

Optimisation SSD avec Ubuntu

Optimiser les réglages d'un SSD avec Ubuntu
14.04

Wulfk
24/06/2015



GNU/LINUX SUR UN SSD

Je ne vais pas revenir sur les avantages et inconvénients des SSD, j'ai déjà traité le sujet en introduction dans l'article [Optimisation Windows 7 sur SSD](#)

par contre l'utilisation d'un SSD avec une distribution GNU/Linux est un peu particulière, et pour certains elle pourra paraître plus délicate à maîtriser.

D'abord il faut quand même prendre en compte, ou tout du moins connaître **trois points techniques** liés aux SSD.

TERMES TECHNIQUES LIÉS AUX SSD

- **Alignement des partitions**

Même si l'alignement des partitions n'est pas spécifique aux SSD, il n'a que très peu d'importance sur un *spinoff*, sauf dans le cas des volumes RAID5. L'alignement consiste à faire correspondre les blocs logiques des partitions avec les blocs physiques du SSD pour améliorer les performances de celui-ci afin de limiter les opérations de lecture et d'écriture.

- **Wear levelling**

C'est une Technique utilisée par les contrôleurs des SSD. Elle consiste à répartir l'usure des puces mémoires en écrivant le moins souvent possible dans les mêmes cellules, et en profitant ainsi au maximum du nombre de cycles de lecture-écriture de chacune des cellules. De ce fait, avec un bon algorithme de *wear levelling*, on arrive à faire en sorte qu'un SSD ait une durée de vie de l'ordre de plusieurs années.

- **Garbage collector**

C'est un Mécanisme visant à réorganiser la table d'allocation à la volée, permettant de conserver un bon niveau lors d'écritures séquentielles sur une zone précédemment écrite de manière aléatoire.

Avantage : on récupère les performances d'origine du SSD en écriture séquentielle ou presque.

Inconvénient : on génère de nombreuses écritures dans les puces mémoire, ce qui a tendance à amoindrir la durée de vie du SSD, et le disque travaillant en interne, il ne libère pas les pages de Flash comme le fait le TRIM, ce qui fait que cela n'est efficace que pour les écritures séquentielles, sans améliorer les écritures aléatoires.

Tous les SSD semblent utiliser ce procédé, de manière plus ou moins agressive, et ce à la volée ou en tâche de fond (on parle alors de *background garbage collection*) en fonction de l'objectif du fabricant.

ALIGNEMENT DES PARTITIONS

- **C'est quoi l'alignement ?**

Le contrôleur d'un SSD gère la mémoire par "blocs", généralement de 1 Mio. Cela sert à plusieurs choses : d'une part, les accès mémoire se font généralement par blocs pour améliorer les performances, et d'autre part, ces blocs sont régulièrement permutés pour prolonger la durée de vie du disque. Pour de bonnes performances, il est préférable que le début de partitions coïncide avec le début des blocs. On appelle cela l'alignement.

- **Vérification de l'alignement du SSD?**

Optimisation SSD avec Ubuntu

Par défaut, lors du formatage d'un disque, Ubuntu détecte qu'il s'agit d'un SSD et **aligne les partitions automatiquement** (y compris en partitionnement manuel).

Si vous voulez en être sûr, tapez dans un terminal :

```
sudo fdisk -lu /dev/sda
```

Vous devriez obtenir quelque chose comme ceci :

```
Disque /dev/sda: ---- Go, ----- octets
--- têtes, -- secteurs/piste, ----- cylindres, total ----- secteurs
Unités = secteurs de 1 * 512 = 512 octets
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Identifiant de disque : -----
```

Périphérique	Amorce	Début	Fin	Blocs	Id	Système
/dev/sda1		XXXX	YYYY	ZZZZ	83	Linux

Vous pouvez alors vérifier que le début de chaque partition ("XXXX") est un multiple de 2048 (secteurs). Comme un secteur fait 512 octets, et que $2048 \times 512 = 1$ Mio, votre SSD est aligné !

