StorIQ

# Serveur NAS

## StorIQ System v. 5.0

## Administration Avancée

**Auteur: Emmanuel FLORAC**
**Réf. NAS-ADV-MAN**
**Version: 1.0.1**

Mise à jour: 15/10/2015

Contacts :
tel: 01 78 94 84 00
support@intellique.com
info@Intellique.com

Intellique.com

# Table des matières

## Installation en mode avancé
### Redimensionnement des partitions

parted ne sait pas redimensionner XFS. Il faut donc supprimer et recréer la partition avec exactement le même point de départ. C'est évidemment dangereux...

Si on agrandit le RAID (soit "tw_cli migrate", soit "arcconf MODIFY") depuis lequel on a démarré, le système ne peut pas toujours relire la table de partitions (sauf avec un contrôleur adaptec 7xxx ou supérieur). Dans ce cas il faut REDÉMARRER LE SERVEUR avant de tenter de manipuler les partitions. Après le redémarrage, on peut si nécessaire supprimer et recréer la partition agrandie.

Si on peut relire la table des partitions ( utiliser `blockdev --rereadpt /dev/sdXX` ); on doit redimensionner les pv et lv qui sont dessus, en utilisant `pvresize /dev/sdXX`, puis `lvextend -L +XXG /dev/vg/lv /dev/sdXX` puis utiliser xfs_growfs ou resize_reiserfs pour retailler le système de fichiers.

**ATTENTION** Si on modifie la table de partition, ne pas oublier de faire `mount /mnt/rescue && lilo` (sur une StorIQ 2.x) ou `grub-install /dev/sda` (sur une StorIQ 3.x et supérieur).

# 1. Configuration des disques

Pour optimiser les performances, la règle générale est la suivante : minimiser l'antélecture (read-ahead) sur les disques physiques, le maximiser sur les disques logiques (RAID logiciel ou LVM). Une configuration typique comprenant un RAID logiciel avec deux disques /dev/sdb et /dev/sdc en RAID 0 LVM serait donc comme suit :

```
blockdev –setra 256 /dev/sdb
blockdev –setra 256 /dev/sdc
blockdev –setra 16384 /dev/vg0/lv0
```

La valeur optimale typique de read-ahead pour une machine avec des contrôleurs RAID matériels se situe entre 4096 et 131 072 selon le nombre et le type de disques. Une bonne approximation sera entre `1024 * <nombre de disques de données>` et `4096 * <nombre de disques de données>`.

Il faut aussi augmenter la longueur de file d'attente:

```
# 3Ware : 512
echo 512 > /sys/block/sda/queue/nr_requests

# Adaptec : 1024
echo 1024 > /sys/block/sda/queue/nr_requests
```

Attention, les Adaptec 5xxx ont un bug lors de fortes sollicitations elles cessent de répondre. Malheureusement il n'y a pas de correctif, on peut juste augmenter le *timeout* par défaut de 30 secondes à 45 (voire plus):

```
echo 45 > /sys/block/sda/device/timeout
```

On aura de bien meilleures performances en laissant le contrôleur RAID gérer lui-même les files d'attente d'IOs, donc on utilisera le scheduler "noop" sur les périphériques disques (c'est le réglage par défaut des noyaux StorIQ):

```
echo 'noop' > /sys/block/sda/queue/scheduler
```

Penser également à déclarer les SSD comme tels:

```
echo 0 > /sys/block/sda/queue/rotational
```

### Optimisation des performances XFS

Pour optimiser les performances, on peut paramétrer le FS au moment du mkfs en fonction de la géométrie du raid (paramètres swidtth et sunit, ou sw et su). On doit aussi jouer sur les options de montage :

- nobarrier, noatime : pour minimiser les accès au journal
- align, noalign : pour que les limites de blocs du FS correspondent bien aux blocs du RAID
- inode64 : pour les FS de plus de 4 To, permet de répartir les inodes au plus près des blocs de données. Rend incompatible avec les noyaux 32 bits et avec certains clients NFS, attention!
- allocsize=<size> : taille de préallocation (défaut 64Ko). S'il n'y a que des gros fichiers, mieux vaut mettre un allocsize grand voire très grand ( 1M ou plus).

Autres options :

### Filestreams

Filestreams (sur XFS) permet d'allouer des fichiers successifs sur des extents contigus. Ainsi, sur des séries d'images numérotées, on assure que les fichiers seront optimalement regroupés à la lecture.

