

Montage de partitions agrégées sur des systèmes FreeBSD

par [Nicolas Vallée](#)

Date de publication : 05/06/2006

Dernière mise à jour : 31/10/2006

Avoir un système de partitions complexes pour simplifier l'administration système...

I - Introduction

Avant-Propos

Pourquoi vouloir agréger des partitions ?

Organisation des disques sous BSD

Divers

I-1 - Présentation

I-1.1 - GVinum

I-1.2 - Concaténation (JBOD)

I-1.3 - Raid-0 (Striping)

I-1.4 - Raid-1 (Mirroring)

I-1.5 - Raid-5

I-1.6 - Divers

II - Mise en place

II-1 - Installation de GVinum

II-2 - Préparation d'un disque

II-3 - Configuration de GVinum

II-4 - Lancement automatique de GVinum

II-5 - Montage automatisé

III - Divers

III-1 - Compléments

III-1.1 - Tester sa configuration

III-1.2 - Sauvegarder la configuration

III-1.3 - Ajouter un disque sur un volume concaténé

III-1.4 - Changer un disque défectueux sur une configuration Raid-1

III-1.5 - Changer un disque défectueux sur une configuration Raid-5

III-2 - Liens utiles

Conclusion

I - Introduction

Avant-Propos

Pourquoi vouloir agréger des partitions ?

- pour bénéficier de **disques virtuels très grands**

Les disques deviennent de plus en plus gros, mais tout comme les besoins en stockage.

Vous vous apercevrez souvent que vous avez besoin d'un système de fichiers plus grand que les disques que vous avez à votre disposition.

- pour créer de la redondance, ce qui permettra de **restaurer les données** en cas de défaillance matérielle
- pour améliorer les **performances**

En multipliant le nombre d'accès concurrent, on réduit considérablement les performances globales du système... Il devient alors très pratique de répartir la charge sur différents disques.

Organisation des disques sous BSD

Sur une machine BSD, les disques sont placés dans `/dev` avec des points de montage du type `/dev/adX` pour les disques ide, et `/dev/daX` pour les disques scsi. Sur ces disques, on crée des partitions avec les conventions suivantes :

| Partition | Convention |
|-----------|--|
| a | Contient normalement le système de fichiers racine |
| b | Contient normalement l'espace de pagination |
| c | Normalement de la même taille que la tranche <i>slice</i> contenant les partitions. Cela permet aux utilitaires devant agir sur l'intégralité de la tranche (par exemple un analyseur de blocs défectueux) de travailler sur la partition c. Vous ne devriez normalement pas créer de système de fichiers sur cette partition. |
| d | La partition d a eu dans le passé une signification spécifique, c'est terminé maintenant. A ce jour, quelques outils peuvent fonctionner curieusement si on leur dit de travailler sur la partition d, aussi on ne créera normalement pas de partition d. |

Chaque partition contenant un système de fichiers est stockée dans ce qu'on appelle une tranche, ou *slice*, numérotées de 1 à 4. Les numéros de tranche suivent le nom du périphérique, avec le préfixe s.

Il ne peut y avoir que quatre tranches physiques sur un disque, mais vous pouvez avoir des tranches logiques dans des tranches physiques, numérotées à partir de 5. Elles sont utilisées par des systèmes de fichiers qui s'attendent à occuper une tranche entière.

Les tranches, les disques "en mode dédié", et les autres disques contiennent des partitions, qui sont représentées

par des lettres allant de a à h. Cette lettre est ajoutée au nom de périphérique.

En conclusion chaque disque présent sur le système est identifié. Le nom d'un disque commence par un code qui indique le type de disque, suivi d'un nombre, indiquant de quel disque il s'agit. Contrairement aux tranches, la numérotation des disques commence à 0.


Divers

Ce tutoriel va décrire cette opération pour des systèmes BSD, bénéficiant de **gvinum**

 *En ce qui concerne Linux, il existe LVM décrit par Sylvain Luce dans son [tutorial](#).*