- **Formatage manuel du SSD (usage avancé)**

Je ne m'étendrai pas sur ce point car cette méthode provient du forum <http://forum.hardware.fr> et elle a été initialement conçue pour **Arch**. La méthode de partitionnement est susceptible de varier sous **Ubuntu**. **Ne suivez cette méthode que si vous êtes sûr de vous et que si vous savez résoudre des problèmes pouvant arriver au cours d'un formatage !**

- **Formatage automatique du SSD (de base)**

Ne pas confondre formatage et partitionnement, même si les deux sont liés, on peut parfaitement faire un formatage automatique, et un partitionnement manuel.

C'est ce que j'ai choisi pour mon cas, j'ai partitionné mon **SSD CORSAIR P64** (64Go) comme ceci :

```
sdd1 : /
```

```
sdd2 : /home
```

il n'y a pas de **partition swap**

Note: sur ma config principale Ubuntu 14.04 64Bits avec un **SSD SAMSUNG Série 850 EVO 250Go SATA III** j'ai créé une partition Swap)

[Pour plus d'info sur le Swap, lire le paragraphe 3](#)

- **Optimiser l'usage du SSD**

Il n'existe pas d'étude suffisamment claire quant à la durée de vie des mémoires [SSD](#) par rapport aux disques durs. Les sections qui suivent vous expliquent comment minimiser les accès en écriture du SSD.

Les points 2 à 4 sont réservés aux PC ayant plus de 512 Mo de mémoire vive (RAM).

Optimisation SSD avec Ubuntu

1 - RÉDUIRE LA FRÉQUENCE D'ÉCRITURE DES PARTITIONS

Utilisez l'option **noatime** pour éviter d'écrire sur le disque la date du dernier accès en lecture lorsqu'il n'y a pas d'écriture. De même avec **nodiratime** pour les dossiers.

Note : *il existe le même type de réglage sur Windows pour les partitions NTFS*

Vérifier que vos partitions sont montées avec cette option en ouvrant le fichier **/etc/fstab** avec les droits d'administration: [!] Préférable d'utiliser [nano](#) plutôt que gedit.

```
sudo gedit /etc/fstab
```

Dans lequel vous trouvez des lignes telles que :

```
UUID=57480a3f-e7db-4a5e-9fca-7df45f5a7d9d / ext4
defaults,noatime,errors=remount-ro 0 1
```

Si **noatime** n'est pas indiqué après **defaults**, vous pouvez le rajouter (séparé par une virgule)
nodiratime est superflu, car noatime est un sur-ensemble de nodiratime (qui est alors sous-entendu)

2 - METTRE LES FICHIERS TEMPORAIRES EN MÉMOIRE VIVE

Le système utilise un certain nombre de fichiers temporaires, qu'il n'est pas nécessaire de conserver d'un démarrage à l'autre. Il est possible de les placer dans la mémoire vive (qui est vidée à l'arrêt de l'ordinateur) au lieu de les avoir dans le SSD.

Cependant, certains logiciels (tels que l'environnement de bureau KDE) utilisent un grand nombre de fichiers temporaires, et devront alors les recréer, ce qui peut ralentir le démarrage si vous utilisez ces logiciels.

Pour mettre les fichiers temporaires en mémoire vive (par exemple, pour une taille maximum de **1 Go**), ouvrez dans un terminal le fichier **/etc/fstab** avec les droits d'administration: [!] Préférable d'utiliser [nano](#) plutôt que gedit.

```
sudo gedit /etc/fstab
```

Et ajoutez-y la ligne suivante :

```
tmpfs /tmp tmpfs defaults,size=1g 0 0
```

On peut vérifier dans un terminal que le "**tmpfs**" est bien présent avec la commande : **df**

Pour ceux qui laissent leur machine allumée 24h/24, une astuce consiste à placer une règle **cron** qui vide les fichiers temporaires à intervalle régulier, afin de ne pas encombrer la mémoire. La commande à mettre dans la tâche **cron** est la suivante :

```
find /tmp -type f -mmin +1440 -delete > /dev/null
```

3 - LE SWAP

Les SSD n'aiment pas le SWAP (*partition d'échange qui permet de délester l'utilisation de la mémoire, mais oblige à de multiple lecture/écriture sur le périphérique*)

Optimisation SSD avec Ubuntu

Il est possible de ne pas créer de partition SWAP durant l'installation d'Ubuntu en définissant les partitions manuellement (mode avancé). Cela va forcer l'utilisation de la mémoire vive et économiser l'espace qu'aurait pris cette partition sur le disque dur !

