Software-RAID HOWTO

Jakob Ø ;stergaard (*jakob@ostenfeld.dk*)

v. 0.90.7 ; 19th of January 2000

Ce document décrit l'utilisation du RAID logiciel 0.90 sous Linux mis au point par Ingo Molnar et d'autres développeurs. Il s'agit de la couche RAID qui sera standard pour les versions 2.4.x du noyau Linux et qui est également disponible sous forme de patches pour la branche 2.2. La couche d'adaptation RAID 0.90 est disponible sous forme de patches pour les branches 2.0 et 2.2. De nombreuses personnes la considèrent comme bien plus robuste que la couche RAID courante. those kernels.

Contents

1 Introduction

Pour une description de la version antérieure de l'interface RAID standard dans les noyaux 2.0 et 2.2, reportez vous à l'excellent document de Linas Vepstas (*linas@linas.org*) disponible via le Linux Documentation Project à l'adresse *linuxdoc.org* .

http ://ostenfeld.dk/˜jakob/Software-RAID.HOWTO/ est la page de référence pour ce HOWTO où les dernières mises à jour seront disponibles. Ce document a été rédigé par Jakob Ostergaard à partir de nombreux messages électroniques échangés entre l'auteur et Ingo Molnar *(mingo@chiara.csoma.elte.hu)* - un des développeurs du RAID - , la liste de diffusion linux-raid *(linux-raid@vger.rutgers.edu)* ainsi que diverses personnes.

La rédaction de ce HOWTO a été motivée par le besoin de documentation du nouveau système RAID alors que le Software-RAID existant ne documentait que les versions précédentes. Des fonctionnalités nouvelles ont notamment été introduites.

Pour utiliser la nouvelle mouture du RAID avec les noyaux 2.0 ou 2.2, vous devez récupérer le patch correspondant, par exemple à l'adresse *ftp://ftp.[your-country-code].kernel.org/pub/linux/daemons/raid/alpha* , ou depuis *http://people.redhat.com/mingo/* . Les noyaux 2.2 officiels ne gèrent directement que l'ancien type de RAID et les patches sont donc nécessaires. *L'ancien système RAID des noyaux 2.0 et 2.2 est buggé.* De surcroît, il lui manque d'importantes fonctionnalitées présentes dans la nouvelle version.

La nouvelle mouture du RAID est en cours d'intégration dans les noyaux de développement 2.3.x et sera donc disponible dans la branche 2.4. Jusqu'à la sortie de celle-ci, il sera nécessaire de patcher manuellement les noyaux.

Peut-être esssayerez vous les versions `-ac` du noyau proposées par Alan Cox pour disposer du RAID. *Certaines* d'entre elles incluent le nouveau système et vous épargneront donc l'application de patches.

Le HOWTO contient des informations triviales pour ceux qui maîtrisent déjà les principes des systèmes RAID. Inutile de vous y attarder.

1.1 Avertissement

L'avertissement indispensable :

Bien que le fonctionnement du système RAID semble stable chez moi et chez de nombreuses personnes, cela pourrait ne pas être le cas pour vous. Si vous perdez vos données, votre emploi, votre femme ou que vous vous faites écraser par un camion, ce ne sera ni de ma faute, ni de celle des développeurs. Vous utilisez les fonctionnalités RAID, ainsi que toutes les informations contenues dans ce documenti, à vos risques et périls. Il n'y a pas la moindre garantie concernant le logiciel ou ce document ni la moindre assurance que cela puisse servir à quoi que ce soit. Sauvegardez toutes vos données avant la moindre manipulation. Il vaut mieux être prudent que désolé.

Ceci étant, je dois reconnaitre que je n'ai pas eu de problèmes de stabilité avec le RAID logiciel, que je l'employe sur quelques machines et que je n'ai entendu personne évoquer des plantages aléatoires ou des instabilités avec le RAID.

1.2 Prérequis

Le HOWTO suppose que vous utilisez un des derniers noyaux 2.2.x ou 2.0.x modifié par le patch raid0145 adéquat ainsi que la version 0.90 des raidtools ou que vous vous servez d'un 2.3 postérieur à la > 2.3.46, voire d'un 2.4. Les patches et les outils se trouvent par exemple à : *ftp://ftp.fi.kernel.org/pub/linux/daemons/raid/alpha* ou pour certains à l'adresse : *http://people.redhat.com/mingo/* . Les patches RAID, le paquetage des raidtools et le noyau doivent s'accorder autant que possible. Il sera peut-être parfois nécessaire de se restreindre à des noyaux plus anciens si les patches ne sont pas disponibles pour le dernièr noyau sorti.

2 Motivation du RAID

Il existe différentes bonnes raisons pour se servir du RAID parmis lesquelles figurent la possibilité de fusionner plusieurs disques physiques en un périphérique virtuel plus important, l'amélioration des performances et la redondance.

2.1 Aspects techniques

Le RAID Linux est adapté à la majeure partie des périphériques de type bloc. Peu importe que vous utilisiez de l'IDE, du SCSI ou un mélange des deux. Certains ont également obtenu quelques succès en s'en servant avec des périphériques de type bloc en réseau (Network Block Device ou NBD).

Vérifiez que les bus d'accès aux périphériques sont assez rapides. Il n'est pas conseillé d'installer 14 disques Ultra Wide sur une même chaine si chacun d'entre eux peut débiter 10 Mo/s car le bus, lui, ne dépassera pas les 40 Mo/s. Vous avez également intérêt à ne pas mettre plus d'un disque par interface IDE sans quoi les performances ne vont pas être fameuses. L'IDE n'est pas adapté pour l'accès simultané à plusieurs disques sur une même interface. Toutes les cartes mères récentes incluent deux ports et vous pourrez donc configurer deux disques en RAID sans acheter de contrôleur supplémentaire.

La couche RAID est indépendante du système de fichier. Vous pourrez donc y superposer celui de votre choix.

2.2 Vocabulaire

RAID sera employé pour "RAID logiciel Linux". Le document ne traite pas du RAID matériel.

Dans la description des configurations, on utilise fréquemment le nombre de disques et leur taille. **N** désignera le nombre de disques dans une matrice RAID, les disques de secours étant exclus, **S** sera la taille du plus petit disque et **P** le débit d'un disque en Mo/s. Quand on se servira de P, on supposera que les disques ont tous les mêmes performances (à vérifier).

Périphérique et disque seront synonymes. En général, les matrices RAID sont davantage construites avec des partitions qu'avec des disques complets. La combinaison de plusieurs partitions appartenant à un même disque ne présente guère d'intérêt et on entendra donc par périphérique et disque "des partitions sur différents disques".

