Dernière mise-à-jour : 2020/01/30 03:29

Oracle Solaris 10 - Administration Système N2

Présentation

Type d'Action (Article L. 6313-1) : Action d'acquisition, d'entretien ou de perfectionnement des connaissances.
Objectif : Maîtriser l'administration de base d'Oracle Solaris 10.
Public : Administrateurs Solaris 10.
Pré requis : Avoir suivi la formation Solaris 10 - Administration Système Niveau 1 ou posséder les compétences équivalentes.
Méthode d'apprentissage : Alternance entre un scénario pédagogique clair et précis et des travaux pratiques basés sur des cas et exemples concrets.
Type d'apprentissage : **Apprentissage Accéléré**.
Validation des acquis : Évaluations à l'aide de tests auto-correctifs.
Modalités : 21 heures.
Moyens pédagogiques : Support de cours en ligne téléchargeable au format PDF.
Ressources : Machine virtuelle Oracle Solaris 10.

Programme

- **Gestion du LVM SUN : SVM**
  - Solaris Volume Manager,
    - Introduction,
    - Terminologie,
    - Volumes,
    - Noms des Volumes,
    - State Database,
    - Partitions logicielles,
    - Réserve de Tranches Dynamiques,
    - Fichiers Importants,

    - Gestion des meta devices,
        - T.P.#1 - Préparation du disque,
        - T.P.#2 - RAID 0 - Concaténation,
        - T.P.#3 - RAID 0 - Striping et Partitions Logicielles,
        - T.P.#4 - RAID 1 - Miroir,
        - T.P.#5 - Réserve de Tranches Dynamiques,
        - T.P.#6 - RAID 5 - Striping avec Parité Distribuée,
        - T.P.#7 - Quotas sur UFS,
    - Validation des acquis,
    - **Commandes** : metadb, metainit, metattach, metadetach, metaparam, metastat, metareplace, metaclear, metarename, metahs, growfs, edquota, quotaon, quotacheck, repquota.

- **Gestion du système de fichiers ZFS**
    - Préparaton de la VM Solaris 10,
    - Introduction,
        - Solaris 10 et ZFS
        - Vocabuliare ZFS,
        - Commandes ZFS,
            - La Commande zpool,
            - La Commande zfs,
        - Tranches et ZFS,
    - TP #1 - Gérer le Stockage ZFS,
        - Visualiser l'aide en ligne,
        - Vérifier le statut du pool,
        - Créer un pool en miroir,
        - Ajouter des file systems à un pool existant,
        - Changer le point de montage d'un pool,
        - Ajouter un Hot Spare,
        - Observer l'actvité d'un pool,
        - Mettre en place un quota,
        - Mettre en place une réservation,
        - Créer et gérer des Snapshots,
        - Créer et gérer des Clônes,
        - Utiliser la compression,

        - Utiliser la De-duplication,
        - Utiliser le cryptage,
        - Remplacer un disque défectueux,
        - Détruire un pool,
        - Créer un pool en RAID-5,
        - Créer un pool en RAID-6,
        - Visualiser l'historique d'un pool,
    - Validation des acquis,
    - **Commandes** : zfs, zonecfg, zoneadm, zlogin.

- **Gestion des Zones**
    - Solaris Containers,
        - Zone Globale,
        - Zone locale,
        - Small zone,
        - Big zone,
    - T.P. #1 - L'Installation d'une Zone Locale,
        - La Configuration de la Dataset de la Zone,
        - Installation d'une Zone,
        - Premier Démarrage de votre Zone,
        - Se Connecter à une Zone directement en tant que root,
        - Se Connecter à une Zone directement en tant qu'un Utilisateur,
    - T.P. #2 - Administration des Zones,
        - Partage de Fichiers entre la Zone Globale et une Zone Locale,
        - Retirer un Partage,
        - Allocation de Ressources CPU,
        - Fair Share Scheduler,
        - Allocation de Ressources en Mémoire,
        - Privilèges et Zones Locales,
        - Modification du Nom d'une Zone,
        - Modification de la Dataset d'une Zone,
        - Sauvegarde d'une Zone,
        - Restauration d'une Zone,
        - Création d'une Clone d'une Zone Locale,

- Validation des acquis,
- **Commandes** : zfs, zonecfg, zoneadm, zlogin.