Avec un noyau récent, on peut utiliser "filestreams" soit en option de montage ( mount -o filestreams ) soit en activant l'attribut étendu filestreams sur un dossier pour indiquer que l'allocation doit se faire en mode "filestreams".

Autre option de montage classique, `logbufs=8,logbsize=256k` pour maximiser les buffers. Je n'ai pas remarqué d'effet sensible sur les systèmes modernes.

### Quotas XFS

XFS permet les quotas par dossier, (appelés "projets" en terminologie XFS)

- monter le FS avec l'option pquota ( "project quota" )
- créer un fichier /etc/projects listant les dossiers à utiliser avec quotas et leurs numéros (n'importe quoi mais unique) :

```
1:/mnt/raid/tamere
19:/mnt/raid/enslip
42:/mnt/raid/chezprisunic
```

- créer un fichier /etc/projid qui fait correspondre les numéros de projets avec un nom (label):

```
1:tamere
19:enslip
42:chezprisunic
```

- initialiser les quotas pour un projet en donnant le label ou le numéro (en pratique selon les version le numéro seul est sûr) :

```
xfs_quota -x -c "project -s 1"
```

- vérifier les quotas :

```
xfs_quota -x -c "project -c 1"
```

- status des quotas, se vérifie au niveau du FS :

```
~# xfs_quota -x -c report /mnt/raid
Project quota on /mnt/raid (/dev/md0)
                               Blocks
Project ID       Used       Soft       Hard    Warn/Grace
---------- --------------------------------------------------
1                   0          0          0     00 [--------]
```

Si on active les quotas dans le fstab et qu'on fait un `mount -o remount /mnt/truc`, l'option quota est bien listé dans la sortie de "mount" mais c'est un piège, les quotas ne sont pas actifs! de manière générale, beaucoup d'options de montage de XFS ne marchent pas lors d'un "remount", il faut donc démonter puis remonter.

### Mise en veille des disques et XFS

Si on active la mise en veille des disques sur les contrôleurs RAID qui le supportent, il faut impérativement allonger le délai de time out du journal pour éviter des Ooops:

```
echo 720000 > /proc/sys/fs/xfs/xfssyncd_centisecs
```

### Alternative : JFS

Dans certains cas rares JFS peut être mieux adapté. Son principal avantage est le support de 8192 ACLs par fichier, contre 25 seulement pour XFS. Ceci au prix d'une performance un peu inférieure à XFS, et l'impossibilité de défragmenter le système de fichiers -- souvent utile quand celui-ci est âgé.

## 2.Chiffrement des partitions

### créer un filesystem crypté

On installe d'abord cryptsetup, puis : Par exemple pour crypter /dev/md0 en AES 256 :

```
cryptsetup -c aes-cbc-essiv:sha256 -y -s 256 luksFormat /dev/md0
```

Pour utiliser le périphérique :

```
cryptsetup luksOpen /dev/md0 crypt1
```

Ensuite il faut penser à créer un filesystem et modifier le /etc/fstab:

```
mkfs -t xfs /dev/mapper/crypt1
```

Enfin il faut remplir le fichier /etc/crypttab avec les informations nécessaires. Voir le "man crypttab". On peut soit utiliser un mot de passe soit par exemple une clef USB, ou autre.

### accès par mot de passe

Exemple de crypttab correspondant:

```
crypt1  /dev/md0        none    tries=3,timeout=60,loud,luks
```

### Utiliser une clé USB pour stocker la clef de chiffrement

Pour crypter en utilisant un fichier sur une clef USB, le mieux est de générer un fichier aléatoire avec /dev/random, un une clef avec ssl-keygen, gpg, etc. Ensuite on met ce fichier sur une clef USB avec un point de montage et on l'utilise pour verrouiller le volume :

```
 cryptsetup luksAddKey /dev/md0 /mnt/key/crypt1.key
Enter any LUKS passphrase:
key slot 0 unlocked.
Command successful.
```

Ensuite il faut modifier le crypttab pour utiliser cette clef plutôt qu'un prompt, ainsi :

```
crypt1  /dev/md0        /mnt/key/crypt1.key     loud,luks
```

Là où ça se corse, c'est qu'il faut qu'au démarrage, la clef USB soit pleinement activée _avant_ le démarrage de /etc/init.d/cryptdisks, ensuite il faut que le montage des FS locaux soit fait suffisamment tard. J'ai modifié la séquence de démarrage ainsi :

```
cryptdisks              :   48              13   13   13   48
cryptdisks-early        :   59              11   11   11   59
mountall.sh             :                   15   15   15   35
```