2.3 Niveaux RAID

Voici une brève description des niveaux de RAID gérés par Linux. Certaines de ces infos sont basiques mais j'ai fait mention d'éléments spécifiques à la mise en oeuvre au sein de Linux. Sautez cette section si vous savez ce qui se cache derrière le RAID. Vous y reviendrez quand vous aurez des problèmes :o)

Les patches RAID pour Linux offrent les possibilités suivantes :

- **mode linéaire**
  - Deux disques et plus sont combinés par concaténation. L'écriture sur le disque RAID se fera donc d'abord sur le premier puis sur le second quand il sera plein et ainsi de suite. Il n'est pas nécessaire que les disques soient de la même taille et, pour tout dire, la taille n'est ici d'aucune importance.
  - Il n'y a aucune redondance à ce niveau. Si un disque tombe en panne, vous perdrez surement toutes vos données. Vous aurez peut être la chance d'en récupérer une partie si, du point de vue du système de fichiers, il ne manque qu'un gros bloc consécutif de données.
  - Les performances en lecture/écriture ne vont pas s'améliorer automatiquement mais si plusieurs utilisateurs se servent simultanément du périphérique, il se peut qu'ils accèdent à des disques différents et que les performances en soient augmentées.
- **RAID-0**

  - Ou "stripe". Semblable au mode linéaire à ceci près que les lectures et les écritures ont lieu en parallèle sur les disques. Les disques doivent avoir sensiblement la même taille. Les périphériques se remplissent progressivement de la même façon. Si l'un des deux est plus grand que l'autre, l'espace supplémentaire est toujours employé pour la matrice RAID mais vous n'utiliserez qu'un des deux disques vers la fin. Les performances en patiront.
  - Comme en linéaire, il n'y a pas de redondance des données mais en plus vous ne serez pas capable de récupérer vos données si un disque tombe en panne. Au lieu de ce qu'il vous manque un gros bloc de données, votre système de fichiers comprendra de nombreux petits trous. e2fsck ne sera vraisemblablement pas en mesure de reconstituer quoi que ce soit.
  - Les performances en lecture/écriture augmenteront puisque les lectures et les écritures auront lieu sur les deux disque en même temps. C'est souvent ce qui motive l'emploi du RAID-0. Si les bus sont assez rapides, vous pourrez flirter avec N*P Mo/s.

- **RAID-1**
  - Il s'agit du premier mode redondant. Le RAID-1 s'employe à partir de deux disques auxquels viennent éventuellement se greffer des disques de secours. Ce mode duplique les informations d'un disque sur l(es)'autre(s). Bien sûr, les disques doivent être de même taille. Si un disque est plus grand que les autres, la matrice sera de la taille du plus petit.
  - Jusqu'à N-1 disques otés (ou défectueux), les données restent intactes et si le contrôleur (SCSI, IDE, etc...) survit, la reconstruction sera immédiatement entamée sur un des disques de secours après détection de l'anomalie.
  - Les performances en écriture sont légèrement inférieures à celles d'un disque unique vu que les données doivent être écrites sur chaque disque de la matrice. Les performances en lecture sont *en général* bien plus mauvaises en raison de la mise en oeuvre au sein du code d'une stratégie d'équilibrage simpliste. Cependant, cette partie des sources a été revue pour le noyau 2.4.

- **RAID-4**
  - Ce niveau RAID n'est pas utilisé très souvent. Il s'employe à partir de trois disques et plus. Au lieu d'effectuer une copie des informations, on conserve la parité sur un disque et on écrit les données sur les autres comme on le ferait avec une matrice RAID-0. Un disque étant dédié à la parité, la taille de la matrice sera (N-1)*S ou S est la taille du plus petit disque. Comme en RAID-1, les disques doivent avoir la même taille sans quoi le S précédent correspondra à celui du plus petit disque.
  - Si un disque lache, l'information de parité permet de reconstruire toutes les données. Si deux disques lachent, toutes les données sont perdues.
  - On n'utilise pas beaucoup ce niveau en raison du stockage de la parité sur un disque unique. L'information doit être mise à jour à *chaque* fois qu'on écrit sur un des disques, ce qui constitue un goulot d'étranglement si le disque de parité n'est pas nettement plus rapide que les autres. Cependant, si vous avez beaucoup de petits disques lents et un disque très rapide, le RAID-4 peut s'avérer très utile.

- **RAID-5**
  - Il s'agit surement du mode le plus approprié quand on souhaite combiner un grand nombre de disques tout en conservant de la redondance. Le RAID-5 s'employe à partir de trois disques avec éventuellement des disques de secours. La matrice sera de taille (N-1)*S, comme en RAID-4. A la différence du RAID-4, l'information de parité est répartie équitablement entre les différents disques, évitant ainsi le goulot d'étranglement du RAID-4.

- Si un des disques tombe en panne les données restent intactes. La reconstruction peut commencer immédiatement si des disques de secours sont disponibles. Si deux disques rendent simultanément l'âme, toutes les données sont perdues. Le RAID-5 ne survit pas à la défaillance de plus d'un disque.
- Les performances en lecture/écriture s'améliorent mais il est difficile de prévoir de combien.

2.3.1 Disques de secours

Les disques de secours ne prenent pas part à la matrice RAID jusqu'à ce qu'un des disques de celle-ci tombe en panne. Quand un disque lache, il est marqué défectueux et la reconstruction est entamée sur le premier disque de secours disponible.

Les disques de secours renforcent donc la sécurité de systèmes RAID-5 qui peuvent être difficilement accessibles. Le système peut fonctionner pendant un certain temps aec un disque défectueux tant que le disque de secours assure la redondance.

Vous ne pouvez être sûr de la survie de votre système en cas de défaillance d'un disque. La couche RAID peut faire son travail mais les gestionnaires SCSI peuvent receller des erreurs, les composants IDE peuvent se bloquer et d'autres phénomènes peuvent se produire.

2.4 RAID et swap

Il n'y a aucune raison d'employer le RAID au dessus du swap pour en améliorer les performances. Le noyau se charge lui-même d'équilibrer le swap sur plusieurs périphériques si toutes les partitions ont la même priorité dans la fstab.

Un fichier fstab correct ressemble à ce qui suit :

```
/dev/sda2       swap           swap    defaults,pri=1   0 0
/dev/sdb2       swap           swap    defaults,pri=1   0 0
/dev/sdc2       swap           swap    defaults,pri=1   0 0
/dev/sdd2       swap           swap    defaults,pri=1   0 0
/dev/sde2       swap           swap    defaults,pri=1   0 0
/dev/sdf2       swap           swap    defaults,pri=1   0 0
/dev/sdg2       swap           swap    defaults,pri=1   0 0
```

Cette configuration permet à la machine de swaper en parallèle avec sept périphériques SCSI. Aucun besoin du RAID pour ça vu qu'il s'agit d'une fonctionnalité présente dans le noyau de longue date.

Le RAID s'emploie pour le swap à des fins de haute disponibilité. Si vous configurez un système pour démarrer sur un périphérique RAID-1, le système doit être capable de survivre à une défaillance de disque. Seulement si le système était en train de swaper sur le périphérique défectueux, vous allez surement avoir des problèmes. Swaper sur une matrice RAID-1 aide dans ce genre de situations.

Il y a eu beaucoup de discussions concernant la stabilité du swap sous une couche RAID logicielle. Le débat continue car il dépend fortement d'autres aspects du noyau. A la date de rédaction de ce document, il semble que swaper via le RAID soit parfaitement stable *à l'exception* des phases de reconstruction (i.e. lorsqu'un nouveau disque est inséré dans une matrice dégradée). La question ne se posera plus lorsque le noyau 2.4 sortira mais jusque là, à vous de pousser le système dans ses retranchements afin de savoir si la stabilité vous satisfait ou bien si vous ne vous servirez pas du RAID pour le swap.

