


Fragmentation

- Objet : Fragmentation des données sur un disque
- Suivi :
 - à-placer
 - Création par  [smolski](#) le 22/09/2012
- Commentaires sur le forum : [C'est ici^{1\)}](#)

Préambule

La taille d'un fichier se mesure en octets.

Un disque dur est compartimenté en blocs de quelques kilo-octets (4 ko pour le système de fichiers ntfs sur les volumes de 2 Go à 2 To [source](#)), ce bloc étant appelé un cluster. Il s'agit en fait de la plus petite taille que peut occuper un fichier donné sur votre disque dur.

Par exemple, si vous avez un fichier de 17 ko, il va occuper 5 clusters (pour un total de 20 ko ; en effet, 4 clusters seraient insuffisants, car avec 16 ko au total, ils ne permettraient pas l'écriture intégrale du fichier), et il restera 3 ko réservés (20 ko disponibles dans les 5 clusters, mais seulement 17 ko occupés par le fichier), mais cependant non utilisés.

Lorsque vous utilisez Windows et que vous avez besoin d'écrire un fichier sur le disque dur, le logiciel se positionnera sur le premier cluster libre qu'il rencontrera, et commencera à le remplir. S'il a besoin de plus d'un cluster, il écrira également dans le cluster voisin, puis dans le suivant, et ainsi de suite, jusqu'à ce que le fichier entier soit écrit, ou que le cluster qui suit soit déjà occupé par un autre fichier.

En ce cas, il continuera à parcourir le disque dur jusqu'à trouver un nouveau cluster libre et écrira dedans et ses voisins, ainsi de suite, jusqu'à ce que le fichier entier soit écrit, en autant de blocs séquentiels logiques possibles.

Bien sûr, il en résulte qu'un même fichier peut être écrit, petit bout par petit bout, en plusieurs endroits du disque.

Si les données sont trop dispersées, la lecture d'un fichier peut en être plus ou moins ralentie, et le phénomène empire à mesure que le disque se remplit.

On dit alors que le disque est *fragmenté*, c'est-à-dire que les données ne sont pas rangées à la suite les unes des autres.

Défragmentation

Préambule

Sous Windows, il existe des utilitaires qui permettent d'analyser le contenu du disque, et de réorganiser son contenu de façon à ce que tous les clusters liés au même fichier soient contigus, et dans le bon ordre.

Cette opération d'analyse et de réorganisation s'appelle la **défragmentation**. Elle permet, dans certains cas, de nettes accélérations du système.

Sous GNU/Linux, cette opération n'est pas nécessaire, car de par leur conception, les systèmes de fichiers²⁾ **ext4** ou **reiserfs** (*par exemple*) vont avoir une tendance naturelle à n'écrire que là où il y aura la place de le faire, et d'optimiser l'espace disque consommé.

Bien sûr, lorsque le disque commence à être plein, et que de gros volumes de données sont manipulés, il n'y aura pas de miracle : le système écrira ce qu'il pourra où il pourra, même s'il en résulte une fragmentation des fichiers.

GNU/Linux

Pour constater l'état de fragmentation de ses disques, voir :

```
man e2freefrag
```

En attendant un tuto df sur cet outil bien sûr ! 😊

Il est incorrect de dire qu'il n'existe pas d'outils de défragmentation sous GNU/Linux, mais il conviendra d'insister sur le fait qu'ils sont le plus souvent totalement inutiles.

Alors doit-on défragmenter sous Linux ?

Si vous écoutez les prêcheurs anti-Windows, jamais de la vie : Linux est tellement plus intelligent que son système de fichier ne fragmente (*presque*, mais on l'oublie souvent) pas.

Il est vrai que les systèmes de fichiers Linux font un usage optimisé de l'espace, mais il n'est malheureusement pas en mesure de savoir aujourd'hui quelle sera l'organisation optimale de votre disque après une ou deux années d'usage intensif.