J'ai donc décalé le démarrage normal de cryptdisks dans les runlevels 2,3,4, puis j'ai refait un mountall après. Par ailleurs j'ai modifié le /etc/default/cryptdisks pour qu'il monte la clef tout seul:

```
~# more /etc/default/cryptdisks
# Run cryptdisks at startup ?
CRYPTDISKS_ENABLE=Yes

# Mountpoints to mount, before starting cryptsetup. This is useful for
# keyfiles on removable media. Seperate mountpoints by space.
CRYPTDISKS_MOUNT="/mnt/key"

# Default check script, see /lib/cryptsetup/checks/
# Takes effect, if the 'check' option is set in crypttab without a value
CRYPTDISKS_CHECK=vol_id

# Default precheck script, see
# Takes effect, if the 'precheck' option is set in crypttab without a value
CRYPTDISKS_PRECHECK=

# Default timeout in seconds for password prompt
# Takes effect, if the 'timeout' option is set in crypttab without a value
CRYPTDISKS_TIMEOUT=180
```

Voici le point de montage de la clef USB (dans /etc/fstab) :

```
/dev/disk/by-id/usb-JUNGSOFT_NEXDISK-part1 /mnt/key vfat
noauto,rw,uid=0,gid=0,umask=277 0 0
```

Noter qu'elle est en noauto, comme ça elle est montée et démontée au démarrage de diskcrypt, on peut donc l'insérer avant le boot, attendre la fin du boot et la récupérer.

# 3.Configuration réseau

## 10 Gigabits

Myricom propose plein d'optimisations. Voir (le site) https://www.myricom.com/scs/READMES/README.myri10ge-linux. On retiendra surtout d'ajouter à /etc/sysctl.conf:

```
net.core.rmem_max = 16777216
net.core.wmem_max = 16777216
net.ipv4.tcp_rmem = 4096 87380 16777216
net.ipv4.tcp_wmem = 4096 65536 16777216
net.core.netdev_max_backlog = 250000
```

## Agrégation des ports réseaux

Paramétrages bonding : le 802.3ad nécessite 2 machines configurées sur le réseau ou une configuration du switch. Il n'y a pas d'amélioration des performances si le switch ne coopère pas (c'est à dire jamais... ) Balance-alb ou balance-rr sont les meilleurs en performances mais cela varie selon les protocoles, les switchs, le nombre d'interface réseau... Il faut tester chaque configuration. En pratique balance-tlb est le plus "tranquille" : gain de performance et pas de problème de compatibilité.

Il est possible de configurer plusieurs groupes d'interfaces 'bond0, bond1, etc.) avec des modes de fonctionnement distincts. Voir la documentation de « bonding_cli ».

## Authentification LDAP

Commencer par faire un dpkg-reconfigure libpam-ldap

Ensuite modifier /etc/nsswitch.conf ainsi :

```
passwd:     files ldap compat
group:      files ldap compat
shadow:     files ldap compat
```

Puis modifier les fichiers dans /etc/pam.d suivants : /etc/pam.d/common-account :

```
account sufficient    pam_ldap.so
account required    pam_unix.so try_first_pass
```

/etc/pam.d/common-auth:

```
auth sufficient        pam_ldap.so
auth required        pam_unix.so nullok_secure try_first_pass
```

/etc/pam.d/common-password:

```
password sufficient    pam_ldap.so
password required    pam_unix.so nullok obscure min=4 max=8 md5
try_first_pass
```

Ensuite il faut entrer les paramètres du serveur LDAP dans /etc/ldap/ldap.conf :

```
BASE  dc=example,dc=org
```

```
URI   ldap://192.168.1.30
```

Pour finir, on redémarre nscd :

```
service nscd restart
```

# 4.Partage de fichiers

## NFS

Note : Pour les clients Macs et BSD, il faut exporter en mode "insecure".

J'ai fabriqué un petit moniteur des serveurs NFS (2,3 et 4), munin-nfsd-perf. Il va automatiquement se configurer. Il donne 4 valeurs :

- CPU time % : le pourcentage de CPU consommé par les démons NFS.
- requests/s : le nombre de connexions par secondes.
- working threads % : le pourcentage de démons nfsd qui travaillent le plus.
- idle threads % : le pourcentage de démons nfsd qui ne font presque rien.