Vous pouvez utiliser un fichier de swap sur un système de fichiers au dessus d'une couche RAID, activer le RAID pour le fichier lui même ou déclarer un périphérique RAID en tant que swap. A vous de voir. Comme d'habitude, le disque RAID apparaitra comme un périphérique de type bloc.

3 Aspects matériels

Cette section a trait à certaines problèmes matériels qui se posent lorsqu'on se sert du RAID logiciel.

3.1 Configuration IDE

Le RAID fonctionne avec des disques IDE. On peut d'ailleur obtenir d'excellentes performances. En fait, compte tenu du prix actuel des disques et des contrôleurs IDE, le choix de ce matériel est à examiner lors de la mise en place d'un système RAID.

- **Stabilité :** les caractéristiques de tenue des disques IDE ont jusqu'ici été moins bonnes que celles des disques SCSI. Aujourd'hui encore, la garantie des disques IDE se cantonne typiquement à un an tandis qu'elle est de trois à cinq ans pour les disques SCSI. Bien qu'il soit exagéré d'affirmer que les disques IDE ont une qualité intrinsèque moindre, il faut reconnaitre que *certains* disques auront tendance à tomber en panne plus souvent que leurs équivalents SCSI. Maintenant, la mécanique est la même pour le SCSI et l'IDE. Il faut juste rester conscient de ce que tous les disques lachent, un jour ou l'autre. Il suffit de s'y préparer.
- **Intégrité des données :**autrefois, il n'y avait aucune garantie que les données écrites sur le disques fussent bien celles qui avaient été émises sur le bus (pas de vérification de parité ni de code d'erreur, etc...). Les disques durs IDE conformes aux spécifications Ultra-DMA effectuent un calcul de vérification sur les données qu'ils reçoivent. Il devient donc fortement improbable que des données soient corrompues.
- **Performance :** je ne vais pas m'étendre sur cet aspect des disques IDE. En résumé :
  - les disques IDE sont rapides (12 Mo/s et plus)
  - le système IDE consomme plus de ressources CPU (mais qui s'en soucie ?)
  - n'utilisez qu'**un** disque IDE par adaptateur sans quoi les performances vont se dégrader.
- **Résistance aux pannes :** le gestionnaire IDE survit généralement à la défaillance d'un disque IDE. La couche RAID étiquetera le disque comme défectueux et si vous employez du RAID-1 et au delà, la machine devrait continuer à fonctionner normalement jusqu'à ce que vous l'arrêtiez pour les opérations de maintenance.

Il est **très** important que vous n'utilisiez **qu'un** disque IDE par nappe. Outre la question des performances, la défaillance d'un disque provoque généralement le blocage de l'interface. Avec une configuration RAID qui supporte les défaillances (RAID-1, 4, 5), la panne d'un disque est supportée mais l'arrêt simultané de deux disques bloque la matrice. Un bus, un disque, telle est la règle.

Les contrôleurs PCI IDE ne manquent pas et vous pourrez vous procurer deux à quatre bus supplémentaires autour de 600 FF. Au vu de la différence de prix entre les disques SCSI et le disques IDE, je dirais qu'un système RAID IDE est des plus attractifs si on est prêt à se cantonner à un nombre de disques relativement peu important (autour de 8 à moins que l'on ne dispose de suffisamment de connecteurs PCI).

L'IDE est limité par la longueur des cables lorsqu'il s'agit de mettre en oeuvre des matrices importantes. Même si votre machine comprend suffisament de connecteurs PCI il est peu probable que vous puissiez loger plus de huit disques sans vous heurter à des problèmes de corruption des données dus à la longueur des cables.

3.2 Ajout et suppression de disque à chaud :

Le sujet a effectivement chauffé la liste de diffusion linux-kernel il y a quelques temps. Bien que la fonctionnalité soit présente dans une certaine mesure, il ne s'agit pas de quelque chose de facile.

3.2.1 Disques IDE

N'essayez pas de manipuler à chaud vos disques IDE ! L'IDE n'est pas prévu pour. Bien sûr, ça se passera peut-être correctement chez vous si le gestionnaire IDE est compilé en tant que module (vous utilisez donc un noyau 2.2 et au delà) et que vous le rechargez une fois le disque remplacé mais vous pouvez tout aussi bien vous retrouver avec un controleur IDE grillé. Le temps d'arrêt du système en cas de problème n'aura alors pas grand chose à voir avec celui d'une maintenance programmée.

Outre les aspects purement électriques qui détruiront joyeusement votre matériel, le problème réside en ce que l'interface IDE doit être réexaminée après que des disques soient échangés. Le gestionnaire IDE actuel ne le permet pas. Si le nouveau disque est rigoureusement identique à l'ancien, il se *peut* que cela fonctionne sans nouvel examen du bus mais, franchement, vous êtes en train de tenter le diable.

3.2.2 Disques SCSI

Le matériel SCSI n'est pas davantage prévu pour. Ca *peut* néanmoins fonctionner. Si votre contrôleur SCSI est capable de réexaminer le bus, d'autoriser l'ajout et la suppression de disques, vous y arriverez peut-être. Je ne vous le conseille vraiment pas mais ça peut fonctionner. Griller un disque neuf pousse parfois à revoir ses façons de faire...

La couche SCSI **devrait** supporter la défaillance d'un disque mais tous les gestionnaires SCSI n'en sont pas capables. Si le pilote SCSI accompagne le disque défectueux, pouvoir échanger ce dernier à chaud est inutile.

3.2.3 SCA

L'échange à chaud doit être possible mais je ne dispose pas du matériel nécessaire pour le vérifier et personne ne m'a fait part d'expériences de ce type. Je ne peux donc pas vous en dire plus.

Si vous voulez essayer, il vous faudra connaitre le fonctionnement interne du SCSI et du RAID. Je ne vais pas écrire quelque chose que je ne peux pas vérifier mais juste vous donner quelques indications :

- partez à la recherche de **remove-single-device** dans le fichier **linux/drivers/scsi/scsi.c**
- jetez un oeil à **raidhotremove** et à **raidhotadd**

Tous les gestionnaires SCSI ne permettent pas l'ajout et la suppression à chaud. Dans la série 2.2.x des noyaux, les pilotes Adaptec 2940 et Symbios NCR53C8xx en semblent capables. Toute information concernant les autres pilotes sera la bienvenue.

4 Configuration du RAID

4.1 Configuration générale

Voici ce que requièrent tous les niveaux de RAID :

- Un noyau, de préférence un 2.2.x ou le dernier 2.0.x. Si la branche 2.4.x est disponible quand vous lirez ces lignes, servez vous en.

- Les patches RAID. Ils existent généralement pour les noyaux récents. Les noyaux 2.4.x ne nécessiteront pas de patch.
- Les utilitaires RAID.
- De la patience, des pizzas et des amph^H^H^H^H substances à la caféine.

Tous les logiciels se trouvent sur `ftp://ftp.fi.kernel.org/pub/linux` Les outils et les patches RAID sont dans le répertoire `daemons/raid/alpha`. Le noyau se trouve dans le répertoire `kernel`.