Pour résumer, si vos disques durs ont une durée de vie supérieure à deux ans, que vous travaillez régulièrement dessus, que vous manipulez quotidiennement de gros fichiers (vidéos, morceaux de musique...), que vous vous livrez au P2P ou encore qu'il vous arrive d'exhumer de vieux (et gros !) fichiers pour retravailler dessus, vous ressentirez vite, comme tout le monde, le besoin de défragmenter votre beau disque Linux !

Donc sous GNU/Linux, si vous avez besoin de réaliser cette opération, il ne vous reste plus qu'à vous intéresser à l'un des projets suivants :

- Shake (<http://vleu.net/shake/>)

ou encore :

- Defrag (<http://ck.kolivas.org/apps/defrag/>)



Mais ne sombrez pas dans la paranoïa et assurez-vous bien d'en avoir réellement besoin avant de vous lancer dans l'aventure : pour un utilisateur lambda, cela n'en vaudra certainement pas la peine...

Source

<http://www.unixgarden.com/index.php/linux-pratique-hs/pourquoi-ny-a-t-il-pas-doutils-de-defragmenta>

tion-sous-linux

màj : Lien mort...

SSD

La fragmentation a peu d'impact sur les performances **en lecture** d'un SSD car le temps d'accès est négligeable par rapport à celui d'un disque dur.

Comme je ne suis pas sous windows mais sous linux, il ne se fragmente pas

L'absence de fragmentation sous Linux est une légende urbaine.

Tous les [systèmes de fichiers](#) qui *vivent* fragmentent, et ceux de Linux n'y font pas exception.

Certains systèmes de fichiers comme ext4 avec la notion d'*extent* permettent des stratégies d'allocation ayant pour but de limiter la fragmentation, mais il ne faut pas oublier que :



plus l'espace libre est faible, plus la probabilité de fragmentation augmente.

D'autre part il ne s'agit que de la fragmentation externe, due au système de fichiers.

Les SSD ont en plus une fragmentation interne qui résulte des contraintes d'écriture de la mémoire flash.

Comme l'a écrit **robert2a**, contrairement à un disque dur on ne peut pas juste réécrire dans un bloc de mémoire flash qui contient des données, on doit l'effacer avant.

Or l'effacement est une opération longue et qui se fait sur des blocs de taille largement supérieure à la taille d'un bloc d'écriture (et qui use la mémoire flash).

Donc, pour augmenter la vitesse d'écriture, lorsqu'on modifie le contenu d'un bloc logique au lieu de :

1. lire les données du bloc d'effacement,
2. effacer le bloc d'effacement et
3. réécrire les données modifiées,

les nouvelles données sont écrites dans un *nouveau bloc physique* déjà effacé. ⚠



Cela implique que tout fichier modifié depuis sa création a des chances d'être physiquement fragmenté sur un SSD.

Encore une fois, ce n'est pas gênant **en lecture** puisque le temps d'accès est négligeable.

En écriture par contre, une fragmentation excessive de l'espace libre peut affecter les performances puisqu'il va être de plus en plus difficile de trouver des blocs effacés prêts à l'écriture.

C'est pourquoi les SSD ont un mécanisme interne de *ramasse-miettes*³⁾ qui a pour but de : rassembler les blocs contenant des données et de libérer d'autres blocs partiellement remplis afin de les effacer pour qu'ils soient prêts à être écrits.



À noter que le marquage des blocs inutilisés par le système hôte avec **TRIM/discard** facilite ce processus.

Si ça ne ressemble pas à de la défragmentation, je ne sais pas ce que c'est. 🤔

De raleur sur le forum là : <https://debian-facile.org/viewtopic.php?pid=192639#p192639>

Merci à tous les intervenants qui permettent ainsi à l'amélioration des connaissances de tous, depuis çui qui demande jusqu'à tous ceux qui participent.