I-1 - Présentation

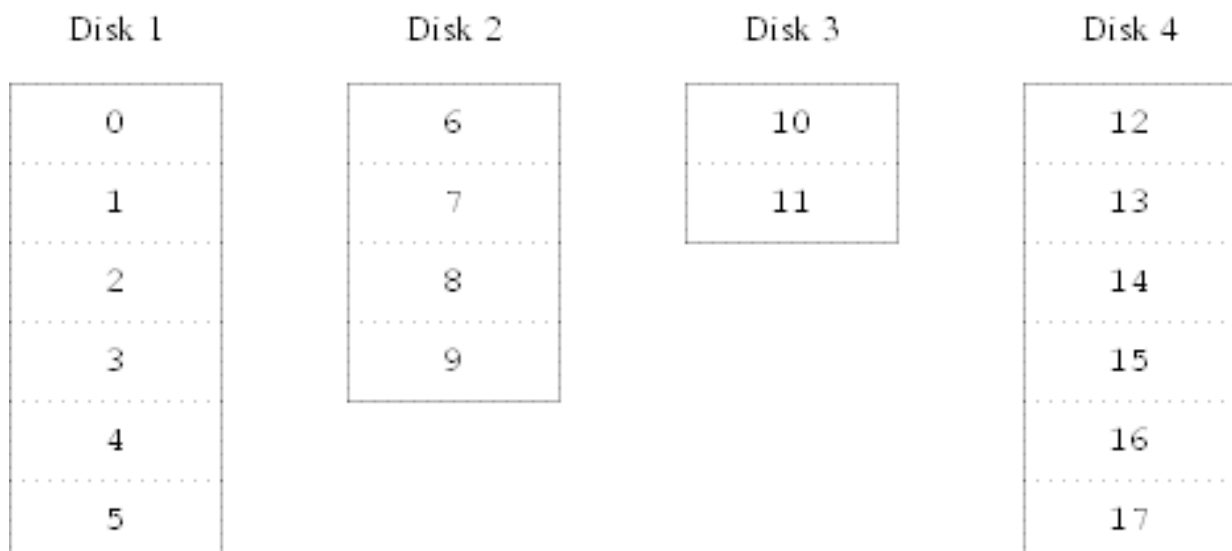
I-1.1 - GVinum

 *GVinum est un gestionnaire de volume, un pilote de disques virtuels.*

I-1.2 - Concaténation (JBOD)

La méthode la plus évidente est de diviser le disque virtuel en groupes de secteurs consécutifs de taille égale aux disques physiques individuels et de les stocker de cette manière. Cette méthode est appelée **concaténation** et a pour avantage que les disques n'ont pas besoin d'avoir de rapport spécifique au niveau de leur taille respective.

Cela fonctionne bien quand l'accès au disque virtuel est réparti de façon identique sur son espace d'adressage.



- On peut ajouter des disques ultérieurement
- Les slices peuvent être quelconques
- La perte d'un disque n'entraîne pas d'autre perte de données... pas mal pour les données non critiques ;)

I-1.3 - Raid-0 (Striping)

Une organisation alternative est de diviser l'espace adressable en composants plus petits, de même taille et de les stocker séquentiellement sur différents périphériques. Par exemple, les 256 premiers secteurs peuvent être stockés sur le premier disque, les 256 secteurs suivants sur le disque suivant et ainsi de suite. Après avoir atteint le dernier disque, le processus se répète jusqu'à ce que les disques soient pleins. Cette organisation est appelée **striping** (découpage en bande ou segmentation) ou RAID-0.

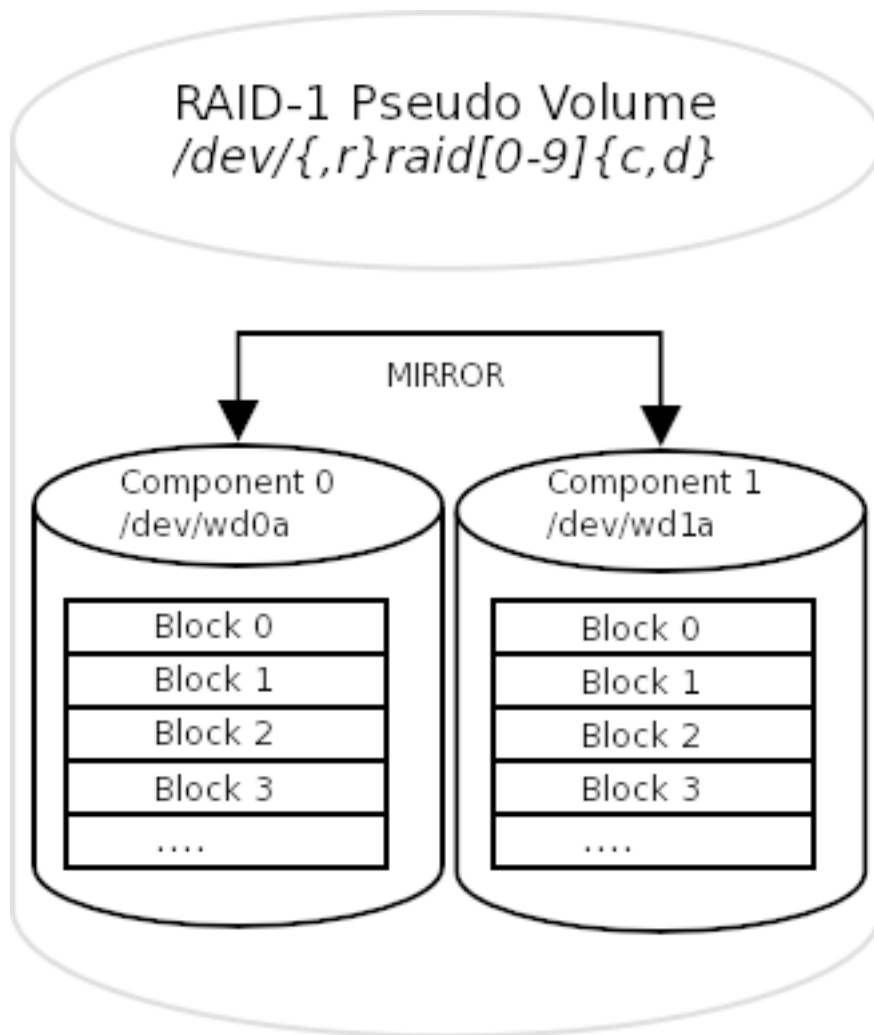
La segmentation exige légèrement plus d'effort pour localiser les données.