Patchez le noyau, configurez le de façon à inclure la gestion du RAID pour les niveaux qui vous intéressent. Compilez et installez.

Détarrez, configurez, compilez et installez les outils RAID.

Jusqu'ici, tout va bien. A présent, si vous redémarrez, vous devriez avoir un fichier appelé `/proc/mdstat`. N'oubliez jamais que ce fichier est votre allié. Examinez son contenu avec un `cat /proc/mdstat`. Il devrait vous confirmer que vous disposez du niveau (personality) RAID voulu et qu'aucun périphérique RAID n'est actif.

Créez les partitions que vous souhaitez inclure dans votre matrice RAID.

La suite des opérations dépend à présent du mode RAID.

4.2 Mode linéaire

On dispose à présent de deux partitions (ou plus) qui ne sont pas nécessairement de la même taille et que l'on va concaténer.

Editez le fichier `/etc/raidtab` de façon à correspondre à votre configuration. Pour deux disques en mode linéaire, voici un fichier type :

```
raiddev /dev/md0
        raid-level      linear
        nr-raid-disks   2
        chunk-size      32
        persistent-superblock 1
        device          /dev/sdb6
        raid-disk       0
        device          /dev/sdc5
        raid-disk       1
```

On ne peut disposer de disques de secours. Si un disque tombe en panne, toute la matrice s'effondre. Il n'y a rien à stocker sur un disque de secours.

Vous vous demanderez peut-être pourquoi on précise un paramètre `chunk-size` quand le mode linéaire ne fait que concaténer les disques en un disque virtuel plus important sans y accéder en parallèle. Vous avez tout à fait raison. Mettez y une valeur quelconque et pensez à autre chose.

On crée la matrice :

```
  mkraid /dev/md0
```

La commande initialise la matrice, écrit les superblocs persistants et active le périphérique.

Jetez un oeil à `/proc/mdstat`. Vous devriez y voir que la matrice fonctionne.

A présent créez un système de fichiers comme sur un périphérique quelconque, montez le, incluez le dans votre fstab etc...

4.3 RAID-0

On dispose de deux disques (ou davantage) de taille sensiblement égale dont on veut additionner les capacités de stockage tout en en améliorant les performances au moyen d'accès simultanés.

Editez le fichier `/etc/raidtab` de façon à correspondre à votre configuration. Voici un fichier type :

```
raiddev /dev/md0
        raid-level      0
        nr-raid-disks   2
        persistent-superblock 1
        chunk-size     4
        device          /dev/sdb6
        raid-disk       0
        device          /dev/sdc5
        raid-disk       1
```

Comme en mode linéaire, il n'y a pas de disque de secours. Le RAID-0 n'offre aucune redondance et la défaillance d'un disque signifie celle de la matrice entière.

On exécute :

```
  mkraid /dev/md0
```

La commande initialise la matrice, écrit les superblocs persistants et active la matrice.

/dev/md0 est prêt à être formaté, monté et à subir les pires outrages.

4.4 RAID-1

On dispose de deux disques de taille sensiblement égale que l'on souhaite mettre en mirroir. On peut avoir des disques supplémentaires que l'on gardera en attente comme disques de secours et qui prendront automatiquement place dans la matrice si un disque actif tombe en panne.

Voici le fichier `/etc/raidtab` typique :

```
raiddev /dev/md0
        raid-level      1
        nr-raid-disks   2
        nr-spare-disks  0
        chunk-size     4
        persistent-superblock 1
        device          /dev/sdb6
        raid-disk       0
        device          /dev/sdc5
        raid-disk       1
```

Pour prendre en compte des disques de secours :

```
        device          /dev/sdd5
        spare-disk      0
```

N'oubliez pas d'ajuster la variable nr-spare-disks en conséquence.

A présent, on peut initialiser la matrice RAID. Son contenu doit être construit et les contenus des deux disques (sans importance pour l'instant) synchronisés.

Exécutez :

```
mkraid /dev/md0
```

L'initialisation de la matrice démarrera.

Examinez le fichier `/proc/mdstat`. On doit y lire que /dev/md0 a été démarré, que le mirroir est en cours de reconstruction et y trouver une estimation de la durée de reconstruction.

La reconstruction a lieu durant les périodes d'inactivité au niveau des entrées/sorties. L'interactivité du système ne devrait donc pas en souffrir. Les LED des disques palpiteront gaiement.

Le processus de reconstruction est transparent et on peut utiliser le périphérique RAID pendant cette phase.

Formattez la matrice pendant la reconstruction. On peut également la monter et s'en servir. Bien sûr, si le mauvais disque lache à ce moment là, il ne s'agissait pas d'un jour de chance.

4.5 RAID-4

Remarque : je n'ai pas testé personnellement cette configuration et ce qui suit correspond à ce qui me paraît le plus vraisemblable.

On dispose de trois disques ou plus de taille sensiblement équivalente, l'un d'eux est nettement plus rapide que les autres et on souhaite les combiner en un périphérique de taille plus élevée tout en conservant un certain niveau de redondance. En outre, on peut introduire des disques de secours.

Fichier /etc/raidtab typique :

```
raiddev /dev/md0
        raid-level      4
        nr-raid-disks   4
        nr-spare-disks  0
        persistent-superblock 1
        chunk-size      32
        device          /dev/sdb1
        raid-disk       0
        device          /dev/sdc1
        raid-disk       1
        device          /dev/sdd1
        raid-disk       2
        device          /dev/sde1
        raid-disk       3
```

Les disques de secours sont traités par les lignes suivantes :

```
        device         /dev/sdf1
        spare-disk     0
```

La matrice s'initialise comme d'habitude :

```
mkraid /dev/md0
```

On se reportera aux options particulières de mke2fs avant de formater le périphérique.

4.6 RAID-5

On dispose de trois disques ou plus de taille sensiblement équivalente que l'on veut combiner en un périphérique de taille plus élevée tout en assurant la redondance des données. On peut introduire des disques de secours.

Si on employe N disques dont le plus petit est de taille S, la taille de la matrice sera (N-1)*S. L'espace manquant sert au stockage des données de parité (redondance). Si un disque tombe en panne, les données restent intactes. Si deux disques lachent, toutes les données sont perdues.

Fichier de configuration /etc/raidtab typique :

```
raiddev /dev/md0
        raid-level      5
        nr-raid-disks   7
        nr-spare-disks  0
        persistent-superblock 1
        parity-algorithm        left-symmetric
        chunk-size      32
        device          /dev/sda3
        raid-disk       0
        device          /dev/sdb1
        raid-disk       1
        device          /dev/sdc1
        raid-disk       2
        device          /dev/sdd1
        raid-disk       3
        device          /dev/sde1
        raid-disk       4
        device          /dev/sdf1
        raid-disk       5
        device          /dev/sdg1
        raid-disk       6
```

Les disques de secours sont traités par les lignes suivantes :

```
        device         /dev/sdh1
        spare-disk     0
```

Et ainsi de suite.

