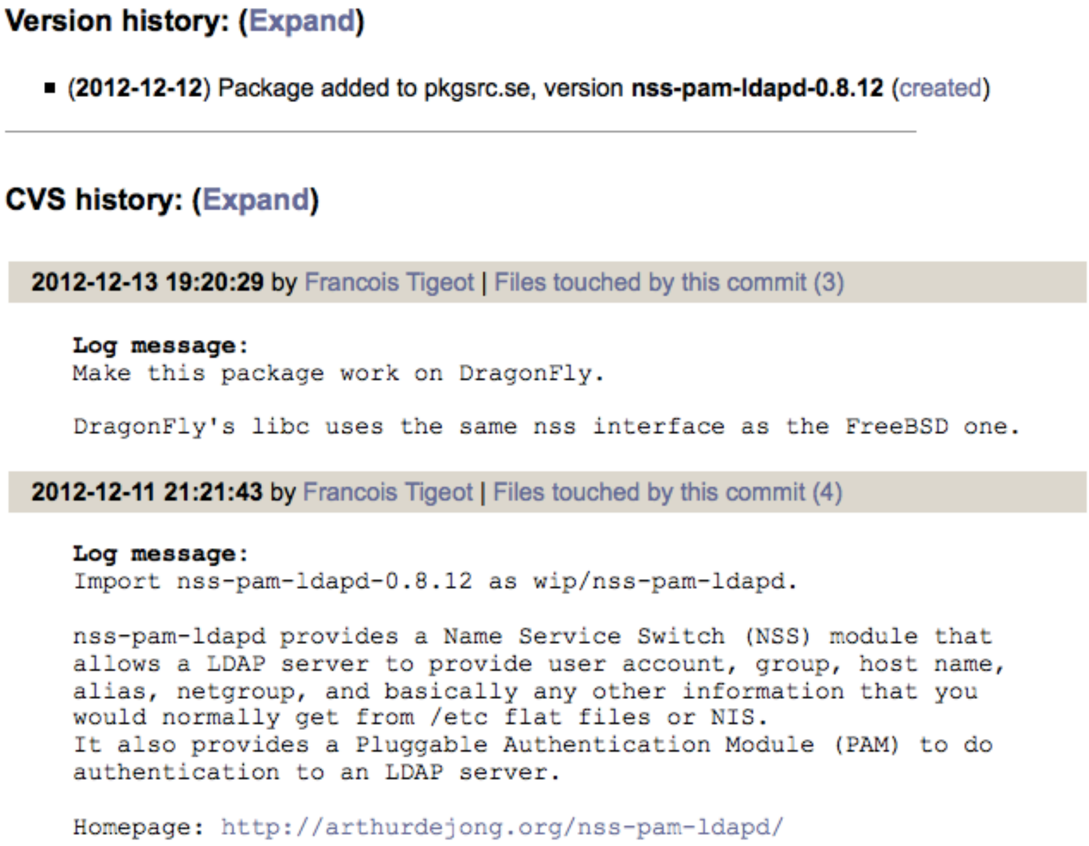
nss-pam-ldapd provides a Name Service Switch (NSS) module that
allows a LDAP server to provide user account, group, host name,
alias, netgroup, and basically any other information that you
would normally get from /etc flat files or NIS.
It also provides a Pluggable Authentication Module (PAM) to do
authentication to an LDAP server.

**Required to build:**
[devel/autoconf]

**Master sites:**

- http://arthurdejong.org/nss-pam-ldapd/ (Download)

**SHA1:** 9c320172df0cdd4eca6cd97ad4c2438e6552ffe0
**RMD160:** 3c550253d122d6934dc2a967cf6d4a7b8f00cb49
**Filesize:** 473.099 KB



**Version history: (Expand)**

- (**2012-12-12**) Package added to pkgsrc.se, version **nss-pam-ldapd-0.8.12** (created)

**CVS history: (Expand)**

**2012-12-13 19:20:29** by Francois Tigeot | Files touched by this commit (3)

```
Log message:
Make this package work on DragonFly.

DragonFly's libc uses the same nss interface as the FreeBSD one.
```

**2012-12-11 21:21:43** by Francois Tigeot | Files touched by this commit (4)

```
Log message:
Import nss-pam-ldapd-0.8.12 as wip/nss-pam-ldapd.

nss-pam-ldapd provides a Name Service Switch (NSS) module that
allows a LDAP server to provide user account, group, host name,
alias, netgroup, and basically any other information that you
would normally get from /etc flat files or NIS.
It also provides a Pluggable Authentication Module (PAM) to do
authentication to an LDAP server.

Homepage: http://arthurdejong.org/nss-pam-ldapd/
```

# 3 I Identification / Authentification : nss_ldap, kerberos

La base n'intègre pas kerberos

➔ pas de support GSSAPI dans le sshd de la base

➔ possibilité de compiler sshd depuis pkgsrc mais version plus ancienne que celle de la base

La version de kerberos par défaut sous pkgsrc est celle de HEIMDAL

Il faut utiliser la version du MIT car tout ne compile pas avec HEIMDAL

➔ /usr/pkg/etc/mk.conf

```
KRB5_DEFAULT=mit-krb5
PKG_OPTIONS.netatalk=kerberos ldap pam
PKG_OPTIONS.samba=ads ldap pam
```

Intégration de kerberos dans la base de DragonflyBSD comme c'est le cas sous OpenBSD ???

# 4 I Userland : pkgsrc / dports

**Objectif: accélérer le portage de dports ➔ nss_ldap, kerberos…**

➔ poudrière http://fossil.etoilebsd.net/poudriere/doc/trunk/doc/index.wiki

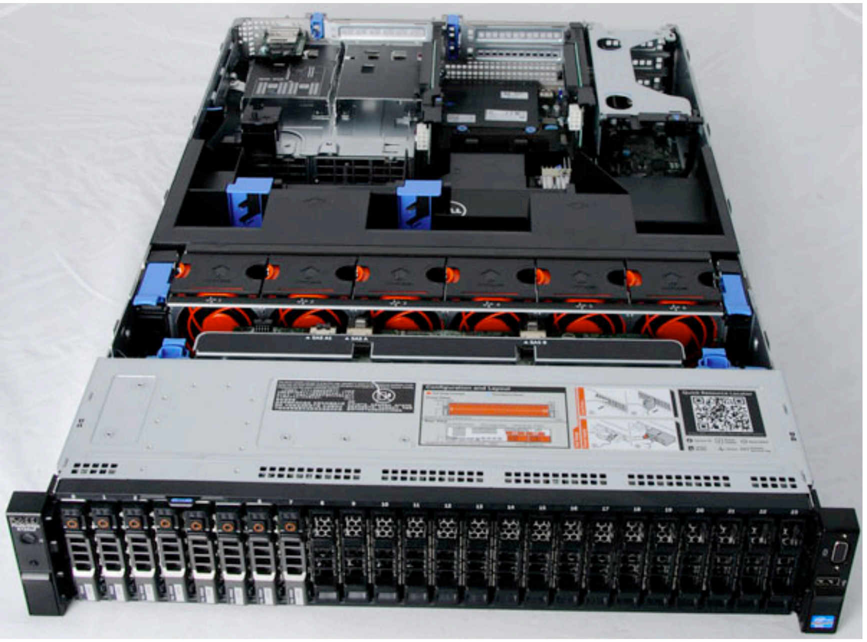
➔ http://www.unix-heaven.org/continuous-package-building-with-poudriere-and-jenkins

1 DELL R720 **prêté par l'IAS** (15/12-15/01)

1 DELL R720xd **prêté par DELL** (15/01-15/02)

- 2 E5-2650 (8C, 16T, 20Mo cache, 8GTs QPI)
- 64Go de RAM
- 12 disques 146GO, SAS 6Gb/s