- **Supprimer le swap**

Cette astuce ne s'adresse qu'à ceux dont la machine dispose d'au moins **1 Go de RAM**. Il est possible dans ce cas, et suivant l'utilisation que l'on fait de la machine de ne pas se servir de swap. Pour cela, soit on ne crée aucune partition de swap lors de l'installation de la distribution, soit, dans le fichier `/etc/fstab`, on commente la ligne montant le fichier swap avec le signe `#` :

```
UUID=bd746caf-bd0c-4649-baa7-d680bb91a6d0 swap swap defaults 0 0
```

Devient alors :

```
#UUID=bd746caf-bd0c-4649-baa7-d680bb91a6d0 swap swap defaults 0 0
```

Il peut être préférable d'utiliser la méthode consistant à ne pas monter une partition de swap existante, de manière à pouvoir la réactiver facilement en cas de besoin, l'utilisation du swap pouvant varier en fonction de l'utilisation de la machine.

Si malgré tout vous avez créé une partition SWAP (notamment pour bénéficier de l'hibernation sur les portables), mais que vous souhaitez en minimiser l'usage, rien n'est perdu pour autant, dans un terminal ouvrir le fichier `/etc/sysctl.conf` avec les droits d'administration [!] Préférable d'utiliser [nano](#) plutôt que gedit.

```
sudo gedit /etc/sysctl.conf
```

Et ajoutez à la fin :

```
vm.swappiness=0
```

Cette ligne indique à Ubuntu de n'utiliser le swap qu'en dernier recours quand votre RAM est "full" ! (" 1 " indique de l'utiliser seulement lorsqu'il ne reste que 1 % de disponible en RAM, " 2 " quand il reste 2 %, etc...)

Une autre méthode consiste à activer [zRAM](#), qui est un module du noyau qui compresse la mémoire vive au lieu de la déplacer dans la partition d'échange en cas de manque de mémoire vive. La partition d'échange ne servant que si la mémoire vive est entièrement compressée. Pour ce faire il suffit d'installer le paquet **zram-config** et le tour est joué.

```
sudo apt-get install zram-config
```

4 - MODIFIER LE CACHE DE FIREFOX

Pour déplacer le cache de Firefox dans le répertoire `/tmp/firefox` (RAM). Taper `about:config` dans Firefox et créer une nouvelle chaîne de caractères que vous nommerez `browser.cache.disk.parent_directory` et saisissez `/tmp/firefox`.

Si vous avez une connexion suffisamment rapide, vous pouvez désactiver complètement le cache persistant en modifiant l'option `browser.cache.disk.enable` avec la valeur `false`.

Optimisation SSD avec Ubuntu

5 - LE TRIMMING.

Comme il a déjà été dit, les disques SSD voient leurs performances diminuer à mesure que les cellules (bloc mémoire) se remplissent (même partiellement) et cela induit des performances en écriture de plus en plus médiocres à mesure que le SSD vieillit et qu'il y a de moins en moins de cellules vierges disponibles. Ce phénomène est dû à la manière dont le SSD fonctionne en écriture au niveau des cellules de mémoire flash : si une cellule est vierge, le contrôleur peut directement écrire dedans - par contre si elle contient déjà des données (même 1 seul bloc d'allocation de 4k alors que la cellule contient 1024k au total), le contrôleur doit lire la cellule + la remettre à zéro + écrire les données.