Une taille de bloc (chunk-size) de 32 ko est un bon choix par défaut pour de nombreux systèmes de fichiers. La matrice dérivée du fichier de configuration précédent est de 7 fois 6 Go soit 36 Go (n'oubliez pas que (n-1)*s = (7-1)*6 = 36). Il contient un système de fichiers ext2 avec une taille de blocs de 4 ko. Rien n'empèche d'aller au-delà via les paramètres de bloc de la matrice et du système de fichiers si ce dernier est plus grand ou s'il doit contenir des fichiers de grande taille.

A présent, on exécute :

```
  mkraid /dev/md0
```

Normalement les disques devraient s'activer furieusement durant la construction de la matrice. On examinera le contenu du fichier `/proc/mdstat` pour savoir ce qui se passe.

Si le périphérique a été créé avec succès, la reconstruction est en cours. La matrice ne sera pas cohérente tant que celle-ci n'aura pas pris fin. La matrice est cependant parfaitement opérationnelle (à la gestion des défaillances près) et on peut déjà la formater et s'en servir.

On se reportera aux options particulières de mke2fs avant de formatter le périphérique.

Maintenant que le disque RAID fonctionne, on peut l'arrêter ou le redémarrer via les commandes :

```
raidstop /dev/md0
```

et

```
raidstart /dev/md0
```

Au lieu de mettre ces commandes dans les scripts d'initialisation et de rebooter un milliard de fois pour arriver à tout faire fonctionner, on lira les paragraphes suivants qui traitent de l'autodétection.

4.7 Les superblocs persistants

Autrefois (TM), les utilitaires RAID analysaient le fichier de configuration et initialisaient la matrice. Il fallait donc que le système de fichiers sur lequel figurait le fichier `/etc/raidtab` soit monté : plutôt pénible pour démarrer sur un système de fichiers RAID.

L'ancienne approche conduisait de surcroît à des complications pour monter des systèmes de fichiers reposant sur périphériques RAID. Ils ne pouvaient être simplement mis dans le fichier `/etc/fstab` habituel et nécessitaient des interventions chirurgicales dans les scripts de démarrage.

Les superblocs persistants résolvent ces problèmes. Lorsqu'une matrice est initialisée avec l'option `persistent-superblock` dans le fichier `/etc/raidtab`, un superbloc de type particulier est écrit au début de chaque disque prenant part à la matrice. Le noyau est alors capable d'obtenir directement la configuration de la matrice depuis les disques qui la composent au lieu de devoir analyser un fichier de configuration à la disponibilité aléatoire.

On gardera quand même cohérent le fichier `/etc/raidtab` puisqu'on peut en avoir besoin ultérieurement en cas de reconstruction de la matrice.

Les superblocs persistants sont obligatoires si on souhaite bénéficier de l'auto-détection des périphériques RAID au démarrage du système. On se reportera à la section **Autodétection**.

4.8 Taille des blocs (chunk size)

Ce paramètre mérite quelques explications. On ne peut jamais écrire de façon rigoureusement parallèle sur un ensemble de disques. Dans le cas de deux disques sur lesquels on devrait écrire un octet, on pourrait souhaiter que les quatres bits de poids fort aillent toujours sur le même disque, ceux de poids faible allant sur l'autre. Le matériel ne le permet pas. On définit donc de façon plus ou moins arbitraire une taille de bloc élémentaire qui correspondra à la plus petite quantité de données "atomique" qui sera écrite sur les disques. L'écriture de 16 ko avec une taille de bloc de 4 ko provoquera l'envoi du premier et du troisième bloc de 4 ko vers le premier disque et celui du deuxième et du quatrième bloc vers le second disque pour une matrice RAID-0 comportant deux disques. Pour de grosses écritures, la consommation de ressources sera minimisée par une taille de blocs importante tandis qu'une matrice composée essentiellement de petits fichiers profitera davantage d'une taille de blocs réduite.

Ce paramètre peut être spécifié pour tous les niveaux de RAID, même le mode linéaire où il n'a aucun effet.

A vous de modifier ce paramètre, ainsi que la taille de blocs du système de fichier, pour obtenir les meilleurs performances possibles.

L'argument de l'option chunk-size dans le fichier `/etc/raidtab` précise la taille en ko.

4.8.1 RAID-0

Les données sont écrites successivement sur chaque disque par paquets de `chunk-size` octets.

Pour une taille de bloc de 4 ko, lors de l'écriture de 16 ko de données sur un système muni de trois disques, la couche RAID écrira simultanément 4 ko sur chacun des trois disques puis écrira les 4 ko restant sur le disque 0.

32 ko semble un bon point de départ pour une majorité de matrices mais la valeur optimale dépend étroitement du nombre de disques impliqués, du contenu du système de fichiers et de divers autres facteurs. A vous d'expérimenter pour trouver la meilleure valeur.

4.8.2 RAID-1

Pour les écritures le paramètre importe peu vu que les données doivent être écrites sur tous les disques. Cependant, pour les lectures, il fixe la quantité de données à lire en une fois depuis un disque. Tous les disques contenant la même information, les lectures peuvent être équilibrées d'une façon similaire au RAID-0.

4.8.3 RAID-4

Lors d'une écriture dans une matrice RAID-4, l'information de parité doit être mise à jour sur le disque dédié. La taille de bloc spécifie alors la taille des blocs de parité. Si un octet est écrit dans une matrice RAID-4, `chunk-size` octets seront lus depuis N-1 disques, la parité sera calculée et `chunk-size` octets seront écrits sur le disque de parité.

Le paramète affecte les performances de la même façon que pour le RAID-0 puisque les lectures fonctionnent de la même façon.

4.8.4 RAID-5

Le paramètre a la même signification que pour le RAID-4.

128 ko est une valeur raisonnable. A vous de l'ajuster.

On se reportera également à la section traitant des options particulières de mke2fs qui affectent les performances du RAID-5.

4.9 Options de mke2fs

L'option `-R stride=nn` permet à mke2fs d'optimiser l'emplacement des structures de contrôle spécifiques à ext2 lors du formatage d'un disque RAID-4 ou RAID-5.

Si la taille de bloc RAID est de 32 ko, 32 ko de données consécutives résideront sur un même disque. Si on souhaite construire un systéme de fichiers ext2 avec une taille de blocs de 4 ko, il y aura 8 blocs de données consécutifs dans un bloc du tableau. On fournit l'information à mke2fs de la manière suivante :

```
mke2fs -b 4096 -R stride=8 /dev/md0
```

Les performances du RAID-{4,5} dépendent fortement de cette option. Je ne suis pas sûr de son influence sur les autres niveaux RAID. Si quelqu'un a des informations à ce sujet, elles seront les bienvenues.

La taille de bloc ext2 joue `très` fortement sur les performances du système de fichier. Dès que la taille de ce dernier dépasse quelques centaines de Mo, une taille de bloc de 4 ko est conseillée (à moins que le système de fichiers ne doivent stocker de très nombreux petits fichiers).

4.10 Autodétection

L'autodétection permet une reconnaissance automatique des périphériques par le noyau au démarrage, jsute après l'identification des partitions usuelles.