| Disk 1 | Disk 2 | Disk 3 | Disk 4 |
|--------|--------|--------|--------|
| 0 | 1 | 2 | 3 |
| 4 | 5 | 6 | 7 |
| 8 | 9 | 10 | 11 |
| 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 |
| 20 | 21 | 22 | 23 |

- On ne peut pas ajouter de disques ultérieurement
- La perte d'un disque entraîne la perte de toutes les données... déconseillé dans les serveurs de fichiers ;)

I-1.4 - Raid-1 (Mirroring)

Le RAID 1 consiste en l'utilisation de disques redondants, c'est-à-dire n disques (en général deux), sur lesquels sont copiées exactement les mêmes données.



- On ne peut pas ajouter de disques ultérieurement
- Les slices doivent être identiques
- Demande au moins deux fois plus d'espace disque réel
- Les écritures doivent être effectuées sur les deux disques
- La perte d'un disque n'entraîne aucune perte de données... conseillé pour de petite capacité.

I-1.5 - Raid-5

Le RAID 5 consiste en l'utilisation d'un calcul de parité des données, afin d'introduire la redondance nécessaire à la reconstruction des données en cas de panne matérielle. Il existe plusieurs méthodes (raid 3 et 4), mais celle-ci semble être la plus efficace...

| Disk 1 | Disk 2 | Disk 3 | Disk 4 |
|--------|--------|--------|--------|
| 0 | 1 | 2 | Parity |
| 3 | 4 | Parity | 5 |
| 6 | Parity | 7 | 8 |
| Parity | 9 | 10 | 11 |
| 12 | 13 | 14 | Parity |
| 15 | 16 | Parity | 17 |

- On ne peut pas ajouter de disques ultérieurement
- Les slices doivent être identiques
- Demande exactement un tiers d'espace disque réel en plus
- La perte d'un disque n'entraîne aucune perte de données... conseillé pour de grosse capacité.

I-1.6 - Divers




Il faut savoir que ces méthodes peuvent être mixées pour obtenir plus de performances, plus de tolérance aux pannes

ici on supporte au plus la perte d'un disque, etc.

N'hésitez pas à vous renseigner...

II - Mise en place

II-1 - Installation de GVinum

 *Gvinum (ou Vinum suivant votre version) est intégré dans le noyau... il n'y a donc aucune installation à effectuer :D*

II-2 - Préparation d'un disque

On part d'un disque sans aucune partition. Considérons que son point de montage est /dev/ad0.

On va faire une slice BSD recouvrant tout le disque.

Préparation d'une slice BSD

```
#!/bin/sh
dd if=/dev/zero of=/dev/ad0 count=2 disklabel /dev/ad0 | disklabel -B -R -r ad0 /dev/stdin
newfs /dev/ad0c
```

II-3 - Configuration de GVinum


On dispose de 3 slices BSD montées respectivement en /dev/ad0c, /dev/ad1c, /dev/ad2c

Pour connaître leur taille, utilisez la commande **df -m /dev/adXc**

On va écrire un gvinum.conf (nb: on se moque du nom, il ne va servir qu'une seule fois)

Configuration JBOD


```
drive a device /dev/ad0c
drive b device /dev/ad1c
drive c device /dev/ad2c
    volume myvol
        plex org concat
            sd length 250000m drive a
            sd length 200000m drive b
            sd length 300000m drive c
```

 *A partir de maintenant, les blocs élémentaires doivent être identiques...*

donc on peut perdre de l'espace disque s'ils sont différents.