Extrait du Cours

Solaris Volume Manager

Introduction

Le Solaris Volume Manager est un service évolué de la gestion de l'espace disque. Il a été introduit avec Solaris 9.

Le Solaris Volume Manager implémente les niveaux :

- RAID 0 - Concaténation
- RAID 0 - Striping
- RAID 1 - Mirroring
- RAID 5 - Striping avec parité distribuée

> Sur un système SPARC, il existe 8 tranches tandis que sur un système x64/x86 il en existe 16. Sur un système x64/x86, la 8ième tranche est utilisée pour contenir le code pour le boot tandis que le 9ième est utilisée pour des secteurs alternatives sur certains systèmes. La commande format ne permet pas de manipuler plus de 8 tranches. Pour les systèmes x64/x86, il existe un logiciel sous licence GPL appelé **dskpart** qui permet de manipuler les 16 tranches. Le README se trouve à cette adresse.

Terminologie

Volumes

L'entité de base du Solaris Volume Manager est un **volume**. Un volume est composé d'un ensemble de tranches et apparait comme un disque logique via un **pseudo-device** autrement appelé un **meta-device**.

Noms des Volumes

Les noms des volumes sont générés dans les répertoires **/dev/md/dsk** et **/dev/md/rdsk**. Chaque volume possède un nom unique allant de **d0** jusqu'à **d127**. Il n'y a pas de standardisation au niveau des noms. Par conséquent un RAID 5 peut avoir le nom d5, d0 ou d127 par exemple.

State Database

Les informations concernant la configuration courante du Solaris Volume Manager sont stockées dans la ***State Database***. Il est nécessaire de copier cette base de données en plusieurs exemplaires, appelés des ***replicas*** ou **répliques**. Chaque réplique a une taille de 4 Mo par défaut (8192 blocks).

La création des copies de la *State Database* est la première opération à effectuer sur un serveur. Il est conseillé que les répliques résident dans des tranches dédiées à cet effet. Par convention on utilise la partition S7 des disques pour stocker les répliques.

La *State Database* joue un rôle très important sous Solaris Volume Manager. Quand intervient une désynchronisation entre des sous-miroirs, le système doit savoir où se trouvent les données valides.

Dans ce cas il va avoir une **élection à majorité absolue** entre les copies de la *State Database*. Pour cette raison il est nécessaire de créer au moins trois copie de cette dernière.

Qui plus est au moins la moitié des répliques doit être accessible lorsque le système est en cours de fonctionnement. Dans le cas contraire le système crash.

Au moins la moitié des répliques + 1 doit être accessible pour qu'un système puisse démarrer.

Dans le cas d'un système à 2 disques, si un disque tombe en panne, la règle des 50% +1 ne pourra jamais être respectée et le système crash en boucle.

Il est cependant possible de démarrer un système sans obtenir la majorité absolue. Pour cela on positionne le paramètre md:mirrored_root_flag=1 dans le fichier /etc/system :

```
echo "set md:mirrored_root_flag=1" >>/etc/system
```

Les répliques sont utilisées également comme zone de synchronisation des volume-miroirs (RAID 1). Un nombre insuffisant de répliques provoquerait une baisse significative des performances d'entrée/sortie pour les volumes en miroir. En général, on préconise la création d'au moins 2 répliques par volume RAID 1 jusqu'à une limite de 50 répliques par disque.

Partitions logicielles

Les **Partitions logicielles** ou *soft partitions* en anglais permettent de diviser des devices en plusieurs parties. Cette division permet de créer une multitude de systèmes de fichiers sur un device. Il est possible de créer ces partitions logicielles, soit à partir d'un volume, soit directement à partir d'une partition.

Réserve de Tranches Dynamiques

Une **Réserve de Tranches Dynamiques** ou *hot spare pool* en anglais est un ensemble de tranches disponibles en remplacement en cas d'une défaillance d'un disque composant un miroir ou un RAID-5. Lorsqu'une défaillance d'un disque survient, SVM va utiliser une partition de la Réserve de Tranches Dynamiques afin de remplacer ce disque. Il est nécessaire bien entendu que les tranches stockées dans la Réserve de Tranches Dynamiques soient d'une taille égale ou supérieure à la tranche à remplacer. Il est impératif d'ajouter les tranches par ordre de taille de la plus petite à la plus grande. En effet lorsque SVM recherche une partition de remplacement, celui-ci lit la liste des tranches dans l'ordre dans lequel celles-ci ont été ajoutées dans la réserve.