Requis :

1. La gestion de l'autodétection par le noyau.
2. Les disques RAID doivent avoir été créés avec l'option de persistance des superblocs.
3. Les partitions doivent être de type **0xFD** (à positionner avec fdisk).

Remarque : on vérifiera que la matrice RAID est arrêtée avant de changer le type des partitions (`raidstop /dev/md0`).

En suivant les trois étapes précédentes, l'autodétection devrait être en place. Essayez de redémarrer. Une fois le système initialisé, `/proc/mdstat` doit confirmer que la matrice RAID fonctionne.

Des messages semblables aux suivants apparaitront au démarrage :

```
Oct 22 00:51:59 malthe kernel: SCSI device sdg: hdwr sector= 512
 bytes. Sectors= 12657717 [6180 MB] [6.2 GB]
Oct 22 00:51:59 malthe kernel: Partition check:
Oct 22 00:51:59 malthe kernel:  sda: sda1 sda2 sda3 sda4
Oct 22 00:51:59 malthe kernel:  sdb: sdb1 sdb2
Oct 22 00:51:59 malthe kernel:  sdc: sdc1 sdc2
Oct 22 00:51:59 malthe kernel:  sdd: sdd1 sdd2
Oct 22 00:51:59 malthe kernel:  sde: sde1 sde2
Oct 22 00:51:59 malthe kernel:  sdf: sdf1 sdf2
Oct 22 00:51:59 malthe kernel:  sdg: sdg1 sdg2
Oct 22 00:51:59 malthe kernel: autodetecting RAID arrays
Oct 22 00:51:59 malthe kernel: (read) sdb1's sb offset: 6199872
Oct 22 00:51:59 malthe kernel: bind<sdb1,1>
Oct 22 00:51:59 malthe kernel: (read) sdc1's sb offset: 6199872
Oct 22 00:51:59 malthe kernel: bind<sdc1,2>
Oct 22 00:51:59 malthe kernel: (read) sdd1's sb offset: 6199872
Oct 22 00:51:59 malthe kernel: bind<sdd1,3>
Oct 22 00:51:59 malthe kernel: (read) sde1's sb offset: 6199872
Oct 22 00:51:59 malthe kernel: bind<sde1,4>
Oct 22 00:51:59 malthe kernel: (read) sdf1's sb offset: 6205376
Oct 22 00:51:59 malthe kernel: bind<sdf1,5>
Oct 22 00:51:59 malthe kernel: (read) sdg1's sb offset: 6205376
Oct 22 00:51:59 malthe kernel: bind<sdg1,6>
Oct 22 00:51:59 malthe kernel: autorunning md0
Oct 22 00:51:59 malthe kernel: running: <sdg1><sdf1><sde1><sdd1><sdc1><sdb1>
Oct 22 00:51:59 malthe kernel: now!
```

```
Oct 22 00:51:59 malthe kernel: md: md0: raid array is not clean --
 starting background reconstruction
```

Il s'agit des messages à l'autodétection des partitions d'une matrice RAID-5 qui n'a pas été arrêtée correctement (la machine a planté). La reconstruction a lieu spontanément. Le montage de l'unité est parfaitement licite puisque la reconstruction est transparente et que toutes les données sont cohérentes (seule l'information de parité qui ne sert qu'en cas de défaillance d'un disque est incohérente).

Les périphériques reconnus automatiquement sont stoppés de même quand le système s'arrête. Oubliez les scripts d'initialisation et servez vous des disques /dev/md comme s'il s'agissait de /dev/sd ou /dev/hd.

C'est aussi simple que ça.

Les lignes comportant les commandes raidstart et raidstop dans les scripts d'initialisation ne servent que pour les matrices RAID qui reposent sur l'ancienne mouture du code. Elles peuvent être supprimées sans hésitation dans le cadre de matrices RAID qui ont recours à l'autodétection.

4.11 Démarrage sur un disque RAID

Il existe plusieurs façons de mettre en place un système qui monte directement sa partition racine depuis un périphérique RAID. Pour l'instant, seuls les outils d'installation graphiques de la RedHat 6.1 permettent l'installation directe sur une matrice RAID. Il vous faudra donc surement effectuer quelques manipulations à la main mais il n'y a là rien d'impossible.

La dernière version officielle de lilo (21) ne gère pas les disques RAID et le noyau ne peut donc pas être chargé au démarrage depuis ce genre de périphériques. Il faudra donc que le répertoire `/boot` réside sur un système de fichier hors RAID. Afin d'être sûr que le système démarre quel que soit son état, dupliquez une partition `/boot` similaire sur chaque disque. Le BIOS sera ainsi toujours capable de charger les données depuis, par exemple le premier disque disponible. Il faudra donc que le système ne démarre pas avec un disque défectueux.

Avec la RedHat 6.1 est fourni un patch pour lilo 21 qui permet d'accéder à `/boot` sur du RAID-1. On notera que le patch n'est pas adapté aux autres niveaux RAID. Le patch est disponible dans tous les mirroirs RedHat via `dist/redhat-6.1/SRPMS/SRPMS/lilo-0.21-10.src.rpm`. La version modifiée de lilo acceptera un argument du type `boot=/dev/md0` dans le fichier `/etc/lilo.conf` et rendra chaque disque du mirroir utilisable au dénarrage.

On peut également avoir recours à une disquette de démarrage.

4.12 Installer le système de fichiers racine sur une couche RAID

Deux méthodes sont fournies ci-dessous. A ma connaissance, aucune distribution ne permet l'installation sur un disque RAID et la méthode que je suggère suppose que vous installez d'abord le système sur une partition normale avant de mouvoir les fichiers sur la matrice RAID une fois l'installation complète.

4.12.1 Première méthode :

On dispose d'un disque supplémentaire où on peut installer le système.

- Installez un système normal sur le disque supplémentaire.
- Mettez en place un noyau incluant les patches nécessaires pour le RAID et vérifiez que le système s'initialise correctement avec ce noyau. Il faudra veiller à ce que le support RAID soit **dans** le noyau et non sous forme de modules.

- A présent, configurez et créez la matrice RAID dont vous comptez vous servir en tant que racine. Il s'agit de la procédure standard telle que décrite précédemment dans le document.
- Redémarrez le système afin de vérifier que la matrice est détectée correctement (elle devrait en tout cas).
- Créez un système de fichier sur la nouvelle matrice avec mke2fs et montez la en /mnt/newroot (par exemple).
- Copiez le contenu de la racine courante sur le disque RAID. Il existe différentes façons de faire, par exemple :

  ```
  cd /
  find . -xdev | cpio -pm /mnt/newroot
  ```

- modifiez le fichier `/mnt/newroot/etc/fstab` de façon à ce qu'il pointe vers le périphérique `/dev/md?` adéquat pour la racine.
- Démontez le répertoire `/boot` courant et montez le à la place en `/mnt/newroot/boot`.
- Mettez à jour `/mnt/newroot/etc/lilo.conf` de façon à pointer vers le bon périphérique. Le périphérique de boot doit rester un disque normal (non-RAID) mais le disque racine doit pointer vers la matrice RAID. Ceci fait, exécutez un

  ```
  lilo -r /mnt/newroot
  ```

  . Lilo ne devrait pas émettre d'erreurs.
- Redémarrez le système et admirez avec quel facilité tout se passe comme on le souhaite :o)

Dans le cas de disques IDE, on spécifiera dans le BIOS les disques comme étant de type "auto-detect" pour que la machine puisse redémarrer même si un disque manque.