➔ 32 cpu-threads

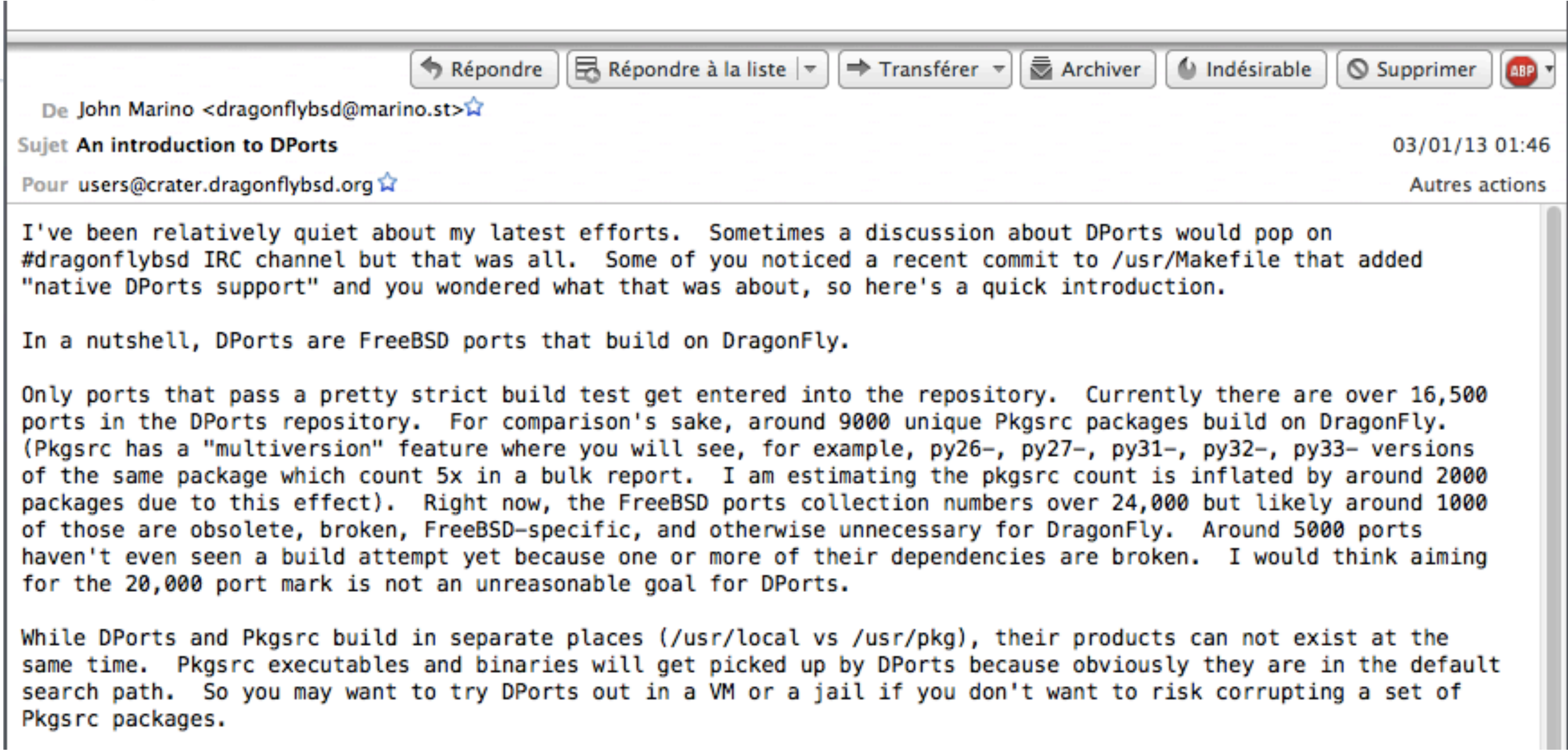


```
Bugs not fixed yet:
-------------------
- ~~tmpfs doesn't release space~~
- ~~dangling vnode: impossible to unmount a filesystem~~
- unsafe vfs quota memory allocation
- ~~Bug #2510: processes stuck in "vfs_busy" state when interrupting a poudriere run~~


Major bugs fixed:
-----------------
kernel - Fix kernel panic caused by rename race
authorMatthew Dillon <dillon@apollo.backplane.com>
Fri, 1 Feb 2013 21:47:37 +0000 (13:47 -0800)

kernel - Fix deadlock when umount races an access on the underlying filesystem
authorMatthew Dillon <dillon@apollo.backplane.com>
Tue, 29 Jan 2013 23:04:05 +0000 (15:04 -0800)

kernel - Attempt to fix NULL pointer dereference during console switch
authorMatthew Dillon <dillon@apollo.backplane.com>
Tue, 29 Jan 2013 19:19:22 +0000 (11:19 -0800)
Bug 2481.

kernel - Fix tty cool-aid
authorMatthew Dillon <dillon@apollo.backplane.com>
Tue, 29 Jan 2013 19:11:49 +0000 (11:11 -0800)

socket: Mark the asynchronous rcvd netmsg dead, when it is dropped
authorSepherosa Ziehau <sephe@dragonflybsd.org>
Tue, 29 Jan 2013 08:29:35 +0000 (16:29 +0800)

kernel - Fix signal FP save/restore issues when AVX is enabled
authorMatthew Dillon <dillon@apollo.backplane.com>
Sun, 13 Jan 2013 00:16:11 +0000 (16:16 -0800)
```

# 4 I Userland : pkgsrc / dports

Au 05-02-2013

- Un peu plus de 18,000 ports ont été compilés au moins une fois sous DragonflyBSD (sur un total d'environ 24,000)



De John Marino <dragonflybsd@marino.st>

Sujet **An introduction to DPorts** 03/01/13 01:46

Pour users@crater.dragonflybsd.org

```
I've been relatively quiet about my latest efforts.  Sometimes a discussion about DPorts would pop on
#dragonflybsd IRC channel but that was all.  Some of you noticed a recent commit to /usr/Makefile that added
"native DPorts support" and you wondered what that was about, so here's a quick introduction.

In a nutshell, DPorts are FreeBSD ports that build on DragonFly.

Only ports that pass a pretty strict build test get entered into the repository.  Currently there are over 16,500
ports in the DPorts repository.  For comparison's sake, around 9000 unique Pkgsrc packages build on DragonFly.
(Pkgsrc has a "multiversion" feature where you will see, for example, py26-, py27-, py31-, py32-, py33- versions
of the same package which count 5x in a bulk report.  I am estimating the pkgsrc count is inflated by around 2000
packages due to this effect).  Right now, the FreeBSD ports collection numbers over 24,000 but likely around 1000
of those are obsolete, broken, FreeBSD-specific, and otherwise unnecessary for DragonFly.  Around 5000 ports
haven't even seen a build attempt yet because one or more of their dependencies are broken.  I would think aiming
for the 20,000 port mark is not an unreasonable goal for DPorts.

While DPorts and Pkgsrc build in separate places (/usr/local vs /usr/pkg), their products can not exist at the
same time.  Pkgsrc executables and binaries will get picked up by DPorts because obviously they are in the default
search path.  So you may want to try DPorts out in a VM or a jail if you don't want to risk corrupting a set of
Pkgsrc packages.
```

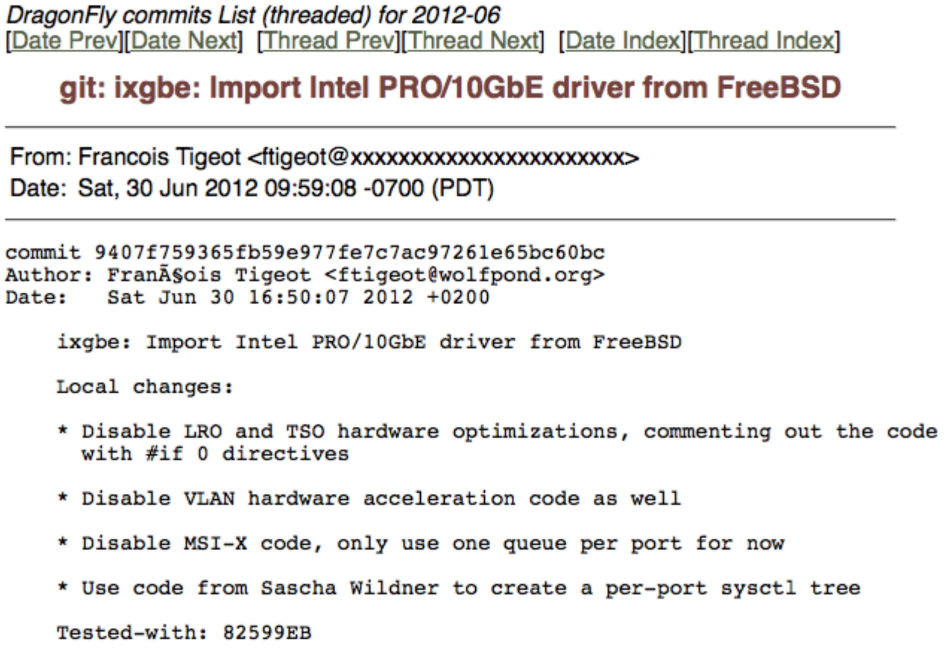
# 5 I Performances : 10Gbe, scheduler & swapcache, tmpfs

**Objectif : portage du pilote 10Gb/s « ixgbe »**

2 DELL R410 du CMLS et du CPHT (juin-octobre 2012)

- 2 X5650 @ 2,7Ghz (6C, 12T, 12Mo, 6,4GT/s QPI)
- 24 Go RAM
- 1 carte réseau INTEL X-520 10Gb/s Cuivre