Fichiers Importants

- **/etc/lvm/mddb.cf**
  - contient l'emplacement de la metadb et de ses répliques,
- **/kernel/drv/md.conf**
  - reflète les informations du fichier mddb.cf. Il est nécessaire au fonctionnement du driver md.

Gestion des meta devices

LAB#1 - Préparation du disque

Préparez votre disque de la façon suivante :

- tranche 3 de 20Mo minimum
- tranche 4 d'une taille deux fois celle du tranche 5
- tranche 5 et tranche 6 d'une taille égale.

Vous devez obtenir une table des partitions **similaire** à celle-ci :

```
partition> pri
Volume:  My Disk
Current partition table (original):
Total disk cylinders available: 2607 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders        Size            Blocks
  0       root    wm      68 - 1189        8.59GB    (1122/0/0) 18024930
  1       swap    wu       1 -   67      525.56MB    (67/0/0)    1076355
  2     backup    wm       0 - 2606       19.97GB    (2607/0/0) 41881455
  3 unassigned    wm    1190 - 1192       23.53MB    (3/0/0)       48195
  4 unassigned    wm    1193 - 1454        2.01GB    (262/0/0)   4209030
  5 unassigned    wm    1455 - 1637        1.40GB    (183/0/0)   2955960
  6 unassigned    wm    1639 - 1821        1.40GB    (183/0/0)   2939895
  7       home    wm    1822 - 2606        6.01GB    (785/0/0)  12611025
  8       boot    wu       0 -    0        7.84MB    (1/0/0)       16065
  9 unassigned    wm       0               0         (0/0/0)           0
```

Voici quelques indications pour vous faciliter la tâche ci-dessus :

- Créez un répertoire **/backup** à la racine de votre système de fichiers,

- Créez un **tar** du contenu de **/export/home** et placez-le dans **/backup**,
- Démontez **/export/home**,
- Commentez la ligne **/export/home** dans le fichier **/etc/vfstab**,
- Lancez la commande **format** et choisissez le menu **partitions**,
- Diminuez la taille de la tranche 7 de façon à récupérer 4 Go d'espace disque,
- Créez la tranche 3 de 20mo,
- Créez la tranche 4 de 2Go,
- Créez les tranches 5 et 6 d'une taille égale, la somme étant équivalente à l'espace disque restant,
- Nommez la nouvelle table de partitions “My Table” en utilisant les menus **partition > name**,
- Montez au menu supérieur en utilisant la commande **quit**,
- Nommez le volume “My Disk” avec le menu **volname**,
- Sauvez votre table dans l'emplacement par défaut en utilisant la commande **save**,
- Quittez la commande format avec **quit**,
- Créez un nouveau FileSystem sur S7 avec la commande **newfs /dev/dsk/c0t0d0s7**,
- Éditez le fichier **/etc/vfstab** en supprimant le caractère **#** devant la ligne **/export/home**,
- Montez /export/home avec la commande **mount /export/home**,
- Restaurez le contenu du fichier tar dans /backup vers /export/home.

Vérifiez ensuite qu'il n'existe pas de State Database :

```
# metadb -i
metadb: unknown: aucune base de données existante
```

Créez maintenant la *State Database* ainsi que 2 répliques dans votre tranche 3 :

```
# metadb -f -a -c 3 c0t0d0s3
# metadb
        flags           first blk       block count
     a        u         16              8192            /dev/dsk/c0t0d0s3
     a        u         8208            8192            /dev/dsk/c0t0d0s3
     a        u         16400           8192            /dev/dsk/c0t0d0s3
```

Les options de la commande sont :

- **-a** : Permet d'ajouter une réplique.
- **-f** : Permet de forcer un état de la metadb. Cette option est à utiliser lors de la création de la première réplique ou lors de la suppression de la dernière.
- **-c < n >** : Permet de spécifier le nombre de répliques à ajouter sur une partition. Cette option permet de placer plusieurs répliques sur une seule partition.