La somme des "workers" et des "idlers" est normalement inférieure à 100%. S'il y a une forte proportion de "idlers" (plus de 10 ou 20%), c'est qu'il y a plus de démons nfsd qu'il n'est nécessaire ( abaisser la valeur de RPCNFSDCOUNT dans /etc/default/nfs-kernel-server et relancer ). S'il y a une forte proportion de "workers" (plus de 50%), il peut ne pas y avoir assez de démons nfsd. De même si le nombre de connexions par seconde est supérieur au nombre de clients connectés, cela signifie que les clients « migrent » sans cesse d'un serveur à l'autre, donc qu'il n'y a sans doute pas suffisamment de démons.

Problème ennuyeux le plus courant avec NFS: le message "*rpc.statd not responding, timed out*". En général la procédure suivante résout le problème :

```
service nfs stop

service rpcbind stop

rm -rf /var/lib/nfs/statd/sm/*

rm -rf /var/lib/nfs/statd/sm.bak/*

service rpcbind start

service nfs start
```

## NFS4

NFS4 utilise le concept de "racine virtuelle unique". Les exports font partie d'un pseudo-fs, identifiés par un "fsid" (toujours 0, pour l'instant). *Tous les exports* doivent être des sous-dossiers de la racine virtuelle. Pour les détails : https://wiki.linux-nfs.org/wiki/index.php/Nfsv4_configuration_fr

## SAMBA

### Samba et les Macs

Les performances sont très faibles comparées à NFS.

Le finder de Mac OS X 10.5 ne peut pas (via command-K) se connecter à un partage samba si le nom du serveur contient un tiret ( - ). On arrive à forcer en utilisant comme nom d'utilisateur `<adresse IP>\user` cependant.

Si on utilise des droits un tant soit peu sophistiqués dans un réseau mixte

Windows/Mac, il faut désactiver les extensions Unix dans Samba, sinon ça pose des problèmes d'accès. Ajouter dans la section "global" de *smb.conf*:

```
unix extensions = no
```

puis redémarrer samba.

### Authentification SAMBA avec LDAP

Ajouter la section suivante dans /etc/samba/smb.conf :

```
;LDAP-specific settings
   ldap admin dn = "cn=Manager,dc=syroidmanor,dc=com"
   ldap server = localhost
   ldap port = 389
   ldap ssl = no
   ldap suffix = "ou=Users,dc=syroidmanor,dc=com"
```

En remplaçant naturellement les informations serveur données en exemple. Par contre pour être complet il faut utiliser les scripts IdealX de synchronisation des mots de passe avec le LDAP.

### interface avancée de partage réseau Windows

Vous pouvez modifier les paramètres de création des UID et GID des utilisateurs du Domaine. Cependant ce n'est utile que si vous utilisez un autre serveur Samba ( Unix, Linux, Mac OS X...) ou un autre NAS (NetApp...) et que vous souhaitez partager les données d'authentification avec ces derniers (par exemple pour des partages NFS).

### VSFTPd

Les vidéastes appellent très improprement "FTP passif" le FXP. Pour activer le mode FXP dans VSFTPd il faut mettre ceci dans le /etc/vsftpd.conf:

```
pasv_promiscuous=yes
port_promiscuous=yes
```

### iSCSI

### Côté initiateur (client)

Attention à l'alignement des volumes. Par défaut, la table de partitions MS-DOS introduit un décalage : quand on crée une table de partition sur un périphérique iSCSI, on peut décaler les blocs par rapport au périphérique sous-jacent, ce qui fait que chaque lecture ou écriture induit deux lectures ou deux écritures sur le disque physique, très mauvais pour les performances!

Pour aligner les volumes :

1. Avec un initiateur Linux, ne pas créer de table de partition sur un périphérique iSCSI est encore le mieux.
2. avec un initiateur windows, on est obligé de créer une table de partitions. Utiliser

la ligne de commande DISKPART :

```
C:\>diskpart
DISKPART> list disk
DISKPART> select disk 1
DISKPART> list partitions
DISKPART> create partition primary align=64
```

Il faut que l'offset (ici 64K, soit 65536) soit toujours un multiple de 4096, ou idéalement de la taille de stripe du périphérique physique utilisé.

### Côté cible (serveur)

Pour optimiser pour les IOPS, partir de cette configuration:

```
Target iqn.2012-06.test5:testing
        Lun 0 Path=/dev/md100,Type=blockio
        MaxConnections          8
        InitialR2T              No
        ImmediateData           Yes
        MaxRecvDataSegmentLength 65536
        MaxXmitDataSegmentLength 65536
        MaxBurstLength          1048576
        FirstBurstLength        262144
        MaxOutstandingR2T       1
        HeaderDigest            None
        DataDigest              None
        NOPInterval             60
        NOPTimeout              180
        Wthreads                8
        QueuedCommands          64
```

### iSCSI et VMWare

VMware a besoin des SCSI ID et SN pour identifier correctement les LUNs des différentes cibles. Vous pouvez forcer des valeurs manuelles par exemple si vous souhaitez mettre les cibles iSCSI en cluster, etc.