DragonFlyBSD

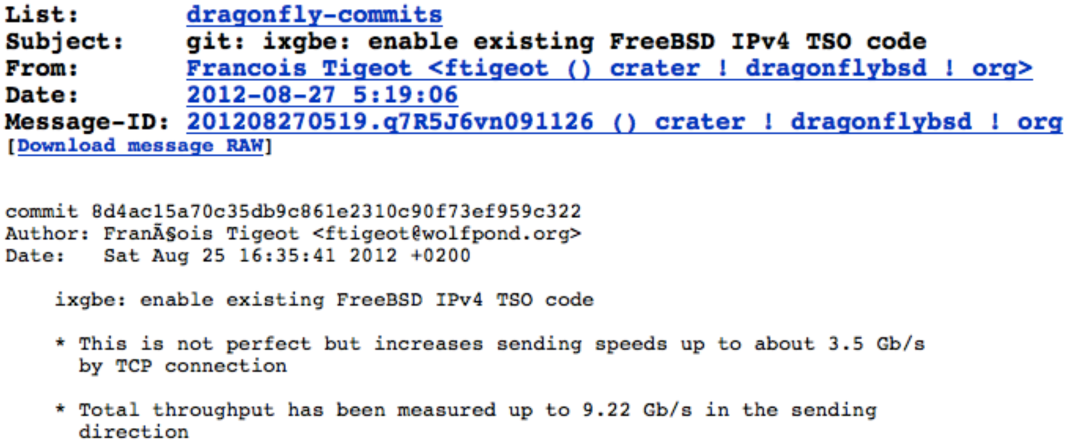


*DragonFly commits List (threaded) for 2012-06*
[Date Prev][Date Next] [Thread Prev][Thread Next] [Date Index][Thread Index]

**git: ixgbe: Import Intel PRO/10GbE driver from FreeBSD**

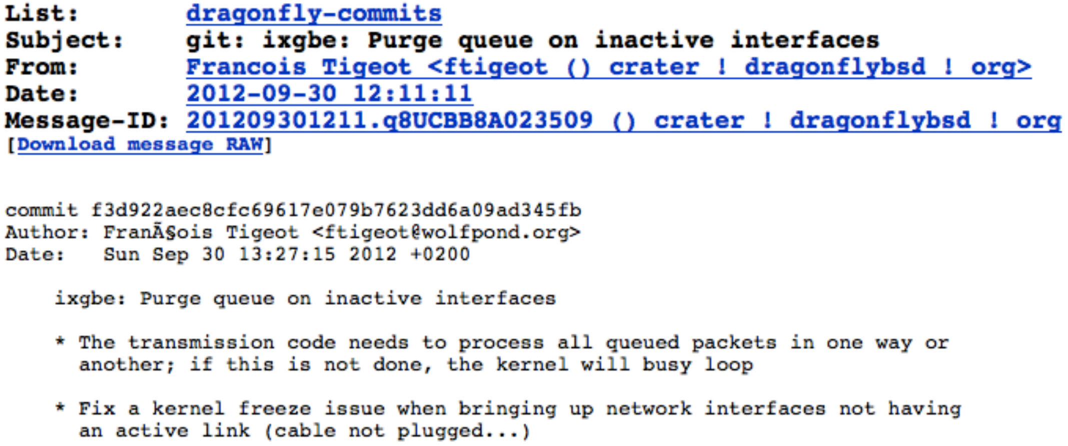
From: Francois Tigeot <ftigeot@xxxxxxxxxxxxxxxxxxxxxx>
Date: Sat, 30 Jun 2012 09:59:08 -0700 (PDT)

```
commit 9407f759365fb59e977fe7c7ac97261e65bc60bc
Author: FranÃ§ois Tigeot <ftigeot@wolfpond.org>
Date:   Sat Jun 30 16:50:07 2012 +0200

    ixgbe: Import Intel PRO/10GbE driver from FreeBSD

    Local changes:

    * Disable LRO and TSO hardware optimizations, commenting out the code
      with #if 0 directives

    * Disable VLAN hardware acceleration code as well

    * Disable MSI-X code, only use one queue per port for now

    * Use code from Sascha Wildner to create a per-port sysctl tree

    Tested-with: 82599EB
```



**List:** dragonfly-commits
**Subject:** git: ixgbe: enable existing FreeBSD IPv4 TSO code
**From:** Francois Tigeot <ftigeot () crater ! dragonflybsd ! org>
**Date:** 2012-08-27 5:19:06
**Message-ID:** 201208270519.q7R5J6vn091126 () crater ! dragonflybsd ! org
[Download message RAW]

```
commit 8d4ac15a70c35db9c861e2310c90f73ef959c322
Author: FranÃ§ois Tigeot <ftigeot@wolfpond.org>
Date:   Sat Aug 25 16:35:41 2012 +0200

    ixgbe: enable existing FreeBSD IPv4 TSO code

    * This is not perfect but increases sending speeds up to about 3.5 Gb/s
      by TCP connection

    * Total throughput has been measured up to 9.22 Gb/s in the sending
      direction
```

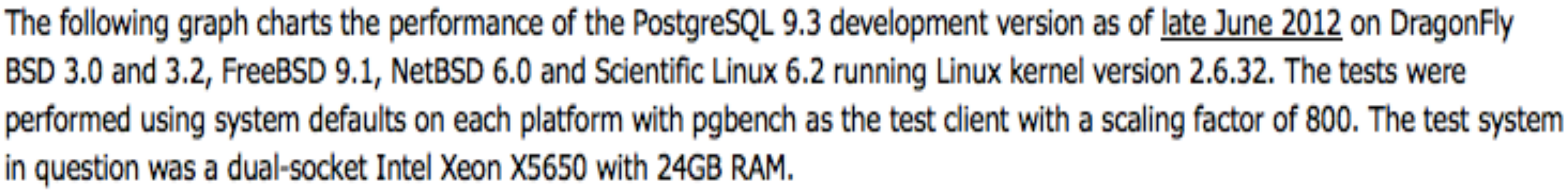


**List:** dragonfly-commits
**Subject:** git: ixgbe: Purge queue on inactive interfaces
**From:** Francois Tigeot <ftigeot () crater ! dragonflybsd ! org>
**Date:** 2012-09-30 12:11:11
**Message-ID:** 201209301211.q8UCBB8A023509 () crater ! dragonflybsd ! org
[Download message RAW]

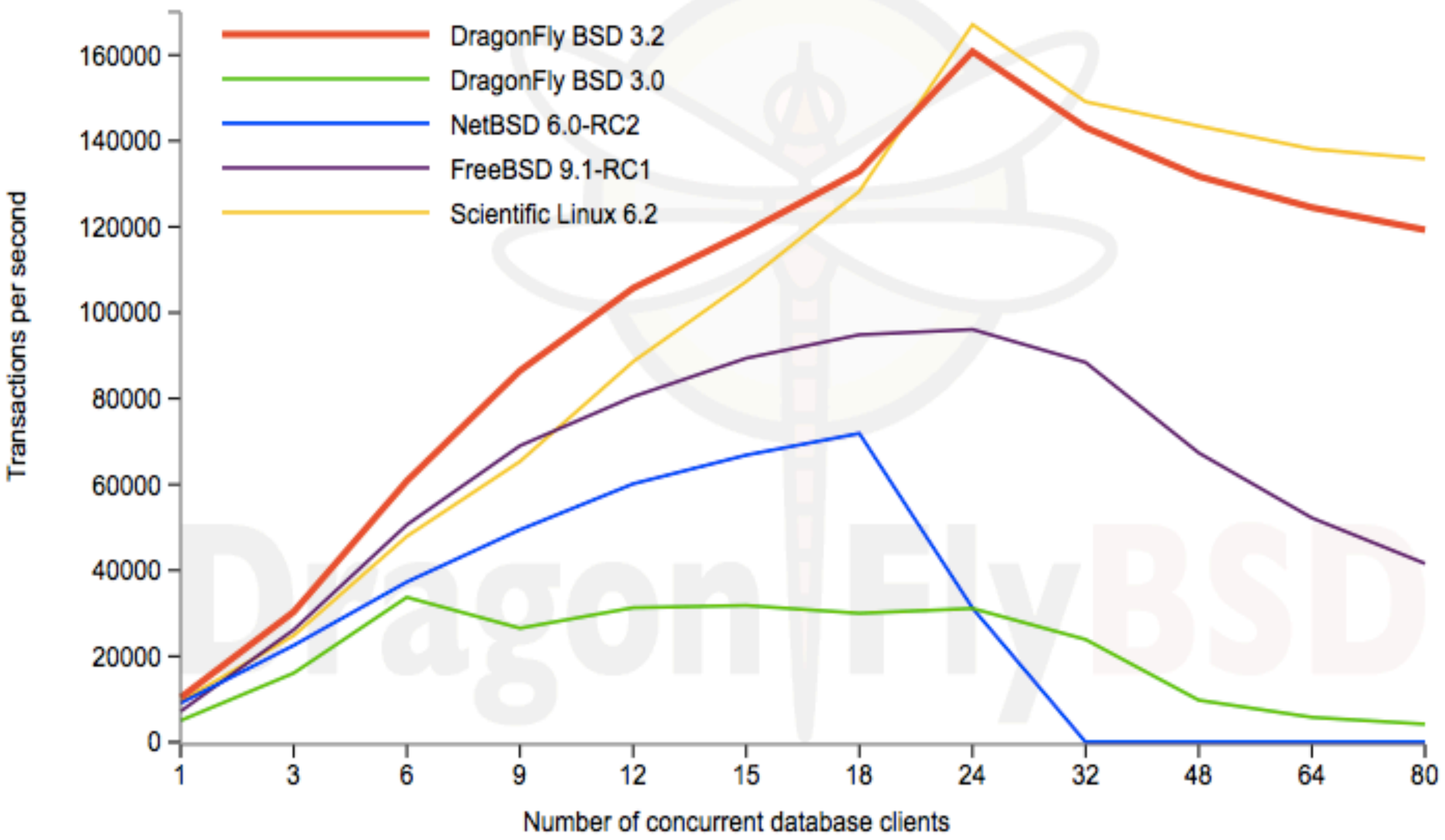
```
commit f3d922aec8cfc69617e079b7623dd6a09ad345fb
Author: FranÃ§ois Tigeot <ftigeot@wolfpond.org>
Date:   Sun Sep 30 13:27:15 2012 +0200

    ixgbe: Purge queue on inactive interfaces

    * The transmission code needs to process all queued packets in one way or
      another; if this is not done, the kernel will busy loop

    * Fix a kernel freeze issue when bringing up network interfaces not having
      an active link (cable not plugged...)
```