Constatez le contenu du fichier **/etc/lvm/mddb.cf** :

```
# cat /etc/lvm/mddb.cf
#metadevice database location file do not hand edit
#driver minor_t daddr_t device id       checksum
sd      3       16      id1,sd@SATA_____VBOX_HARDDISK____VB11d23ca4-8967fc4b/d -4385
sd      3       8208    id1,sd@SATA_____VBOX_HARDDISK____VB11d23ca4-8967fc4b/d -12577
sd      3       16400   id1,sd@SATA_____VBOX_HARDDISK____VB11d23ca4-8967fc4b/d -20769
```

ainsi que le fichier **/kernel/drv/md.conf** :

```
# cat /kernel/drv/md.conf
#
#pragma ident   "@(#)md.conf    2.2     04/04/02 SMI"
#
# Copyright 2004 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#
# The parameters nmd and md_nsets are obsolete.  The values for these
# parameters no longer have any meaning.
name="md" parent="pseudo" nmd=128 md_nsets=4;
# Begin MDD database info (do not edit)
mddb_bootlist1="sd:3:16:id1,sd@SATA_____VBOX_HARDDISK____VB11d23ca4-8967fc4b/d
sd:3:8208:id1,sd@SATA_____VBOX_HARDDISK____VB11d23ca4-8967fc4b/d
sd:3:16400:id1,sd@SATA_____VBOX_HARDDISK____VB11d23ca4-8967fc4b/d";
# End MDD database info (do not edit)
```

Constatez maintenant la création de la metadb :

```
# metadb -i
        flags           first blk       block count
     a        u         16              8192            /dev/dsk/c0t0d0s3
     a        u         8208            8192            /dev/dsk/c0t0d0s3
     a        u         16400           8192            /dev/dsk/c0t0d0s3
 r - la réplique ne contient pas d'informations de transfert de périphérique
 o - réplique active avant la dernière modification de la configuration de la base de données du métapériphérique
mddb
 u - la réplique est mise à jour
 l - le localisateur pour cette réplique a été lu avec succès
 c - la réplique se trouvait dans /etc/lvm/mddb.cf
 p - un patch a été appliqué à la réplique dans le noyau
 m - la réplique est maître : elle est sélectionnée comme réplique d'entrée
 W - la réplique contient des erreurs d'écriture de périphérique
 a - la réplique est active, des validations ont lieu dans cette réplique
 M - la réplique a rencontré des problèmes avec les blocs maîtres
 D - la réplique a rencontré des problèmes avec les blocs de données
 F - la réplique a rencontré des problèmes liés au format
 S - la réplique est trop petite pour contenir la base de données actuelle
 R - la réplique contenait des erreurs de lecture de périphérique
```

Créez ensuite un filesystem UFS sur le tranche 5 :

```
# newfs -v -b 4096 -m 10 /dev/rdsk/c0t0d0s5
newfs: construction d'un nouveau système de fichiers /dev/rdsk/c0t0d0s5 : (y/n)? y
pfexec mkfs -F ufs /dev/rdsk/c0t0d0s5 2955960 63 -1 4096 1024 32 10 3 4096 t 0 -1 8 256 n
Avertissement : 5448 secteur(s) du dernier cylindre n'ont pas été alloués
/dev/rdsk/c0t0d0s5:     2955960 secteurs dans 482 cylindres de 48 pistes, 128 secteurs
        1443,3 Mo dans 69 groupes de cylindres (7 c/g, 21,00 Mo/g, 5056 i/g)
sauvegardes de super-blocs (pour fsck -F ufs -o b=#) at:
 32, 43168, 86304, 129440, 172576, 215712, 258848, 301984, 345120, 388256,
 2545056, 2588192, 2631328, 2674464, 2717600, 2752544, 2795680, 2838816,
 2881952, 2925088
```

Montez votre tranche 5 à la racine de votre système de fichiers et créez un fichier appelé **SVM test** conteant la chaîne *svm_test* :

```
# mkdir /slice5
# mount /dev/dsk/c0t0d0s5 /slice5
# cd /slice5
# ls
lost+found
# echo "SVM test" > svm_test
```