4.12.2 Seconde méthode :

Cette méthode nécessite l'emploi d'outils RAID et du patch qui autorisent la directive failed-disk. Il faut donc disposer d'un noyau 2.2.10 ou au delà.

Il **faut** que la matrice soit au moins de type 1. L'idée consiste à installer le système sur un disque marqué défectueux du point de vue RAID puis à copier le système sur la partie restante de la matrice RAID qui sera considérée comme dégradée avant de réinsérer le disque d'installation et de déclencher sa resynchronisation.

- Installez un système normal sur un des deux disques (qui fera plus tard partie de la matrice). Il est important que ce disque ne soit pas le plus petit sans quoi il ne sera pas possible de l'ajouter à la matrice !
- Récupérez le noyau, le patch, les outils, etc... Redémarrez le système avec le noyau qui est muni de la gestion RAID.
- Créez votre matrice en indiquant le disque qui occupe la racine actuelle comme `failed-disk` dans le fichier `raidtab` Ne mettez pas ce disque en première position dans le fichier ou vous aurez du mal à démarrer la matrice. Activez la matrice et mettez y un système de fichiers.
- Redémarrez et vérifiez que la matrice RAID est correctement activée.

- Copiez les fichiers de la racine et modifiez les fichiers système du disque RAID de façon à ce qu'il se référence bien en tant que racine.
- Lorsque le système démarre correctement depuis le disque RAID, modifiez le fichier `raidtab` en emplaçant la directive `failed-disk` par une directive `raid-disk`. Ajoutez à présent ce disque à la matrice avec `raidhotadd`
- Le système doit à présent démarrer depuis une matrice non-dégradée.

4.13 Démarrer le système depuis le RAID

Pour que le noyau soit capable de monter le système de fichiers racine, les pilotes des périphériques nécessaires doivent être présents dans le noyau (NdT : ou chargés via un initrd qui peut également contenir les modules RAID).

La façon normale de procéder consiste à compiler un noyau qui inclut en dur toutes les options RAID nécessaires (NdT : je proteste !).

La redHat-6.0 étant fournie avec un noyau modulaire qui gère la nouvelle mouture du RAID, je vais cependant en décrire l'emploi si on souhaite s'en servir pour démarrer son systéme depuis un volume RAID.

4.13.1 Démarrage avec le RAID modularisé

Il faut préciser à lilo qu'il doit également charger un équivalent de ramdisk en sus du noyau au démarrage. La commande `mkinitrd` permet de créer un ramdisk (ici un initrd) contenant les modules nécessaires au montage de la racine. Commande type :

```
mkinitrd --with=<module> <ramdisk name> <kernel>
```

Par exemple :

```
mkinitrd --with=raid5 raid-ramdisk 2.2.5-22
```

Ceci garantit que le module RAID adéquat sera disponible au démarrage lorsque le noyau devra monter la racine.

4.14 Mises en garde

Ne repartitionnez **JAMAIS** un disque qui appartient à une matrice RAID. Si vous devez modifier la table des partitions d'un disque au sein d'une matrice, arrêtez d'abord la matrice et repartitionnez ensuite.

On a vite fait de saturer un bus. Un bus Fast-Wide SCSI courant n'offre que 10 Mo/s, ce qui est largement en dessous des performances des disques actuels. Mettre six disques sur un canal de ce type n'apportera bien entendu pas le gain en performances souhaité.

L'ajout de contrôleurs SCSI n'est susceptible d'améliorer les performances que si les bus déjà présents sont proches de la saturation. Vous ne tirerez rien de plus de deux contrôleurs 2940 si vous n'avez que deux vieux disques SCSI qui ne satureraient même pas un seul contrôleur.

Si vous omettez l'option de persistance des superblocs votre matrice ne redémarrera pas spontanément après un arrêt. Reprenez la création de la matrice avec l'option correctement positionnée.

Si la resynchronisation d'une matrice RAID-5 échoue après qu'un disque ait été oté puis reinséré, l'ordre des disques dans le fichier raidtab est peut-être le responsable. Essayez de déplacer la première paire "device ..." et "raid-disk ..." en début de description de la matrice.

La plupart des retours d'erreur observés sur la liste de diffusion linux-kernel proviennent de gens qui ont procédé à des mélanges douteux entre les patches et les outils RAID. Si vous utilisez le RAID 0.90, vérifiez que vous vous servez bien de la bonne version des utilitaires.

5 Test de la couche RAID

Si vous utilisez le RAID pour améliorer la tolérance aux pannes, vous voudrez surement tester votre installation afin de vérifier son fonctionnement. Comment simule-t-on donc une défaillance ?

En résumé, on ne peut pas à moins de titiller un disque au lance-flammes pour "simuler" une défaillance. On ne peut pas prévoir ce qui va se passer en cas de perte d'un disque. Il pourrait très bien verouiller électriquement le bus et rendre tous les disques sur le bus inaccessibles. Je n'ai néanmoins jamais entendu d'histoire de ce genre. Le disque signalera peut-être une erreur de lecture/écriture à la couche IDE ou SCSI qui permettra à son tour à la couche RAID de gérer la situation avec élégance. Heureusement, les choses se passent assez souvent ainsi.

5.1 Défaillance d'un disque

Débranchez le disque. Ceci n'est à faire qu'avec le système **hors-tension** . Inutile de jouer les aventuriers de l'ajout/suppression à chaud pour vérifier que les données supportent bien la disparition d'un disque. Arrêtez le système, débranchez le disque et redémarrez le de nouveau.

Syslog et `/proc/mdstat` permettent de vérifier l'état de la matrice.

N'oubliez pas que vous **devez** employer du RAID-{1,4,5} pour que la matrice soit capable de supporter la défaillance d'un disque. Les modes linéaire et RAID-0 échouent complètement dans ce genre de situation.

Une fois le disque rebranché (avec le courant arrêté, merci), on ajoutera le “nouveau” disque au système RAID avec la commande `raidhotadd`.

5.2 Corruption de données

Le RAID, qu'il soit matériel ou logiciel, suppose que si une écriture ne renvoie pas une erreur, alors elle s'est déroulée normalement. Donc, si un diseue corrompt les données sans retourner d'erreur, les données *seront* corrompues. Bien que ce soit fortement improbable, on ne peut l'exclure et cela aura pour conséquence la corruption du système de fichiers.

Le RAID ne peut rien faire face à ce genre de défaillances et il n'a pas été prévu pour de toutes façons. Il est donc inutile de déclencher sciemment des corruptions de données (avec `dd` par exemple) pour vérifier le comportement de la couche RAID. A moins de modifier le superbloc RAID, il est vraisemblable que la couche RAID ne remarque rien mais que le système de fichiers soit détruit.

Il s'agit du fonctionnement normal du système. Le RAID ne garantit pas l'intégrité des données. Il permet juste de les conserver si un disque tombe en panne (pourvu qu'on utilise un niveau de RAID supérieur ou égal à 1).