# 5 I Performances : 10Gbe, scheduler & swapcache, tmpfs

cf http://www.dragonflybsd.org/performance/

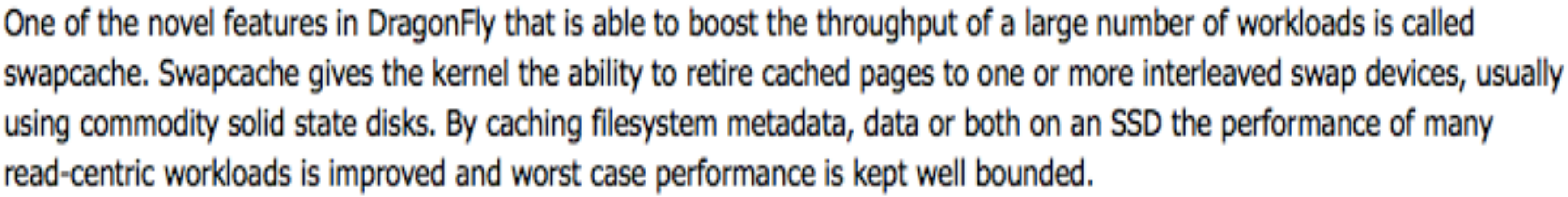


The following graph charts the performance of the PostgreSQL 9.3 development version as of late June 2012 on DragonFly BSD 3.0 and 3.2, FreeBSD 9.1, NetBSD 6.0 and Scientific Linux 6.2 running Linux kernel version 2.6.32. The tests were performed using system defaults on each platform with pgbench as the test client with a scaling factor of 800. The test system in question was a dual-socket Intel Xeon X5650 with 24GB RAM.