En cas de problème de déconnexion sous forte charge, il faut modifier les paramètres VMware. On peut faire ceci sur le serveur VM:

```
esxcfg-advcfg -s 14000 /VMFS3/HBTokenTimeout
```

Autre possibilité, modifier dans les paramètres avancés de l'initiateur iSCSI de VMware :

```
MaxCommands             1
```

### Serveur de courrier électronique intégré

Le serveur est exim4. Utiliser **dpkg-reconfigure exim4-config**.

# 5.Mise en grappe des serveurs

## OCFS2

Il suffit de faire

```
aptitude install ocfs2console
```

pour tout installer.

D'abord, configurer avec `dpkg-reconfigure ocfs2-tools` .

**Tres important** : il faut beaucoup augmenter le délai *heartbeat* "seuil de battement O2CB" : par défaut il est à 7; normalement une bonne valeur se situe entre 30 et 35.

Ne pas oublier de faire *service o2cb enable*.

Ensuite, configurer le cluster avec *ocfs2console* : d'abord configurer les nœuds ( *cluster -> configure nodes* ), ensuite configurer les systèmes de fichiers en les formatant si nécessaire. Faire le montage depuis ocfs2console, puis copier les lignes correspondantes de `/etc/mtab` dans `/etc/fstab` sur tous les nœuds.

L'erreur la plus courante avec OCFS2 c'est le time-out, dans ce cas il se peut que le OCFS2 *heartbeat* soit regle trop court sur une machine, verifier ( `/etc/default/o2cb` ), l'autre erreur classique etant le "out of memory" : avec 1Go par CPU c'est juste, très juste.

On peut agrandir un volume OCFS2 à la volée après l'avoir démonté sur tous les nœuds: on redimensionne la partition (avec *parted* ou autre) puis on agrandit le volume avec *tunefs.ocfs2 -S /dev/sdXX*. Ensuite on relit la table de partition sur tous les nœuds avec *blockdev --rereadpt /dev/sdXX*, après quoi on peut remonter le volume.

## Cluster haute disponibilité

1. bien identifier les deux nœuds.
2. mettre les machines en baie.
3. câbler les secteurs, claviers, souris.
4. câbler ensemble les baies SAN avec les câbles SAS.
5. câbler le FC :
   - si deux ports (pas chez Dubbing) c'est un port sur chaque contrôleur.
   - si un port c'est cluster01 -> contrôleur 0 (bas), cluster02 -> contrôleur 1 (haut)
6. câbler le réseau sur les NAS. Pas besoin sur les SAN.
7. allumer les machines dans l'ordre : baie SAN esclave, attente, baie SAN maître, attente, les têtes NAS.

## configuration de base

1. vérifier les alarmes storiq : ne monitorer les disques qu'une seule fois!
2. vérifier que chaque noeud voit son copain sur l'adresse spéciale "cluster"

(10.128.1.xx)
3. configurer le réseau "normal" en utilisant une interface virtuelle (bond0:0)
4. vérifier que chaque noeud voit le réseau "normal".

### configuration de heartbeat

1. activer le heartbeat avec chkconfig
2. mettre l'adresse IP virtuelle du cluster dans /etc/hosts sur les deux noeuds
3. mettre l'adresse IP virtuelle du cluster dans /etc/ha.d/haresources
4. modifier l'email dans haresources
5. configurer le smtp local pour qu'il transfère les mails (dpkg-reconfigure exim4)
6. démarrer le heartbeat d'abord sur cluster01 puis sur cluster02 *

### configuration des partages

1. faire tous les partages (NFS, samba, etc) sur cluster01
2. vérifier les partages.
3. vérifier que le volume de configuration partagé est bien monté.
4. ATTENTION samba et nfs ne doivent PAS démarrer au boot. Si vous voulez ajouter le FTP ou Netatalk de même.
5. les fichiers de configuration Samba, NFS sont sur le volume partagé */mnt/configdata*. Comme ça on garantit une configuration identique des deux côtés.
6. /var/lib/nfs, /var/lib/samba, /**var**/**run**/**samba,** /etc/samba et /etc/exports sont sur le volume partagé.