6 Reconstruction

Si vous avez lu le reste du document, vous devez déjà avoir une bonne idée de la procédure à suivre pour la reconstruction d'une matrice dégradée. Pour résumer :

- Arrêtez le système.
- Remplacez le disque défectueux.
- Redémarrez le système.
- Utilisez `raidhotadd /dev/mdX /dev/sdX` pour réinsérer le disque dans la matrice.
- Allez prendre un café pendant que la reconstruction s'effectue.

C'est tout.

Enfin, c'est généralement tout. Sauf si vous jouez de malchance et que le système RAID est devenu inutilisable à cause de la défaillance de plus de disques qu'il n'y en a de redondant. Ca risque de se produire si plusieurs disques résident sur un même bus qui est bloqué par le disque en panne. Bien qu'en état, les autres disques sur le bus vont être inaccessibles à la couche RAID et marqués comme défectueux. Pour une matrice RAID5 où on peut utiliser un disque en secours, la perte de deux disques ou davantage risque de s'avérer fatale.

La section suivante est tirée d'une explication que m'a donnée Martin Bene et présente une méthode possible de récupération dans le cas d'un scénario catastrophe tel que celui décrit. Elle implique l'emploi de la directive `failed-disk` dans le fichier `/etc/raidtab`. Elle ne pourra donc fonctionner qu'avec un noyau 2.2.10 et au delà.

6.1 Rattrapage d'une défaillance de plusieurs disques

Scénario :

- un contrôleur rend l'âme et bloque simultanément l'accès à deux disques ;
- tous les disques d'un même bus SCSI sont bloqués à cause d'un même disque défectueux ;
- un cable s'envole pour le grand centre de traitement automatisé.

En bref : le plus souvent, une panne *temporaire* se produit sur plusieurs disques. Les superblocs RAID sont désynchronisés et la matrice RAID refuse de s'initialiser.

Une seule chose à faire : reécrire les superblocs RAID via `mkraid -force`.

Pour que ça marche, le fichier `/etc/raidtab` ; doit être à jour. S'il ne correspond pas **exactement** à l'organisation des disques et à leur ordre, ça ne marchera pas.

Examinez la sortie de syslog produite lors de la tentative de démarrage de la matrice, vous y releverez le compteur d'évenements pour chaque superbloc. En général, il vaut mieux laisser de côté le disque avec le compteur le plus faible, c'est à dire le plus ancien.

Si vous exécutez `mkraid` sans la directive `failed-disk`, le thread de récupération va se mettre à fonctionner immédiatement et commencer à reconstruire les blocs de parité - ce qui est surement un peu prématuré.

Avec `failed-disk`, vous préciserez quels disques vous souhaitez voir actifs et essaierez peut-être différentes combinaisons pour obtenir les meilleurs résultats. Pendant la reconstruction, ne montez le système de fichier qu'en lecture seule. J'ai été en contact avec deux personnes qui s'en sont sorties ainsi.

7 Performances

Cette partie offre quelques évaluations de performances issues de tests de systèmes employant le RAID.

Chunk size	Block size	Lecture ko/s	Ecriture ko/s
4k	1k	19712	18035
4k	4k	34048	27061
8k	1k	19301	18091
8k	4k	33920	27118
16k	1k	19330	18179
16k	2k	28161	23682
16k	4k	33990	27229
32k	1k	19251	18194
32k	4k	34071	26976

Les tests ont été conduits avec `bonnie` et à chaque fois sur des fichiers de taille égale à deux fois ou plus celle de la mémoire physique de la machine.

Ces tests ne mesurent *que* la bande passante en lecture/écriture pour un seul fichier de grande taille. On ne sait donc rien des performances qu'on observerait avec un serveur web, un serveur de news, etc... Peu d'applications du monde réel font la même chose que `bonnie` et bien que ce genre de nombres soit agréable à regarder, il ne s'agit pas d'indicateurs de performances pour de véritables applications. On en est loin.

Concernant ma machine :

- Bi-Pentium Pro 150 MHz
- 256 Mo RAM (60 MHz EDO)
- trois IBM UltraStar 9ES 4.5 GB, SCSI U2W
- Adaptec 2940U2W
- un IBM UltraStar 9ES 4.5 GB, SCSI UW
- Adaptec 2940 UW
- un noyau 2.2.7 avec les patches RAID

Les trois disques U2W sont connectés au contrôleur U2W et le disque UW au contrôleur UW.

Avec ou sans RAID, il ne semble pas possible de tirer plus de 30 Mo/s du bus SCSI sur cette machine. Je soupçonne que cela vienne de la vétusté de ce dernier et de la limitation de la bande passante de la mémoire (Nd : pardon ?).

7.1 RAID-0

Lecture correspond à **Sequential block input**, et **Ecriture** à **Sequential block output**. La taille du fichier était de 1 Go pour tous les tests. Les test ont eu lieu en mono-utilisateur. Le gestionnaire SCSI était configuré de façon à ne pas utiliser la queue de commands SCSI.

A la lecture de ce tableau il semble que le paramètre chunk-size du RAID n'ait pas un impact important. Néanmoins, la taille de bloc pour ext2 a intérêt à être aussi élevée que possible, soit 4 ko (i.e. la taille d'une page) avec une architecture IA32.

Chunk size	Block size	Lecture ko/s	Ecriture ko/s
32k	4k	33617	27215

Chunk size	Block size	Lecture ko/s	Ecriture ko/s
8k	1k	11090	6874
8k	4k	13474	12229
32k	1k	11442	8291
32k	2k	16089	10926
32k	4k	18724	12627

7.2 RAID-0 avec queue de commandes SCSI (TCQ)

La queue de commandes est cette fois-ci activée avec une profondeur égale à 8. Le reste est inchangé.

Aucun autre test n'a été mené. L'emploi de la queue de commandes améliore les performances en écriture mais la différence n'est pas énorme.

7.3 RAID-5

On reprend les mêmes tests.

Les deux paramètres semblent jouer.

7.4 RAID-10

On désigne sous ce terme la mise en mirroir de disques concaténés ou un RAID-1 au dessus d'un RAID-0. La taille de bloc est commune à tous les disques RAID. Je n'ai pas procédé à des tests avec des tailles différentes bien qu'il s'agisse là d'une configuration tout à fait licite.

Il n'y a pas eu d'autres tests. La taille des fichiers était de 900 Mo car les partitions n'offraient que 500 Mo chacune, ce qui ne suffit pas pour un fichier de 1 Go dans cette configuration (RAID-1 de deux matrices de 1000 Mo).

8 Remerciements

Les personnes suivantes ont contribué à la création de ce document :

- Ingo Molnar
- Jim Warren
- Louis Mandelstam

Chunk size	Block size	Lecture ko/s	Ecriture ko/s
32k	1k	13753	11580
32k	4k	23432	22249

- Allan Noah
- Yasunori Taniike
- Martin Bene
- Bennett Todd
- les abonnés de la liste de diffusion Linux-RAID
- ceux que j'ai oublié, désolé :o)

Envoyez vos remarques et suggestions à l'auteur de ce document. Il n'y a pas d'autre façon d'en améliorer la qualité.