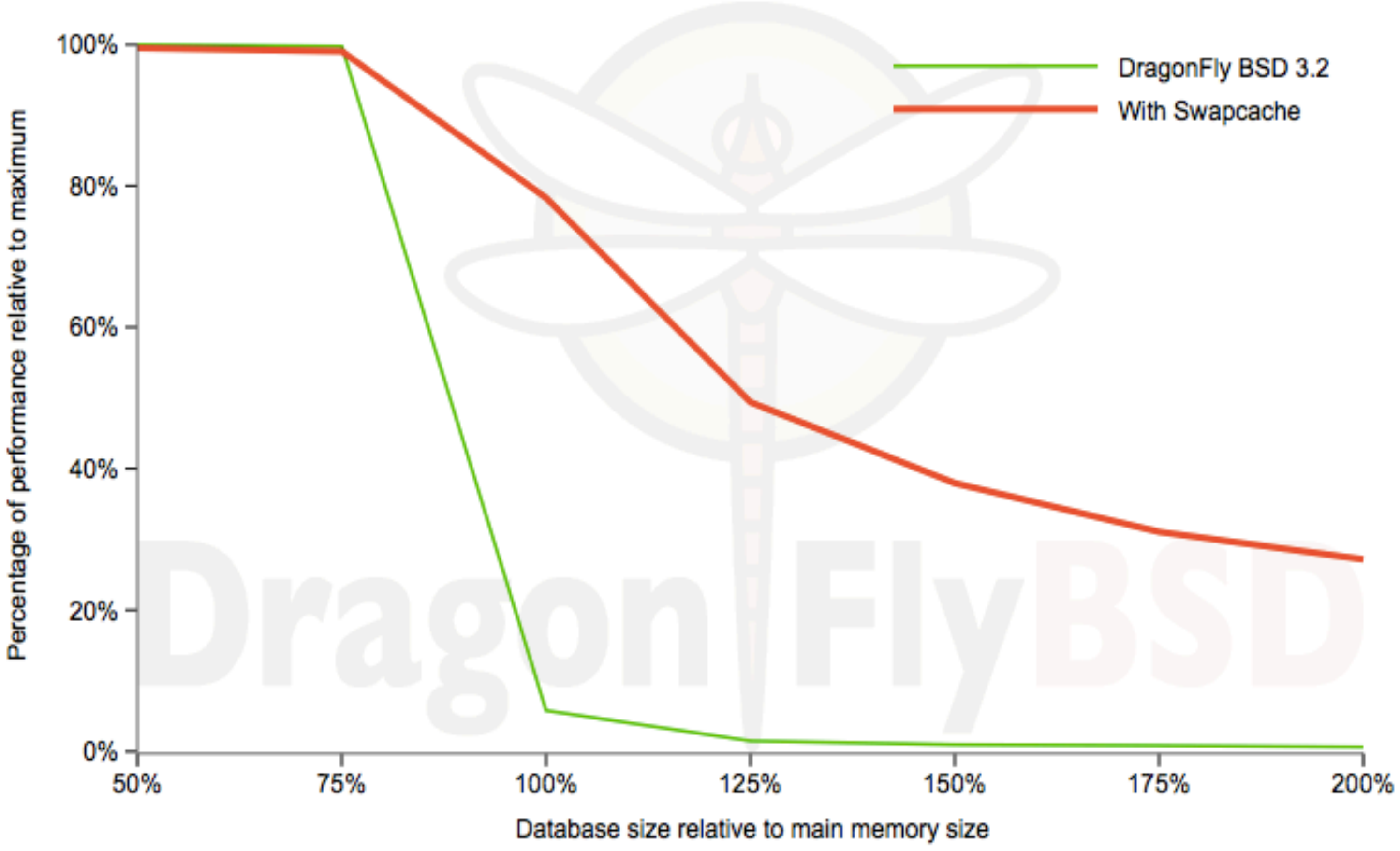


# 5 I Performances : 10Gbe, scheduler & swapcache, tmpfs

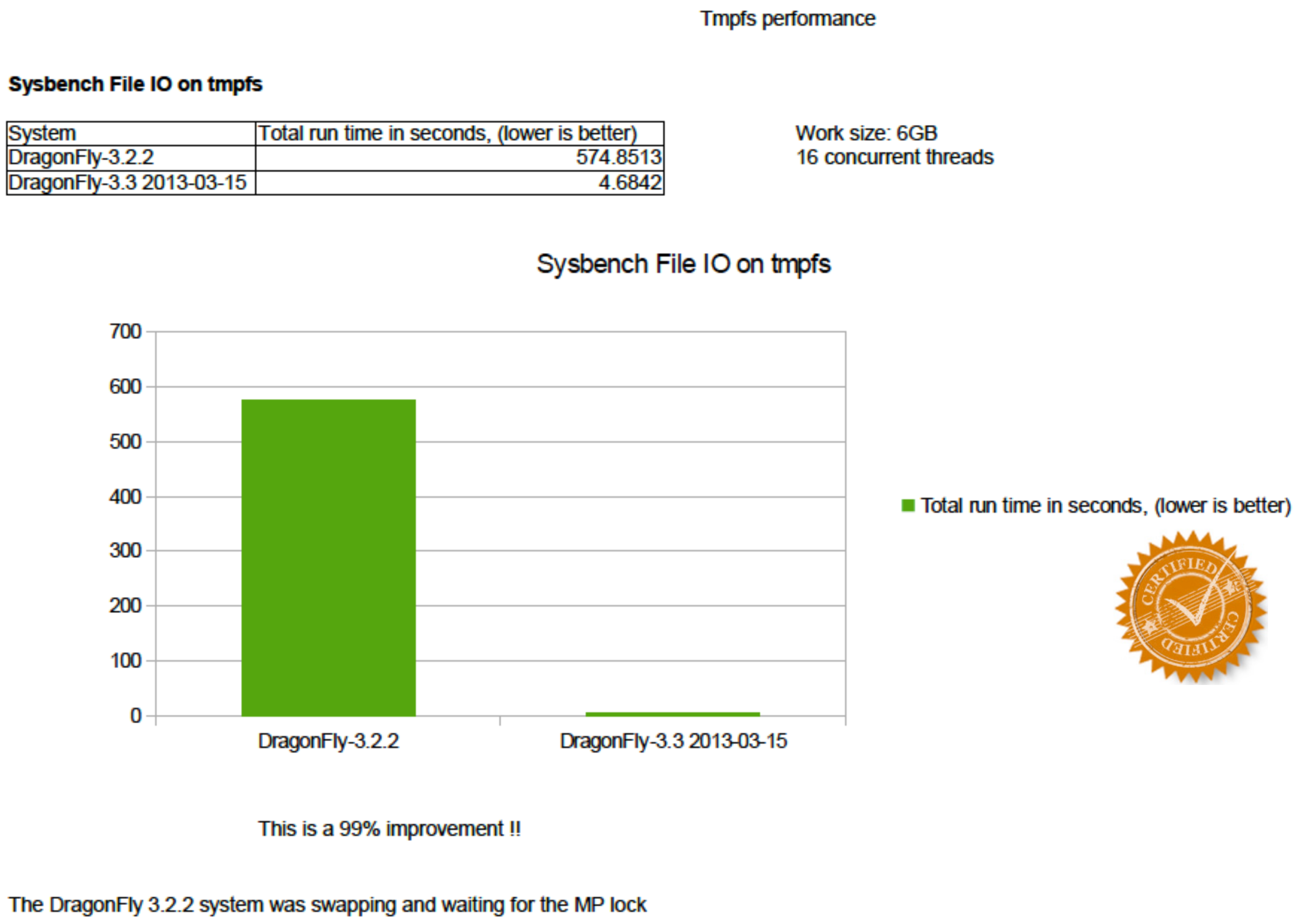
cf http://www.dragonflybsd.org/performance/



One of the novel features in DragonFly that is able to boost the throughput of a large number of workloads is called swapcache. Swapcache gives the kernel the ability to retire cached pages to one or more interleaved swap devices, usually using commodity solid state disks. By caching filesystem metadata, data or both on an SSD the performance of many read-centric workloads is improved and worst case performance is kept well bounded.



# 5 I Performances : 10Gbe, scheduler & swapcache, tmpfs



Tmpfs performance

**Sysbench File IO on tmpfs**

| System | Total run time in seconds, (lower is better) |
|---|---:|
| DragonFly-3.2.2 | 574.8513 |
| DragonFly-3.3 2013-03-15 | 4.6842 |

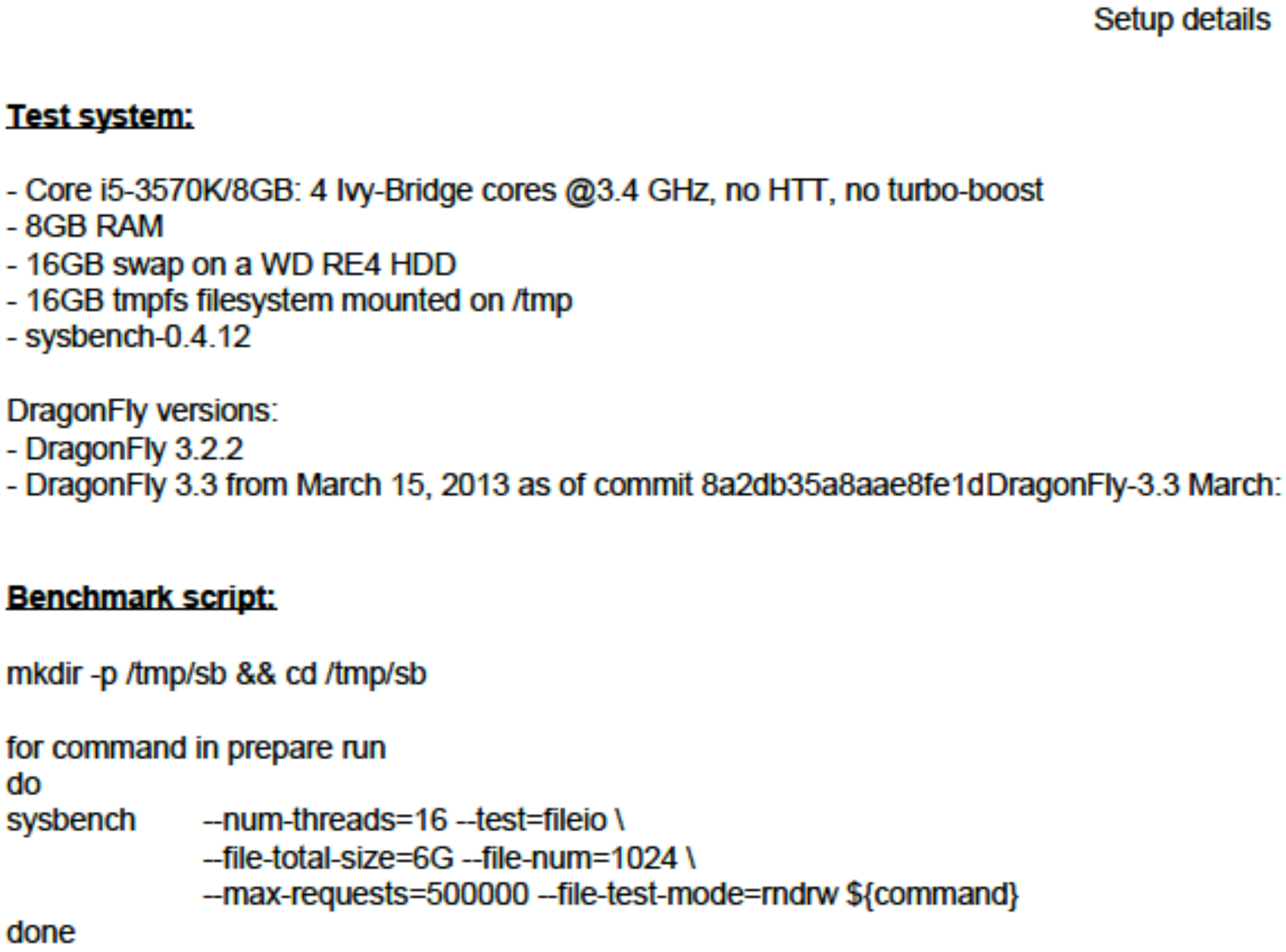
Work size: 6GB
16 concurrent threads

Sysbench File IO on tmpfs

This is a 99% improvement !!

The DragonFly 3.2.2 system was swapping and waiting for the MP lock

# 5 | Performances : 10Gbe, scheduler & swapcache, tmpfs



Setup details

**Test system:**

- Core i5-3570K/8GB: 4 Ivy-Bridge cores @3.4 GHz, no HTT, no turbo-boost
- 8GB RAM
- 16GB swap on a WD RE4 HDD
- 16GB tmpfs filesystem mounted on /tmp
- sysbench-0.4.12