Une opération beaucoup plus lente. Le **Trimming** a pour objet d'indiquer au contrôleur du SSD lorsque des blocs d'allocation se libèrent suite à l'effacement de fichiers par le *filesystem* (ce qu'il ne peut pas savoir directement a priori); et donc que ces plages libérées sont à nouveau disponibles pour écrire dessus - charge étant au contrôleur de remettre à zéro les cellules qui peuvent l'être et ainsi augmenter les cellules vierges disponibles. Bien entendu, il faut que le SSD supporte la commande **TRIM** afin de pouvoir réaliser cette opération (ce qui est le cas pour la plupart des SSD récents). Vous pouvez vérifier cela dans un terminal :

```
sudo hdparm -I /dev/sda
```

Cette commande permet de lister tous les paramètres et fonctionnalités du SSD. Dans le paragraphe "*Commands/features*" une ligne doit clairement indiquer le support TRIM. Il existe plusieurs mécanismes de Trimming des SSD :

- **Trimming à la volée**

Conditions requises :

- Disposer d'un SSD supportant le **TRIM**
- Utiliser une distribution proposant au moins le **kernel 2.6.33**
- Utiliser **EXT4** comme système de fichier

C'est de loin la solution la plus facile et la plus souple. Il suffit alors d'éditer le fichier **/etc/fstab** en administrateur: [!] Préférable d'utiliser [nano](#) plutôt que gedit.

```
sudo gedit /etc/fstab
```

Et de rajouter l'option **discard** dans les lignes correspondant au montage des volumes en ext4.

Note: Avec Ubuntu 14.04 le montage des fichiers systèmes avec l'option "discard" n'est pas recommandé

```
# /etc/fstab: static file system information.
#
# <file system> <mount point> <type> <options> <dump> <pass>
proc /proc proc defaults 0 0
# /dev/sda1
UUID=f0d9c48e-00c4-4225-ab21-1c5a42194bc8 /
async,noatime,discard,errors=remount-ro 0 1
# /dev/sda2
```

Optimisation SSD avec Ubuntu

```
UUID=43e974d7-82d9-43b1-b67b-5233b18f056e none swap sw
0 0
/dev/scd0 /media/cdrom0 udf,iso9660 user,noauto,exec,utf8 0 0
```

Ensuite les commandes de **TRIM** sont directement passées au contrôleur du SSD par le kernel, de façon entièrement transparente.

- **Activer le TRIM avec fstrim**

Le montage des fichiers systèmes avec l'option "**discard**" n'est pas recommandé, car cela se traduira probablement par une baisse des performances en utilisation normale. Cependant, vous pouvez utiliser TRIM en exécutant la commande **fstrim** occasionnellement ou de créer votre propre **tâche cron** qui exécute **fstrim** via un calendrier.

Pour activer le TRIM de votre SSD sur Ubuntu, il suffit d'ouvrir un terminal et d'exécutez la commande suivante:

```
sudo fstrim -v /
```

Si vous souhaitez exécuter TRIM régulièrement, vous pouvez simplement créer une **tâche cron** qui exécute la commande **fstrim** pour vous.

Voilà comment faire une tâche cron qui le fera automatiquement

d'abord, exécutez la commande suivante pour ouvrir l'[éditeur de texte nano](#) avec les permissions root:

```
sudo nano /etc/cron.daily/fstrim
```

Saisir ou copier/coller le code suivant dans le fichier:

```
#!/bin/sh
LOG=/var/log/trim.log
echo "*** $(date -R) ***" >> $LOG
fstrim -v / >> $LOG
fstrim -v /home >> $LOG
```

Les deux dernières commandes dans le code ci-dessus effectuent le Trimming réel pour la racine (/) et la partition (/ home) et consigne les résultats dans **/var/log/trim.log**. Vous devrez peut-être modifier ces commandes pour adapter à votre cas particulier. Si vous ne disposez pas d'une partition / home séparée vous devez supprimer la dernière ligne. Ajoutez ici les partitions de SSD pour lequel vous souhaitez activer le travail quotidien du TRIM (normalement, vous devez ajouter "/" si la partition racine est sur le SSD et **"/ home"** si vous avez mis en place une partition home séparée)

Enregistrez le fichier en appuyant sur **Ctrl + O** et appuyez sur **Entrée** pour confirmer. Appuyez sur **Ctrl + X** pour fermer nano après avoir sauvegardé le fichier

dernièrement, exécutez la commande suivante pour rendre le script exécutable:

```
sudo chmod +x /etc/cron.daily/fstrim
```