### test de la bascule haute-disponibilité

1. forcer la bascule en faisant "killall -9 heartbeat && service nfs stop && service samba stop && ifconfig bond0:1 down && umount -a -t xfs"
2. on peut aussi forcer la bascule en tirant brutalement sur la prise de courant mais heu...

### création de volumes

1. Il faut :
   - arrêter nfs et samba
   - arrêter ocfs2 et o2cb
   - démonter les volumes qui sont sur le SAN
   - faire rmmod qla2xxx && modprobe qla2xxx (adapter selon le protocole et le type de connexion)
   - refaire la liste dans l'autre sens (sauf qu'il ne faut pas redémarrer nfs et samba sur le noeud passif, hein ?)
2. Nota Bene : les volumes à monter par le *heartbeat* doivent être XFS et NOAUTO dans */etc/fstab*.

### Réplication DRBD
### configuration de base

Les *drbd tools* contiennent un fichier de conf d'exemple : */etc/drbd.d/global_common.conf*, utilisant **drbd** comme nom de ressource et */dev/drbd0* comme périphérique commun.

1. Modifier ce fichier pour définir les périphériques et les hôtes à synchroniser.
2. Modifier /etc/hosts pour que les machines soient bien identifiées
3. supprimer éventuellement la ligne `include "drbd.d/*.res";` dans */etc/drbd.conf*

### initialisation

1. démarrer le service sur les noeuds.
2. Vérifier le statut actif/passif dans /proc/drbd

Si nécessaire, créer la signature :

```
drbdadm  create-md drbd
```

Lancer la synchronisation:

```
drbdadm -- --overwrite-data-of-peer primary drbd
```

Vérifier l'avancement avec `cat /proc/drbd`

### Mode actif/actif

passer chaque noeud en primary avec

```
drbdadm primary all
```

vérifier le statut avec `cat /proc/drbd`

Utiliser le périphérique /dev/drbd0 normalement. Pour l'utiliser simultanément sur les deux noeuds, il faut employer OCFS2 (par exemple).

### Système parallèle PVFS2/OrangeFS

### configuration initiale de PVFS2

Tout d'abord il faut de préférence utiliser un fichier /etc/hosts commun aux noeuds du cluster, pour s'assurer que chaque machine connait bien toutes les autres. Ensuite après avoir installé les paquets *pvfs-base* et *pvfs-2.6.XX* procéder à la configuration sur un des noeuds avec *pvfs2-genconfig /etc/pvfs2-fs.conf*. Le cluster sera nommé *pvfs2-fs*. Répondez bien aux questions, et n'utilisez pas *localhost* comme nom de machine mais les noms réels.

Ensuite :

- copiez le fichier */etc/pvfs2-fs.conf* sur tous les noeuds du cluster
- initialisez sur chaque noeud l'espace de stockage : `pvfs2-server /etc/pvfs2-fs.conf -f`
- démarrez le service pvfs2-server normalement sur chaque noeud : `service pvfs2-server start`

### configuration client

Démarrez le service *pvfs2-client* avant le script *mountpvfs2.sh*. *mountpvfs2.sh* utilise /etc/pvfs2tab, dont la syntaxe est identique à celle de fstab :

```
tcp://melodica:3334/pvfs2-fs /mnt/pvfs2 pvfs2 defaults,noauto 0 0
```

## 6.Restauration depuis le rescue

booter sur le CD de préférence. pour réparer la partition système, faire

```
reiserfsck --fix-fixable -y /dev/sda1
```

S'il dit qu'il y a des erreurs nécessitant rebuild-tree, faire

```
reiserfsck --rebuild-tree /dev/sda1
```

- par contre il vaut sans doute mieux restaurer carrément depuis le "rescue" parce qu'il y a un gros risque de fichiers manquants :

```
mount /dev/sda1 /mnt/sda1
ls -l /mnt/sda1/lost+found
```

- S'il y a des fichiers dans lost+found, c'est qu'ils sont perdus, il vaut mieux restaurer le backup :

```
mount /dev/sda3 /mnt/sda3
```

  - vérifier que le "rescue" est à jour ( par exemple il doit dater du 2 mars):

```
ls -ld /mnt/sda3/rdiff-backup-data
```

S'il est bien à jour, tu peux restaurer à la version du 2 mars :

```
rdiff-backup -r "2010/3/2" /mnt/sda3/ mnt/sda1/
```

- Ensuite il vaut mieux réinstaller le bootloader au cas où:

```
umount /mnt/sda3
cd /mnt/sda1
chroot .
mount /proc
mount /mnt/rescue
lilo
umount -a
exit
```