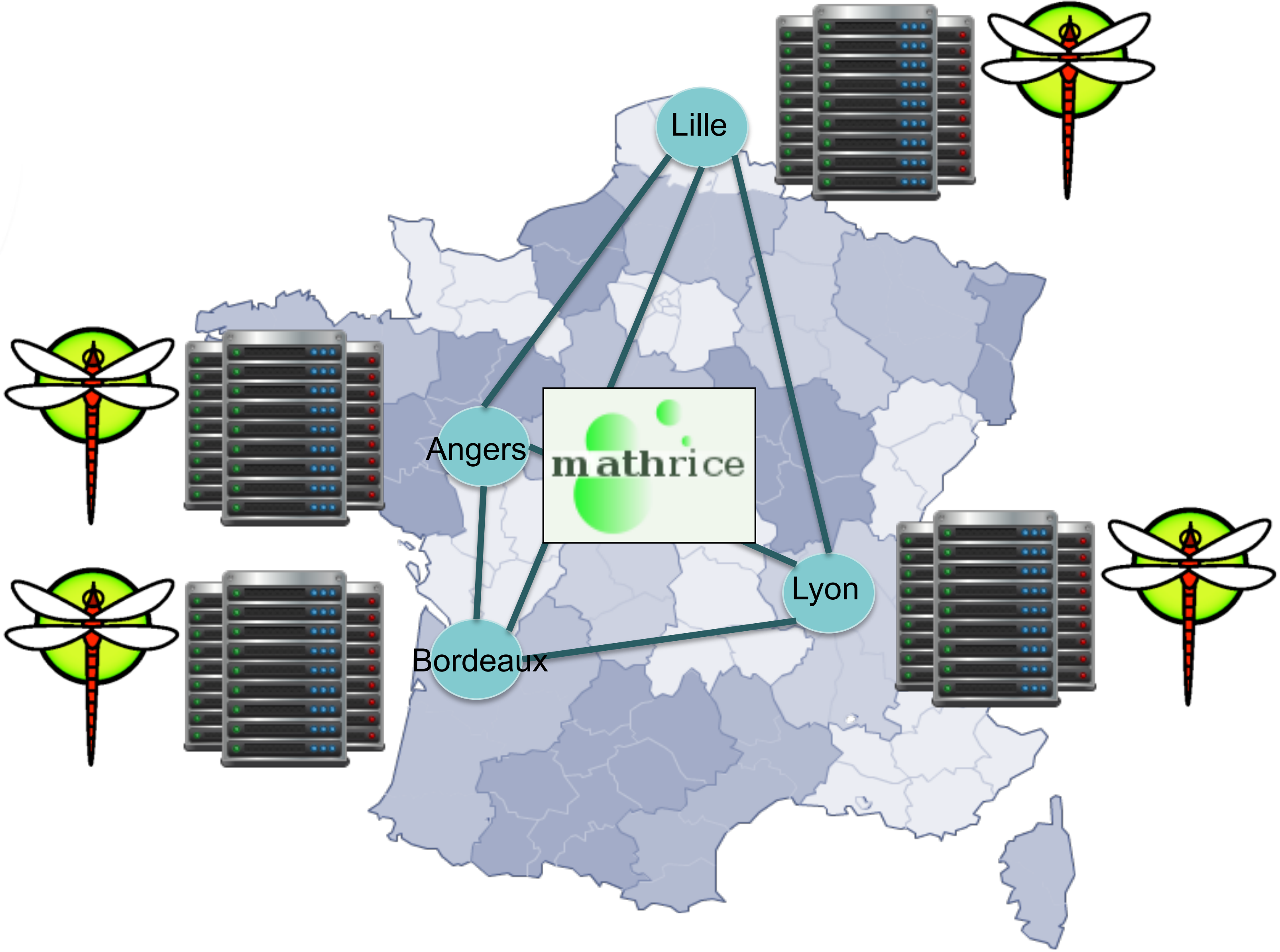
DragonFly versions:

- DragonFly 3.2.2
- DragonFly 3.3 from March 15, 2013 as of commit 8a2db35a8aae8fe1dDragonFly-3.3 March:

**Benchmark script:**

```
mkdir -p /tmp/sb && cd /tmp/sb

for command in prepare run
do
sysbench        --num-threads=16 --test=fileio \
                --file-total-size=6G --file-num=1024 \
                --max-requests=500000 --file-test-mode=rndrw ${command}
done
```

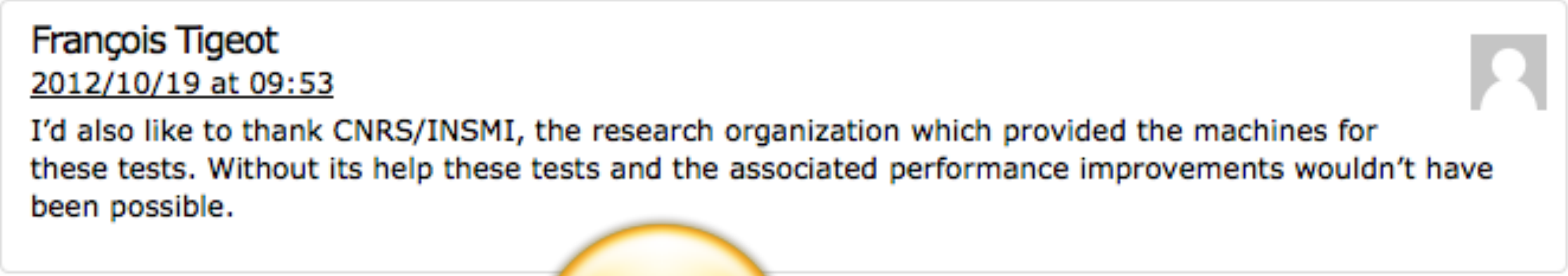
# 6 I Conclusion : perspectives



Hammer & Hammer2

# 6 I Conclusion : perspectives

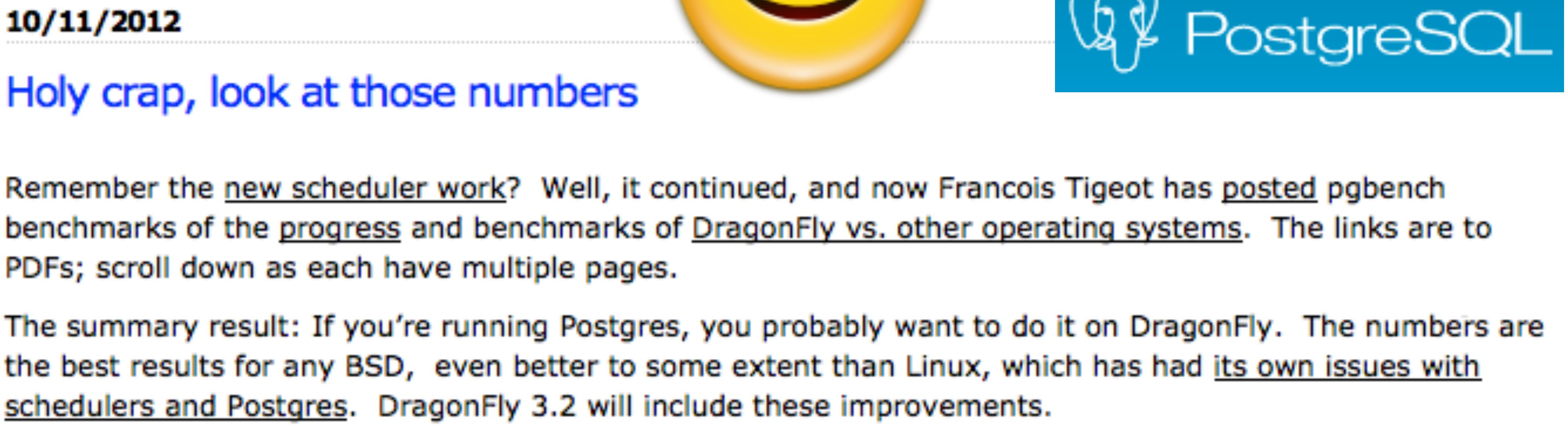
http://www.shiningsilence.com/dbsdlog/2012/10/11/10544.html



François Tigeot

2012/10/19 at 09:53

I'd also like to thank CNRS/INSMI, the research organization which provided the machines for these tests. Without its help these tests and the associated performance improvements wouldn't have been possible.