OPTIMISER LA VITESSE DU SSD

Optimisation SSD avec Ubuntu

- **Input/Output Scheduling**

Ce mécanisme de réarrangement des IOCTL a pour objet d'optimiser les commandes I/O vers le disque dur ATA/SATA en prenant en compte la nature du disque dur en question et certaines contraintes en découlant. Il y a trois différentes options : **cfq**, **noop** et **deadline**. Par défaut dans Ubuntu, l'option "*cfq*" est utilisée car elle convient bien aux disques durs mécaniques, en réorganisant la queue des commandes I/O en fonction des temps de rotation des plateaux et des délais de "seek" des têtes. Vous pouvez vérifier quel I/O Scheduler est utilisé par votre système dans un terminal comme ceci :

```
cat /sys/block/sda/queue/scheduler
```

L'option "**deadline**" est probablement la meilleure car elle optimise les I/O pour un gain en rapidité de temps d'accès aux SSD. Heureusement, il existe une manière simple de déclarer cette option "*deadline*" de façon permanente en la passant directement dans les paramètres donnés au kernel lors du démarrage.

6 - POUR GRUB2 (DEPUIS UBUNTU 9.10)

Il faut modifier le fichier en éditant **/etc/default/grub** avec les droits d'administrateur : [!] Préférable d'utiliser [nano](#) plutôt que gedit.

```
sudo gedit /etc/default/grub
```

Et rajouter "**elevator=deadline**" à la ligne d'options :

```
GRUB_CMDLINE_LINUX_DEFAULT="elevator=deadline quiet splash"
```

Puis **mettre à jour grub** pour que les modifications soient passées à **grub.cfg**

```
sudo update-grub
```

7 - SERVICES FSTRIM ET FSTRIM.TIMER POUR UBUNTU 15.04 ET SUPÉRIEUR

Depuis Ubuntu 15.04, Ubuntu utilise pour le démarrage et l'arrêt, **Systemd** et la commande **systemctl**.

Il n'est donc plus nécessaire de passer par une tâche **cron.daily** pour utiliser **fstrim**

```
systemctl enable fstrim.timer
```

Suffit à activer **fstrim**, il sera lancé selon la périodicité par défaut de **fstrim.timer**

taper ou copier/coller la commande :

```
systemctl status fstrim.timer
```

Il va retourner l'état de **fstrim.timer** : **active (waiting)**

```
systemctl list-timers
```

Doit renvoyer les timers actifs et en particulier il doit y avoir une ligne pour **fstriml.timer** qui indiquera dans combien de jours le service lancera **fstrim** dans la colonne **left** (une fois par semaine, c'est le dimanche à minuit)

Optimisation SSD avec Ubuntu

Pour modifier la périodicité de **fstrim.timer**, passe par **systemctl edit**, ça évite de toucher au fichier de configuration original **/usr/lib/systemd/system/fstrim.timer**
taper ou copier/coller la commande :

```
sudo systemctl edit fstrim.timer
```

Ça va lancer votre éditeur et créer le fichier **/etc/systemd/system/fstrim.timer.d/override.conf**
écrire juste dans **override.conf**

[Timer]

OnCalendar=daily

Quand vous aurez fini l'édition, **systemctl edit** va relancer automatiquement le service
vous pourrez de nouveau vérifier la périodicité avec **systemctl list-timers**
et l'agrégation des fichiers par **systemctl cat "fstrim.timer"**

Si vous voulez retrouver la périodicité par défaut (une fois par semaine), il suffira de supprimer le répertoire
/etc/systemd/system/fstrim.timer.d/ contenant **override.conf** (il faut supprimer le répertoire créé par
systemctl edit, pas seulement **override.conf**)

Merci à Taras_Boulba du forum PC Astuces, pour avoir fournis la configuration de fstrim.timer

Sources originales :

- http://doc.ubuntu-fr...lid_state_drive
- [Les SSD sous Linux : recensement, optimisation, conseils](#)
- [Optimiser son SSD sous Linux](#)
- [le sujet de base que j'ai ouvert sur forum Malekal](#)

Autre sources :

- <http://www.howtogeek...le-it-yourself/>
- <http://www.webupd8.o...ate-drives.html>