- et pour finir :

```
reboot
```

# 7.Remplacement des disques contrôleurs 3Ware

Le remplacement des disques doit de préférence s'effectuer uniquement lorsque les unités RAID sont marquées "OPTIMAL". Attention : les raid_cli version 1.x et 2.x n'affichent pas toutes les erreurs de disque. Il faut faire

```
raid_cli info <contrôleur>
```

pour avoir toutes les infos, en particulier les status d'erreur de disques SMART-ERROR, DEVICE-ERROR. Marche à suivre :

- SMART-ERROR : remplacer préventivement le disque, RMA constructeur.
- DEVICE-ERROR : analyser les logs (*/var/log/messages*).

erreur timeout:

```
May 11 01:12:08 storiq-c1-n1 kernel: 3w-9xxx: scsi6: AEN: ERROR
(0x04:0x0009): Drive timeout detected:port=10.
```

On peut remplacer le disque, mais en général il ne présente pas de défaut lors des tests. En fait le firmware 3Ware en version antérieure à 4.10.07 n'efface pas les erreurs transitoires. Par contre, si le contrôleur est en 4.10.07, alors un "DEVICE-ERROR" est une vraie erreur permanente.

erreur "sector repair" :

```
May  7 12:45:15 storiq-c1-n1 kernel: 3w-9xxx: scsi6: AEN: WARNING
(0x04:0x0023): Sector repair completed:port=10, LBA=0xAFC.
```

On peut conserver le disque, mais il finira en SMART-ERROR tôt ou tard. Plutôt tôt d'ailleurs.

# 8.Gestion de l'alimentation

Il y a deux options : soit on utilise un UPS avec port série ou USB, soit un UPS réseau. Les UPS réseau malheureusement ne sont pas supportés par "nut".

## Onduleur série/USB en local

Pour les séries/USB on utilisera nut : `aptitude install nut`

On configure le fichier: `/etc/nut/ups.conf` :

```
[apc]
    driver = usbhid-ups
    port = auto
```

Le nom entre crochets est libre. Pour les drivers possibles et les ups supportés, voir : http://www.networkupstools.org/compat/stable.html Si l'onduleur est série, il faut que nut puisse y accéder, on modifiera les règles udev pour cela en créant `/etc/udev/rules.d/99_nut-serialups.rules`:

```
# /etc/udev/rules.d/99_nut-serialups.rules
KERNEL=="ttyS1", GROUP="nut"
```

Ensuite on applique les changements dans udev:

```
sudo udevadm control --reload_rules
sudo udevadm control trigger
```

puis on démarre nut :

```
$ sudo upsdrvctl start
```

qui répondra :

```
Network UPS Tools - UPS driver controller 2.2.2
Network UPS Tools: 0.29 USB communication driver - core 0.33 (2.2.2)

Using subdriver: APC HID 0.92
```

## Onduleur série/USB en réseau

Il faut aussi configurer upsd et upsmon. upsd communique avec le pilote; upsmon communique avec upsd et éteint la machine. Plusieurs upsmon et donc plusieurs machines peuvent être commandées depuis un seul upsd et un seul onduleur. Un upsd peut recevoir des messages de plusieurs onduleurs. Créer le fichier de configuration `/etc/nut/upsd.conf`:

```
# /etc/nut/upsd.conf
ACL all 0.0.0.0/0
ACL localhost 127.0.0.1/32
ACCEPT localhost
REJECT all
```

Dans cette configuration upsd n'accepte que les connexions du pilote local. Ensuite il faut remplir `/etc/nut/upsd.users`:

```
# /etc/nut/upsd.users
```

```
[local_mon]
    password = spider77
    allowfrom = localhost
    upsmon master
```

On peut utiliser plusieurs utilisateurs pour différentes machines. Ensuite on configure upsmon via `/etc/nut/upsmon.conf`:

```
# /etc/nut/upsmon.conf
MONITOR apc@localhost 1 local_mon spider77 master
POWERDOWNFLAG /etc/killpower
SHUTDOWNCMD "/sbin/shutdown -h now"
```

"apc" est le nom de l'onduleur indiqué dans `/etc/nut/ups.conf` et le mot de passe est celui donné dans `/etc/nut/upsd.users`. Ces fichiers doivent avoir des permissions restreintes :

```
$ sudo chown root:nut /etc/nut/*
$ sudo chmod 640 /etc/nut/*
```

Enfin il ne faut pas oublier d'activer nut au démarrage via `/etc/default/nut`:

```
# /etc/default/nut
START_UPSD=yes
START_UPSMON=yes
```

On démarre bien sûr avec

```
nut start
```

La commande suivante permet d'avoir un status de l'onduleur:

```
$ upsc apc
```

Attention, nut par défaut ne démarre l'extinction que quand l'onduleur est "critique", afin de ne pas couper sur une brève interruption.