LAB#2 - RAID 0 - Concaténation

Le Solaris Volume Manager peut être utilisé pour concaténer des tranches afin de créer un volume contigu. Afin d'y parvenir, il convient d'utiliser la commande **metainit** en passant en argument à la commande au moins une tranche **démontée**. Dans notre cas, vous allez concaténer les tranches 5 et 6 de votre disque. Saisissez donc les commandes suivantes :

```
# cd ..
# umount /slice5
# metainit d0 2 1 c0t0d0s5 1 c0t0d0s6
d0: Concaténation/Bande configurée
```

Dans ce cas, vous avez créer un volume **d0** de deux tranches qui se suivent physiquement. Le chiffre **2** indique le nombre de **bandes** tandis que les chiffres **1** indiquent le nombre de tranches par bande. Dans le cas d'un volume concaténé il devrait y avoir autant de bandes que de tranches. Constatez ensuite la création du volume en utilisant la commande **metastat** :

```
# metastat -p
d0 2 1 c0t0d0s5 \
         1 c0t0d0s6
# metastat d0
d0: Concat/Stripe
    Taille : 5895855 blocs (2,8 GB)
    Bande 0 :
        Périphérique   Bloc de débu Base        Redistribution
```

```
        c0t0d0s5              0     Non         Oui
    Bande 1 :
        Périphérique   Bloc de débu Base        Redistribution
        c0t0d0s6              0     Non         Oui

Device Relocation Information:
Device   Reloc  Device ID
c0t0d0   Oui    id1,sd@SATA_____VBOX_HARDDISK____VB11d23ca4-8967fc4b
```

Montez ensuite le volume **d0** et constatez l'espace disque libre :

```
# mount /dev/md/dsk/d0 /slice5
# df -h
Système de fichiers  taille utilisé  dispo capacité  Monté sur
/dev/dsk/c0t0d0s0      8,5G   4,1G   4,3G    49%    /
/devices                 0K     0K     0K     0%    /devices
ctfs                     0K     0K     0K     0%    /system/contract
proc                     0K     0K     0K     0%    /proc
mnttab                   0K     0K     0K     0%    /etc/mnttab
swap                   1,5G  1016K   1,5G     1%    /etc/svc/volatile
objfs                    0K     0K     0K     0%    /system/object
sharefs                  0K     0K     0K     0%    /etc/dfs/sharetab
/usr/lib/libc/libc_hwcap1.so.1
                       8,5G   4,1G   4,3G    49%    /lib/libc.so.1
fd                       0K     0K     0K     0%    /dev/fd
swap                   1,5G   112K   1,5G     1%    /tmp
swap                   1,5G    36K   1,5G     1%    /var/run
mypool                 341M    37K   340M     1%    /mypool
/dev/dsk/c0t0d0s7      5,9G   6,0M   5,9G     1%    /export/home
/dev/md/dsk/d0         1,4G   1,4M   1,2G     1%    /slice5
```

Vous noterez que le volume monté ne reflète pas l'addition des deux tranches en termes de taille. En effet, pour l'instant le *file system* précédemment présent sur **c0t0d0s5** n'occupe pas tout le volume d0. Afin d'étendre le *file system* à la taille maximale du volume, il convient d'utiliser la commande **growfs** :

```
# growfs -M /slice5 /dev/md/rdsk/d0
Avertissement : 2386 secteur(s) du dernier cylindre n'ont pas été alloués
/dev/md/rdsk/d0:        5895854 secteurs dans 960 cylindres de 48 pistes, 128 secteurs
        2878,8 Mo dans 138 groupes de cylindres (7 c/g, 21,00 Mo/g, 5056 i/g)
sauvegardes de super-blocs (pour fsck -F ufs -o b=#) at:
 32, 43168, 86304, 129440, 172576, 215712, 258848, 301984, 345120, 388256,
 5505056, 5548192, 5591328, 5634464, 5677600, 5720736, 5763872, 5807008,
 5850144, 5893280
```