Configuration Raid-1

```
drive a device /dev/ad0c
drive b device /dev/ad1c
drive c device /dev/ad2c
    volume myvol
        plex org concat
            sd length 200000m drive a
        plex org concat
            sd length 200000m drive b
        plex org concat
            sd length 200000m drive c
```

 *A partir de maintenant, il faut signaler la taille des secteurs*

(unité élémentaire sur le disque, s'il fait 512Ko un fichier de 1Ko prendra 512Ko sur le disque)

Configuration Raid-0

```
drive a device /dev/ad0c
drive b device /dev/ad1c
drive c device /dev/ad2c
volume myvol
    plex org striped 512k
        sd length 200000m drive a
        sd length 200000m drive b
        sd length 200000m drive c
```

Configuration Raid-5

```
drive a device /dev/ad0c
drive b device /dev/ad1c
drive c device /dev/ad2c
volume myvol
    plex org raid5 512k
        sd length 200000m drive a
        sd length 200000m drive b
        sd length 200000m drive c
```

Ensuite, on configure GVinum, et on initialise la partition avec ce script.

```
#!/bin/sh
gvinum create gvinum.conf
newfs /dev/gvinum/myvol
```

II-4 - Lancement automatique de GVinum

Il semblerait que rien n'ait été prévu pour lancer vinum au démarrage... alors que c'est pourtant le point fort des distributions BSD.

Il nous faut donc nous définir notre service... ça peut toujours servir ;)

On crée un fichier **/etc/rc.d/gvinum**

/etc/rc.d/gvinum

```
# PROVIDE: disks
# KEYWORD: nojail

. /etc/rc.subr

name="gvinum"
start_cmd="gvinum_start"
stop_cmd=""

gvinum_start() {
    case ${gvinum_enable} in
        [Yy][Ee][Ss])
            echo "starting gvinum."
            /sbin/gvinum start ;;
    esac
}

load_rc_config $name
run_rc_command "$1"
# END
```

On ajoute alors dans **/etc/rc.conf** la ligne suivante :

```
/etc/rc.conf
```

```
gvinum_start="YES"
```

II-5 - Montage automatisé

A ce stade, on ne peut pas encore monter une partition sur notre "super-partition". Il nous faut monter le volume avec gvinum, grâce à la commande

```
#!/bin/sh
gvinum start myvol
```

Maintenant, on peut monter une partition. On commence par définir le point de montage dans **/etc/fstab** en ajoutant cette ligne

```
/dev/gvinum/myvol /data ufs defaults,noauto 2 2
```

Ensuite, on peut faire un script pour monter la partition.

```
#!/bin/sh
mount /data
```

III - Divers

III-1 - Compléments

III-1.1 - Tester sa configuration

Vous pouvez visualiser la configuration courante de gvinum.

```
data# gvinum
gvinum -> start
gvinum -> list
(nombre de disques utilisés) drives:
D a                               State: up   (point de montage) A: (taille utilisée par gvinum)/(taille du
disque MB)
...

(nombre de volumes déclarés) volume:
V (nom du volume)                 State: up   Plexes:         (nombre de plexes sur ce volume) Size: (taille du
volume) GB

(nombre de plexes déclarés) plexes:
P (nom du volume).p0              C State: up   Subdisks:      1 Size: (taille du plexe) GB
...

(nombre de sous-disques déclarés) subdisks:
S (nom du volume).p0.s0           State: up   D: a           Size: (taille du sous-disque) GB
...
```

III-1.2 - Sauvegarder la configuration

Vous pouvez sauvegarder la configuration courante de gvinum.

```
data# gvinum printconfig
```

III-1.3 - Ajouter un disque sur un volume concaténé



Pas encore réussi à le faire... si vous avez la solution ;-)

III-1.4 - Changer un disque défectueux sur une configuration Raid-1



En cours de rédaction...

III-1.5 - Changer un disque défectueux sur une configuration Raid-5

Vous venez de changer le disque défectueux, et vous l'initialiser et l'intégrer au volume comme à l'installation...

Ensuite, il faut reconstruire les bits de parité.

```
#!/bin/sh
# option -f possible
gvinum rebuildparity plex
```

nb: il existe l'option **checkparity** qui s'utilise pareillement, et qui ne fait que contrôler...

III-2 - Liens utiles

- [HandBook FreeBSD](#)
- [Vinum Volume Manager](#)

Conclusion

Pour le moment, il semblerait que le développement de GVinum ne soit pas totalement terminé, et il resterait quelques bugs... je vous recommande donc d'éviter de l'utiliser en production dans l'immédiat.