**10/11/2012**

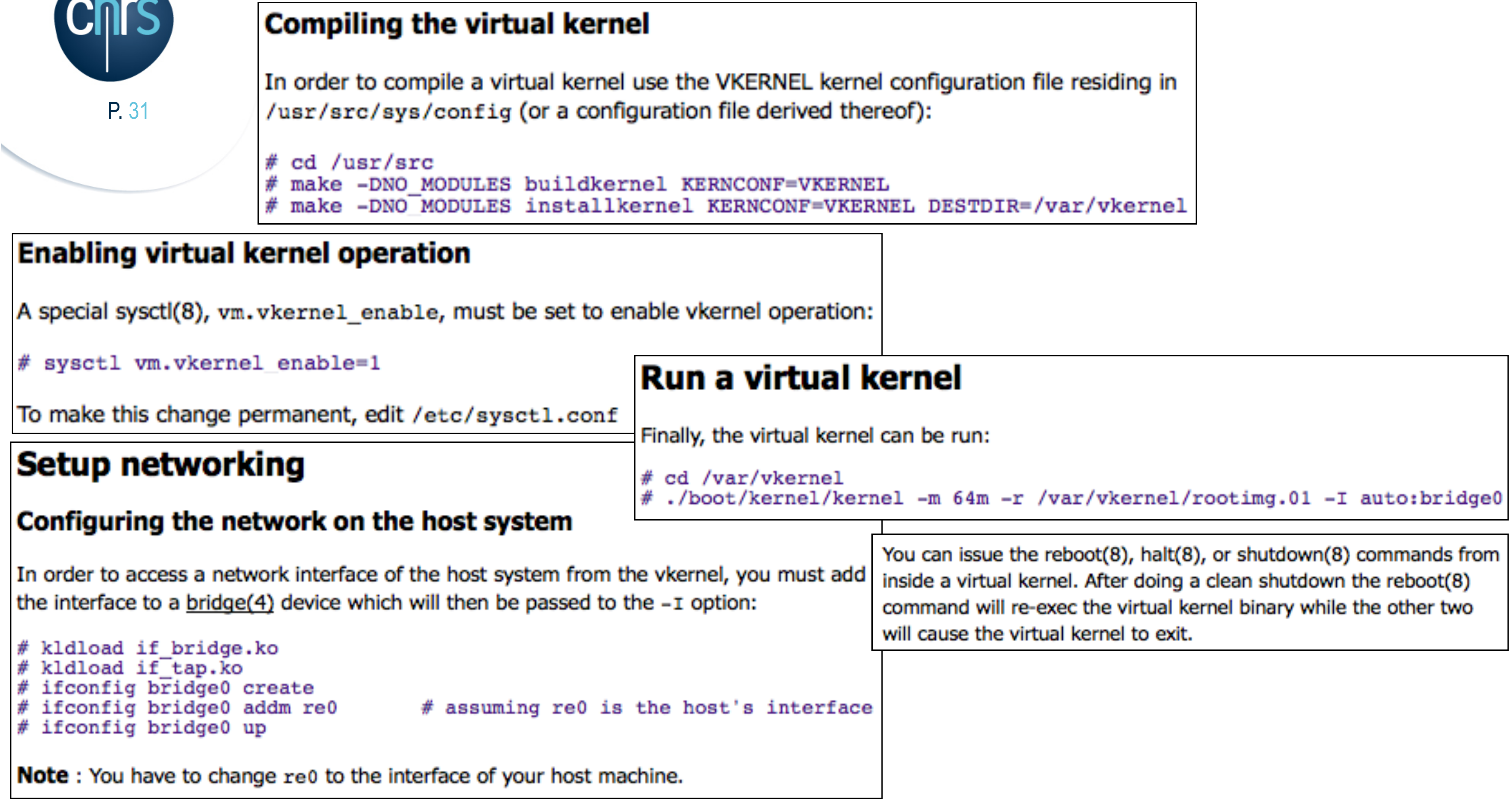
Holy crap, look at those numbers

Remember the new scheduler work? Well, it continued, and now Francois Tigeot has posted pgbench benchmarks of the progress and benchmarks of DragonFly vs. other operating systems. The links are to PDFs; scroll down as each have multiple pages.

The summary result: If you're running Postgres, you probably want to do it on DragonFly. The numbers are the best results for any BSD, even better to some extent than Linux, which has had its own issues with schedulers and Postgres. DragonFly 3.2 will include these improvements.

# 6 I Conclusion : perspectives

http://www.dragonflybsd.org/docs/newhandbook/vkernel/



### Compiling the virtual kernel

In order to compile a virtual kernel use the VKERNEL kernel configuration file residing in `/usr/src/sys/config` (or a configuration file derived thereof):

```
# cd /usr/src
# make -DNO_MODULES buildkernel KERNCONF=VKERNEL
# make -DNO_MODULES installkernel KERNCONF=VKERNEL DESTDIR=/var/vkernel
```

### Enabling virtual kernel operation

A special sysctl(8), `vm.vkernel_enable`, must be set to enable vkernel operation:

```
# sysctl vm.vkernel_enable=1
```

To make this change permanent, edit `/etc/sysctl.conf`

### Setup networking

#### Configuring the network on the host system

In order to access a network interface of the host system from the vkernel, you must add the interface to a bridge(4) device which will then be passed to the `-I` option:

```
# kldload if_bridge.ko
# kldload if_tap.ko
# ifconfig bridge0 create
# ifconfig bridge0 addm re0       # assuming re0 is the host's interface
# ifconfig bridge0 up
```

**Note** : You have to change `re0` to the interface of your host machine.

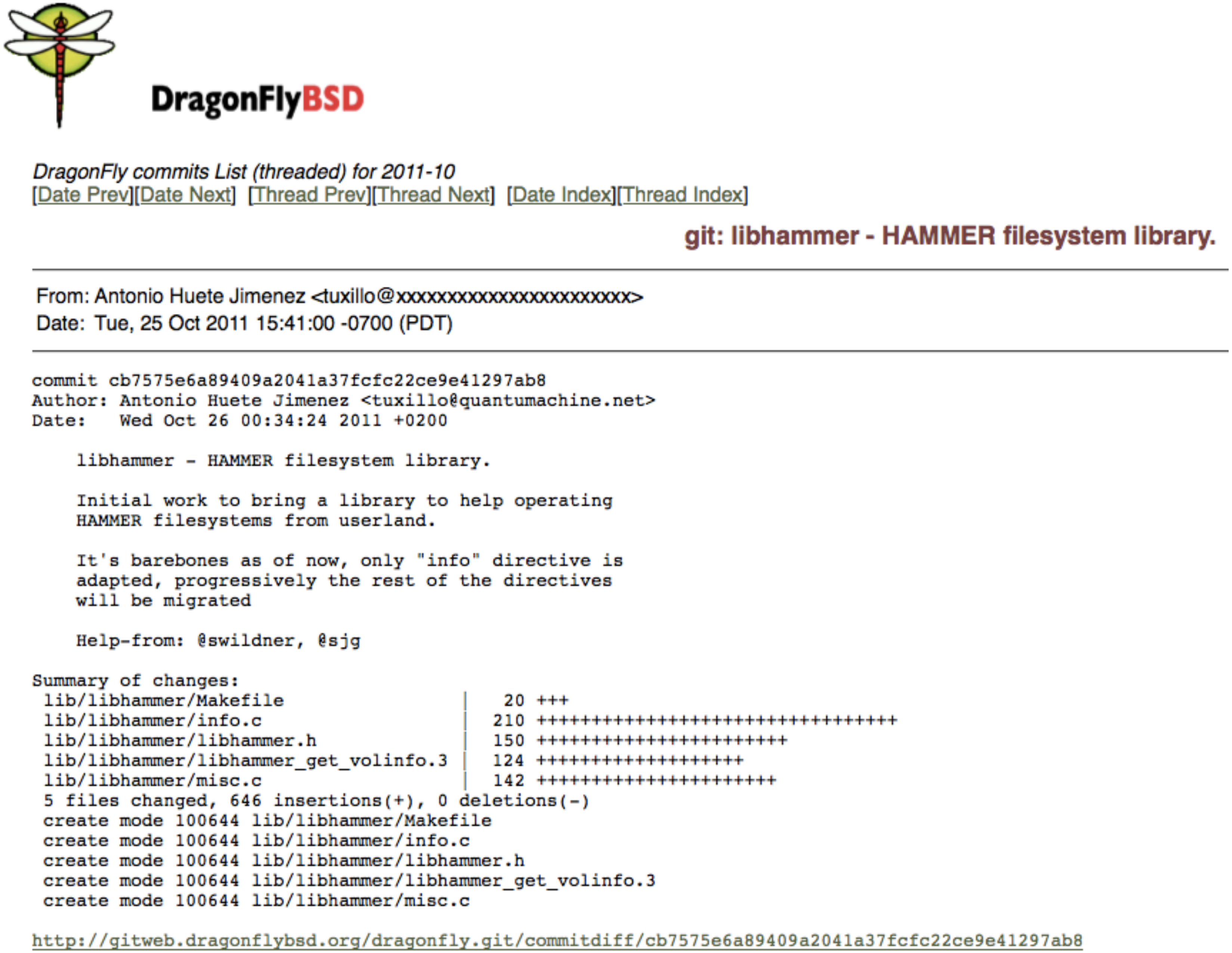
### Run a virtual kernel

Finally, the virtual kernel can be run:

```
# cd /var/vkernel
# ./boot/kernel/kernel -m 64m -r /var/vkernel/rootimg.01 -I auto:bridge0
```

You can issue the reboot(8), halt(8), or shutdown(8) commands from inside a virtual kernel. After doing a clean shutdown the reboot(8) command will re-exec the virtual kernel binary while the other two will cause the virtual kernel to exit.