Constatez ensuite la réussite de la commande :

```
# df -h
Système de fichiers  taille utilisé  dispo capacité  Monté sur
/dev/dsk/c0t0d0s0      8,5G   4,1G   4,3G    49%    /
/devices                 0K     0K     0K     0%    /devices
ctfs                     0K     0K     0K     0%    /system/contract
proc                     0K     0K     0K     0%    /proc
mnttab                   0K     0K     0K     0%    /etc/mnttab
swap                   1,5G  1016K   1,5G     1%    /etc/svc/volatile
objfs                    0K     0K     0K     0%    /system/object
sharefs                  0K     0K     0K     0%    /etc/dfs/sharetab
/usr/lib/libc/libc_hwcap1.so.1
                       8,5G   4,1G   4,3G    49%    /lib/libc.so.1
fd                       0K     0K     0K     0%    /dev/fd
swap                   1,5G   112K   1,5G     1%    /tmp
swap                   1,5G    36K   1,5G     1%    /var/run
mypool                 341M    37K   340M     1%    /mypool
/dev/dsk/c0t0d0s7      5,9G   6,0M   5,9G     1%    /export/home
/dev/md/dsk/d0         2,7G   2,8M   2,6G     1%    /slice5
```

Naviguez ensuite au point de montage de votre volume et constatez la présence du fichier précédemment créé :

```
# cd /slice5
# ls
```

```
lost+found  svm_test
# cat svm_test
SVM test
```

Créez ensuite un deuxième fichier de contrôle :

```
# echo "svm_test_2" > svm_test_2
# ls
lost+found  svm_test    svm_test_2
```

Constatez l'état du volume avec la commande **metastat** :

```
# metastat
d0: Concat/Stripe
    Taille : 5895855 blocs (2,8 GB)
    Bande 0 :
        Périphérique   Bloc de débu Base        Redistribution
        c0t0d0s5             0     Non         Oui
    Bande 1 :
        Périphérique   Bloc de débu Base        Redistribution
        c0t0d0s6             0     Non         Oui

Device Relocation Information:
Device   Reloc  Device ID
c0t0d0   Oui    id1,sd@SATA_____VBOX_HARDDISK____VB11d23ca4-8967fc4b
```

Afin de rajouter une tranche à un volume existant, il convient d'utiliser la commande **metattach**. Ajoutez donc la tranche c0t0d0s4 au volume d0 :

```
# metattach d0 c0t0d0s4
d0: composant connecté
```

Augmentez ensuite la taille du *filesystem* :

```
# growfs -M /slice5 /dev/md/rdsk/d0
```

```
Avertissement : 1996 secteur(s) du dernier cylindre n'ont pas été alloués
/dev/md/rdsk/d0:        10104884 secteurs dans 1645 cylindres de 48 pistes, 128 secteurs
        4934,0 Mo dans 235 groupes de cylindres (7 c/g, 21,00 Mo/g, 5056 i/g)
sauvegardes de super-blocs (pour fsck -F ufs -o b=#) at:
 32, 43168, 86304, 129440, 172576, 215712, 258848, 301984, 345120, 388256,
 9681056, 9724192, 9767328, 9810464, 9853600, 9896736, 9939872, 9983008,
 10026144, 10069280
```

Constatez la taille du volume :

```
# df -h
Système de fichiers  taille utilisé  dispo capacité  Monté sur
/dev/dsk/c0t0d0s0      8,5G   4,1G   4,3G    49%    /
/devices                 0K     0K     0K     0%    /devices
ctfs                     0K     0K     0K     0%    /system/contract
proc                     0K     0K     0K     0%    /proc
mnttab                   0K     0K     0K     0%    /etc/mnttab
swap                   1,5G  1016K   1,5G     1%    /etc/svc/volatile
objfs                    0K     0K     0K     0%    /system/object
sharefs                  0K     0K     0K     0%    /etc/dfs/sharetab
/usr/lib/libc/libc_hwcap1.so.1
                       8,5G   4,1G   4,3G    49%    /lib/libc.so.1
fd                       0K     0K     0K     0%    /dev/fd
swap                   1,5G   112K   1,5G     1%    /tmp
swap                   1,5G    36K   1,5G     1%    /var/run
mypool                 341M    37K   340M     1%    /mypool
/dev/dsk/c0t0d0s7      5,9G   6,0M   5,9G     1%    /export/home
/dev/md/dsk/d0         4,7G   4,8M   4,5G     1%    /slice5
```