Ainsi quel que soient les lacunes, sur DF elles sont sources de Savoir pour tous ! 🤖

Trim

naguam a écrit :

mais imaginons que j'ai besoin de le faire car mon SSD serait supposé plein..... Comment devrais-je faire ?

raleur :

Tant pis. La défragmentation occasionne beaucoup d'écritures, à mon avis ce n'est pas une bonne idée de l'utiliser sur un SSD.

Le ramasse-miettes interne est automatique. On peut juste l'aider en utilisant le TRIM (discard).

nlQnutn a écrit :

il me semble que certains systèmes de fichiers sont plus avancés pour une utilisation avec SSD ?

raleur :

Tous les systèmes de fichiers ne supportent pas l'option "discard" qui utilise la fonction TRIM des SSD pour marquer les blocs qui peuvent être effacés. ext4, btrfs, et peut-être xfs ont cette option, mais pas ext2 ni ext3.

naguam a écrit :

j'ai lu de mettre /tmp dans la ram, ce que je ne veux absolument pas.... car je ne pense pas avoir besoin de mettre 8GB de ram dans ma machine mais j'utilise quand même pas mal ma ram

raleur :

Un système de fichiers tmpfs peut utiliser le swap si nécessaire.

naguam a écrit :

pourquoi copier.... et explications du truc ?

raleur :

Copier pour ne pas avoir à retaper à la main.

But : exécuter régulièrement fstrim pour marquer comme effaçables les blocs ne contenant pas de données utiles. Evidemment, ces fichiers ne marchent qu'avec systemd comme système d'init.

naguam a écrit :

Demande explications là aussi

raleur :

Activer l'option discard pour un système de fichiers ne suffit pas : si le système de fichiers n'est pas

directement dans un disque ou une partition mais dans un périphérique "virtuel" comme un volume logique LVM, un ensemble RAID logiciel ou un volume chiffré, il faut aussi que ce périphérique supporte et propage l'option discard jusqu'au SSD. Dans le cas d'un volume chiffré, il faut ajouter l'option discard dans `/etc/crypttab`. Mais s'il s'agit d'un volume chiffré qui est ouvert par l'initramfs (car contenant la racine, `/usr` ou le swap de l'hibernation), alors il faut que la modification de `crypttab` soit intégrée à l'initramfs. C'est l'objet de la commande `update-initramfs -u`. L'option `"-k all"` dit de reconstruire les initramfs de tous les noyaux présents, et pas seulement celui du noyau actif.

Severian a écrit :

Intégrer le trim directement dans le fstab (en ajoutant l'option discard)

il me semble me rappeler que c'était au final déconseillé car ça créait plus d'écriture sur le ssd

raleur : Non, ce n'est pas le problème. L'option discard, la commande `fstrim` et la commande `blkdiscard` utilisent la même commande TRIM du SSD. Cette commande ne produit aucune écriture par elle-même, elle ne fait que marquer les secteurs désignés comme effaçables. Cela ne signifie pas qu'ils vont être effacés immédiatement. Mais parfois c'est le cas, et c'est ce qui peut poser problème avec la première version de la commande TRIM qui interrompt et empêche l'exécution de toute opération de lecture/écriture en parallèle pendant tout le temps de son exécution, peut-être au mauvais moment où le système produit une intense activité disque.

(Une seconde version de la commande TRIM a été créée dans une révision ultérieure du standard ATA, mais l'implémentation de cette commande dans plusieurs modèles de SSD est buggée et peut provoquer des pertes de données. Les modèles identifiés comme tels sont répertoriés au fur et à mesure dans le noyau qui n'utilise que la première version (bloquante) avec ceux-ci.) C'est pourquoi certains lui préfèrent un TRIM périodique avec `fstrim` programmé à un moment de faible activité (la nuit).

Source 2 sur le forum

- <https://debian-facile.org/viewtopic.php?pid=192855#p192855>

1)

N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !

2)

partition

3)

garbage collector

From:

<http://debian-facile.org/> - Documentation - Wiki

Permanent link:

<http://debian-facile.org/doc:systeme:fragmentation>

Last update: **07/06/2023 15:18**