### Onduleur APC avec interface réseau

Les onduleurs APC avec interface réseau utilisent snmp et doivent être supervisés avec `apcupsd`. Il faut une version 3.10 pour supporter ces onduleurs.

Pour configurer l'adresse IP de l'onduleur, on peut lui attribuer une adresse via arp ainsi :

```
arp -s <IPaddress> <MacAddress>
ping <IPaddress> -s 113
```

Ensuite on se connecte sur l'onduleur en telnet, appuyer sur entrée 4 ou 5 fois, utilisateur "apc" mot de passe "apc",puis on peut enregistrer la configuration:

```
boot -b manual
tcpip -i <addresse>
tcpip -s <masque>
tcpip -g <routeur>
logout
```

ou via le menu, selon le type d'onduleur. Ensuite on configure le démon `apcupsd` via `/etc/apcupsd.conf`:

```
DEVICE 192.168.100.2:161:APC:private
```

Où les directives sont:

- addresse IP de l'onduleur
- port: port SNMP distant, normalement 161
- type d'agent SNMP:
  - "APC" pour les APC PowerNet?
  - "MIB", "APC_NOTRAP" pour la MIB PowerNet? avec les traps SNMP désactivés. ( APC_NOTRAP nécessite apcupsd 3.12 ou supérieur).
- la communauté, normalement "private".

## 9. Économie d'énergie

installer cpufreq-utils ajouter dans /etc/modules powernow-k8 ( noyau < 3.7) ou acpi-cpufreq (noyau > 3.7) et redémarrer.

## 10.Virtualisation avec lib-virt et virt-manager

Le but de la manœuvre : créer une interface bridge comme interface par défaut, qui sera automatiquement utilisé par virt-manager.

On doit donc créer une interface br0 qui intègre de base le groupe bond0 (pour continuer à utiliser le bonding).

Il suffit de modifier le fichier `/etc/network/interfaces` comme ceci:

1. on remplace "bond0" par "br0" partout.
2. on ajoute une ligne "bridge_ports bond0" dans les informations de br0
3. on ajoute une entrée "iface bond0 inet manual"

Ce qui donne ceci (avec vos ip à vous):

```
auto lo br0
iface lo inet loopback

iface bond0 inet manual

iface br0 inet static
        address 10.0.1.9
        netmask 255.255.0.0
        network 10.0.1.0
        broadcast 10.0.1.255
        gateway 10.0.1.1
        # paramètre bridge obligatoire
        bridge_ports bond0
        # paramètres bridge optionnels
        bridge_fd 2
        bridge_maxwait 1
```

Le redémarrage du réseau ( service networking restart ) n'a pas suffit chez moi. J'ai du faire comme ceci :

```
service networking stop
bonding_cli stop bond0
bonding_cli start bond0
service networking start
```

Par précaution, vérifiez que ça fonctionne bien en redémarrant le serveur. Au démarrage, on doit bien avoir un bridge br0.

Dès qu'il y a un bridge actif, virt-manager va automatiquement l'utiliser et insérer des interfaces virtuelles comme nécessaire, ça marche tout seul.

Pour pouvoir lancer virt-manager avec un utilisateur normal (pas root), il faut ajouter l'utilisateur en question au groupe libvirt:

```
adduser <user> libvirt
```

À partir de là toute nouvelle session de l'utilisateur en question pourra lancer le virt-manager.

# 11.Surveillance et supervision

## SNMP

Le P.E.N. Intellique est 37990. La MIB SNMP intégrée à StorIQ 1.x et 2.x n'est pas valide et utilise un PEN de test. L'OID correct doit donc être "1.3.6.1.4.1.37990".

Pour sonder un service:

```
snmpwalk -c public -v1 d242
```

Pour sonder une MIB précise (particulièrement la notre...):

```
snmpwalk -v 1 -c public d242 .1.3.6.1.4.1.37990
```

Pour connaître la description des éléments de la MIB:

```
snmptranslate -IR -Tp -On storiq
```

Pour ajouter une MIB aux MIBS par défaut:

```
export MIBS=+STORIQ-TABLE-MIB
```