Constatez ensuite la présence de vos fichiers :

```
# pwd
/slice5
# ls
```

```
lost+found  svm_test    svm_test_2
# cat svm_test_2
svm_test_2
```

Démontez ensuite votre volume et supprimez-le grâce à la commande **metaclear** :

```
# cd ..
# umount /slice5
# metaclear d0
d0: Concaténation/Bande supprimée#
```

Montez ensuite la tranche c0t0d0s5 sur slice5 et constatez la présence de vos fichiers :

```
# mount /dev/dsk/c0t0d0s5 /slice5
# cd /slice5
# ls
lost+found  svm_test    svm_test_2
```

Notez que les fichiers et leurs contenus sont intacts.

Dans l'exemple précédent, vous avez créé un volume de tranches qui se suivent physiquement. Cette fois-ci, il convient de récupérer deux tranches *orphelines* sur votre disque. Créez donc un volume contenant les tranches 4 et 6 de votre disque :

```
# cd ..
# umount /slice5
# metainit d0 2 1 c0t0d0s4 1 c0t0d0s6
d0: Concaténation/Bande configurée
# newfs -v -b 4096 -m 10 /dev/md/rdsk/d0
newfs: construction d'un nouveau système de fichiers /dev/md/rdsk/d0 : (y/n)? y
pfexec mkfs -F ufs /dev/md/rdsk/d0 7148925 63 240 4096 1024 96 10 3 8192 t 0 -1 8 14 n
Avertissement : 2836 secteur(s) du dernier cylindre n'ont pas été alloués
```

```
/dev/md/rdsk/d0:        7148924 secteurs dans 473 cylindres de 240 pistes, 63 secteurs
        3490,7 Mo dans 158 groupes de cylindres (3 c/g, 22,15 Mo/g, 2688 i/g)
sauvegardes de super-blocs (pour fsck -F ufs -o b=#) at:
 32, 45456, 90880, 136304, 181728, 227152, 272576, 318000, 363424, 408848,
 6722784, 6768208, 6813632, 6859056, 6904480, 6949904, 6995328, 7040752,
 7086176, 7131600
```

Montez ce volume :

```
# mount /dev/md/dsk/d0 /slice5
# df -h
Système de fichiers  taille utilisé  dispo capacité  Monté sur
/dev/dsk/c0t0d0s0      8,5G   4,1G   4,3G    49%    /
/devices                 0K     0K     0K     0%    /devices
ctfs                     0K     0K     0K     0%    /system/contract
proc                     0K     0K     0K     0%    /proc
mnttab                   0K     0K     0K     0%    /etc/mnttab
swap                   1,5G  1016K   1,5G     1%    /etc/svc/volatile
objfs                    0K     0K     0K     0%    /system/object
sharefs                  0K     0K     0K     0%    /etc/dfs/sharetab
/usr/lib/libc/libc_hwcap1.so.1
                       8,5G   4,1G   4,3G    49%    /lib/libc.so.1
fd                       0K     0K     0K     0%    /dev/fd
swap                   1,5G   112K   1,5G     1%    /tmp
swap                   1,5G    36K   1,5G     1%    /var/run
mypool                 341M    37K   340M     1%    /mypool
/dev/dsk/c0t0d0s7      5,9G   6,0M   5,9G     1%    /export/home
/dev/md/dsk/d0         3,4G   3,4M   3,0G     1%    /slice5
```

Vérifiez le contenu de votre volume :

```
# cd /slice5
# ls
```

```
lost+found
```

Vos fichiers de contrôle ne sont pas présents car ils sont stockés sur le file system sur c0t0d0s5 qui ne fait pas partie de ce d0.

Dernièrement, démontez votre volume et détruisez-le avec la commande **metaclear** :

```
# cd ..
# umount /slice5
# metaclear d0
d0: Concaténation/Bande supprimée#
```

...

<html>

Copyright © 2004-2017 Hugh Norris.

Ce(tte) oeuvre est mise à disposition selon les termes de la Licence Creative Commons Attribution - Pas d'Utilisation Commerciale - Pas de Modification 3.0 France.

</html>