# 6 I Conclusion : perspectives



DragonFlyBSD

*DragonFly commits List (threaded) for 2011-10*
[Date Prev][Date Next] [Thread Prev][Thread Next] [Date Index][Thread Index]

**git: libhammer - HAMMER filesystem library.**

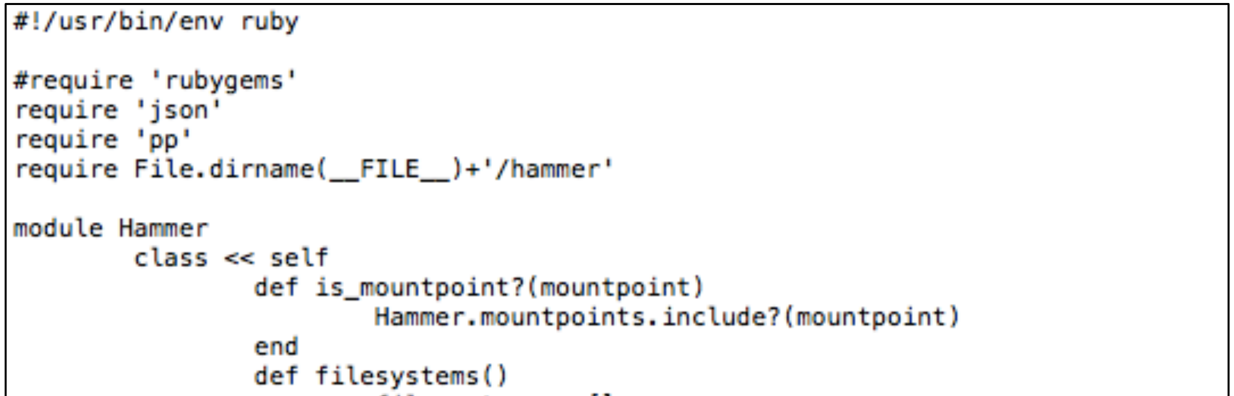
From: Antonio Huete Jimenez <tuxillo@xxxxxxxxxxxxxxxxxxxxxxxx>
Date: Tue, 25 Oct 2011 15:41:00 -0700 (PDT)

```
commit cb7575e6a89409a2041a37fcfc22ce9e41297ab8
Author: Antonio Huete Jimenez <tuxillo@quantumachine.net>
Date:   Wed Oct 26 00:34:24 2011 +0200

    libhammer - HAMMER filesystem library.

    Initial work to bring a library to help operating
    HAMMER filesystems from userland.

    It's barebones as of now, only "info" directive is
    adapted, progressively the rest of the directives
    will be migrated

    Help-from: @swildner, @sjg

Summary of changes:
 lib/libhammer/Makefile                 |  20 +++
 lib/libhammer/info.c                   | 210 +++++++++++++++++++++++++++++++++
 lib/libhammer/libhammer.h              | 150 ++++++++++++++++++++++
 lib/libhammer/libhammer_get_volinfo.3  | 124 +++++++++++++++++++
 lib/libhammer/misc.c                   | 142 +++++++++++++++++++++
 5 files changed, 646 insertions(+), 0 deletions(-)
 create mode 100644 lib/libhammer/Makefile
 create mode 100644 lib/libhammer/info.c
 create mode 100644 lib/libhammer/libhammer.h
 create mode 100644 lib/libhammer/libhammer_get_volinfo.3
 create mode 100644 lib/libhammer/misc.c
```

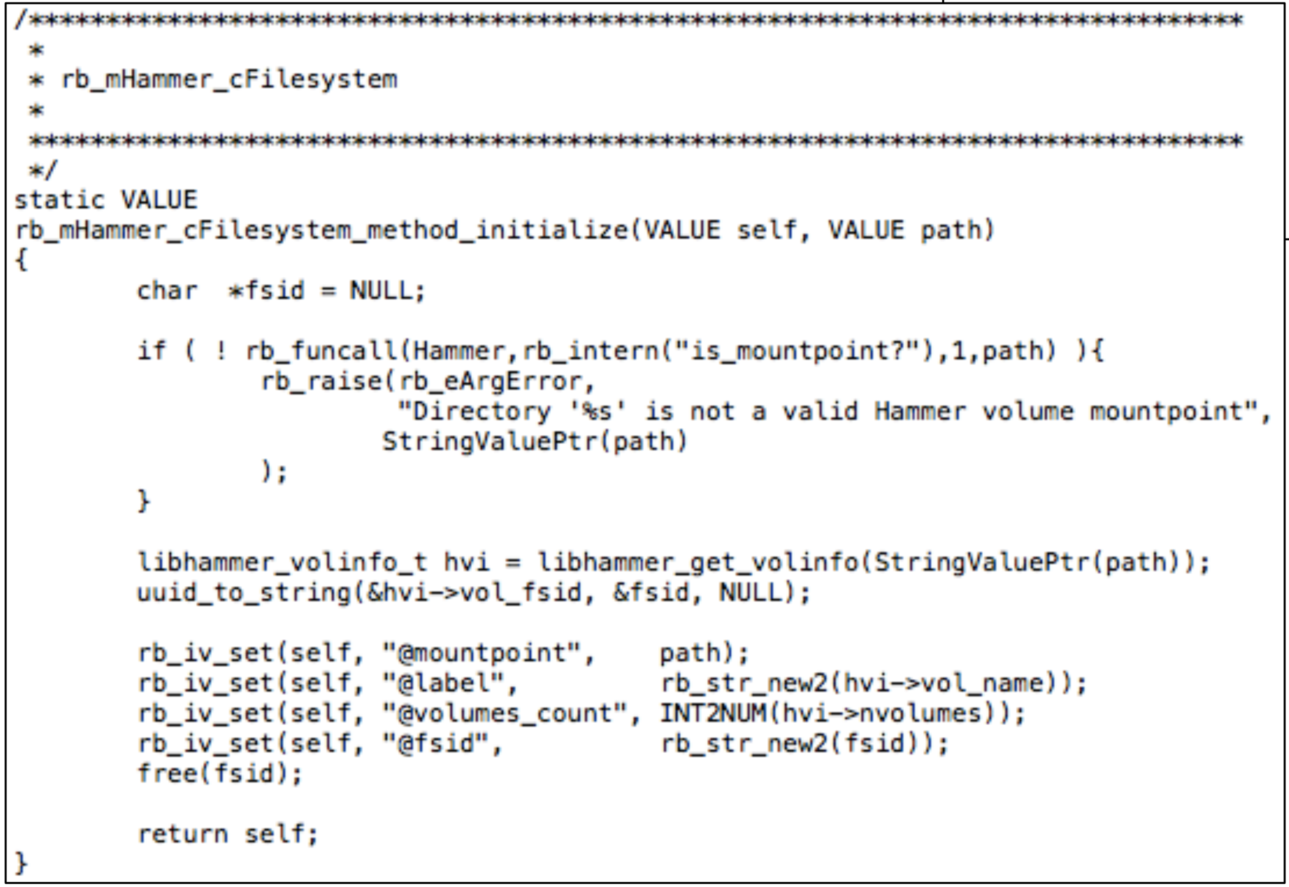
http://gitweb.dragonflybsd.org/dragonfly.git/commitdiff/cb7575e6a89409a2041a37fcfc22ce9e41297ab8

# 6 I Conclusion : perspectives

- Ruby / C Extensions
- Web Services d'accès aux ressources DragonflyBSD



```
#!/usr/bin/env ruby

#require 'rubygems'
require 'json'
require 'pp'
require File.dirname(__FILE__)+'/hammer'

module Hammer
        class << self
                def is_mountpoint?(mountpoint)
                        Hammer.mountpoints.include?(mountpoint)
                end
                def filesystems()
                        filesystems = []
                        Hammer.mountpoints do |mountpoint|
                                filesystem  = Hammer::Filesystem.new(mountpoint)
                                filesystems << filesystem
                                yield filesystem if block_given?
                        end
                        return filesystems
```



```
/*******************************************************************************
 *
 * rb_mHammer_cFilesystem
 *
 *******************************************************************************
 */
static VALUE
rb_mHammer_cFilesystem_method_initialize(VALUE self, VALUE path)
{
        char  *fsid = NULL;

        if ( ! rb_funcall(Hammer,rb_intern("is_mountpoint?"),1,path) ){
                rb_raise(rb_eArgError,
                         "Directory '%s' is not a valid Hammer volume mountpoint",
                        StringValuePtr(path)
                );
        }

        libhammer_volinfo_t hvi = libhammer_get_volinfo(StringValuePtr(path));
        uuid_to_string(&hvi->vol_fsid, &fsid, NULL);

        rb_iv_set(self, "@mountpoint",    path);
        rb_iv_set(self, "@label",         rb_str_new2(hvi->vol_name));
        rb_iv_set(self, "@volumes_count", INT2NUM(hvi->nvolumes));
        rb_iv_set(self, "@fsid",          rb_str_new2(fsid));
        free(fsid);

        return self;
}
```

- WEBUI ExtJS & Sencha Touch
- Présentation et Gestion d'un ensemble de serveurs

&

# Remerciements

**<u>Poudrière & pkgng</u>**

Baptiste Daroussin

FreeBSD

&

**<u>DragonflyBSD</u>**

Matthew Dillon
François Tigeot
John Marino
Sepherosa Ziehau
Venkatesh Srinivas
Sacha Wildner

. . .

Thierry Favereau
Thierry Thiesson

DELL

Stéphane Caminade
Daniel Altenburger

IAS
Institut d'Astrophysique Spatiale
Orsay

# Références

http://www.ntecs.de/sysarch09/HAMMER.pdf

http://leaf.dragonflybsd.org/cgi/web-man?

http://dl.wolfpond.org/Hammer-Presentation.en.pdf

http://www.dragonflybsd.org/hammer/hammer.pdf