

Linux: Résolution et supervision du remplissage des inodes

Table of Contents

[Description du problème](#)[Résolution](#)[Supervision](#)[Première méthode](#)[Deuxième méthode](#)

Les inodes sont très importants dans un système de fichiers UNIX/LINUX. Chaque fichier ou dossier (sous Unix, tout est fichier) doit correspondre à un numéro d'inode. Celui-ci contient les métadonnées des fichiers et en particulier les droits d'accès. La quantité d'inodes est déterminée lors du formatage de la partition. Le nombre maximum dépend du type de système de fichiers:

- **ext4** Le nombre maximal d'inodes disponibles dépend de la taille du disque et des options choisies lors de la création (il n'y a pas de défaut en tant que tel);
- **XFS** a une allocation dynamique, mais a une limite (pourcentage du système de fichiers pouvant être utilisé pour les inodes). Ainsi, le remplissage de la réserve d'inodes dépend du pourcentage défini ainsi que du nombre de fichiers. / tailles de fichier;
- **Btrfs** a une allocation dynamique des inodes, il n'y a donc pas de remplissage comme avec les tables d'inodes d'ext4 (dont la taille est définie au moment de la création du système de fichiers ext4);
- **ZFS** n'utilise pas les inodes.

Description du problème

Généralement lorsqu'on a un message d'erreur pour un problème d'espace disque.

```
No space left on device.
```

Le premier réflexe est de vérifier l'espace disque restant sur les partitions avec la commande **df** :

```
df -h
/dev/mapper/vg00-lvroot
          12G  2.6G  8.3G  24% /
tmpfs          939M  108K  939M   1% /dev/shm
/dev/sda1      488M   54M  409M  12% /boot
/dev/mapper/vg01-lvvar
          178G   15G  154G   9% /var
```

Dans cet exemple l'espace disque est suffisant, il se peut donc qu'il n'y ait plus d'inodes disponibles dans le système de fichiers.

En utilisant l'opérateur **-i**, on peut constater que les inodes du répertoire /var sont utilisés à 100%:

```
df -i
Filesystem                Inodes    IUsed    IFree IUse% Mounted on
/dev/mapper/vg00-lvroot
                          768544    77544   691000    11% /
tmpfs                      240254      28   240226     1% /dev/shm
/dev/sda1                  32768      44    32724     1% /boot
/dev/mapper/vg01-lvvar
                          11796480 11796480      0   100% /var
```

Résolution

Maintenant que le problème a été identifié. Il faut identifier qu'elle en est la cause. On commence par rechercher l'endroit d'où provient le trop grand nombre d'utilisation d'inode (fichier ou dossier). Pour cela on utilise la commande ci-dessous en commençant par la racine de /var:

```
find . -xdev -type f | cut -d "/" -f 2 | sort | uniq -c | sort -n
    1 account
    1 glpi.tar.gz
    2 net-snmp
   14 lock
   18 tmp
   19 run
  483 log
 2390 cache
 9532 lib
11044 www
2164918 spool
```

Cette commande indique le nombre d'inodes utilisé par les fichiers et sous-dossiers. On peut voir que le répertoire qui utilise le plus d'inodes est **/spool**.

Ensuite, il faut se déplacer de sous-dossier en sous-dossier pour terminer au dossier /var/spool/snmpdtt.

```
find . -xdev -type f | cut -d "/" -f 2 | sort | uniq -c | sort -n
    1 at
    1 cron
    1 plymouth
    3 anacron
    7 mail
811291 snmpdtt
```

Celui-ci est utilisé par le démon snmpdtt pour stocker les traps snmp. Un nombre très impressionnant de fichiers de traps a été créé par une expérimentation, malheureusement le démon n'avait pas été arrêté et les fichiers ont continué à alimenter ce dossier, mais pas suffisamment pour diminuer l'espace d'occupation du disque.

La cause étant identifiée, il faut arrêter le processus provoquant la création des fichiers. Ensuite, il

faut supprimer les fichiers existant pour libérer les inodes en utilisant donc la commande `rm`.



Lorsque le nombre de fichier est trop important, l'utilisation de la commande avec des caractères joker (ex. `rm /var/spool/snmp/tt/*`) risque de retourner l'erreur suivante:

`bash: /bin/rm: Liste d'arguments trop longue`

Il est donc préférable de détruire tout le sous dossier puis de le recréer

Au bout de quelques minutes, la situation est devenue correcte avec un nombre d'inode libre suffisant.

```
df -i
Filesystem                Inodes IUsed   IFree IUse% Mounted on
/dev/mapper/vg00-lvroot
                          768544 83272   685272  11% /
tmpfs                      240254   28    240226   1% /dev/shm
/dev/sda1                  32768   44     32724   1% /boot
/dev/mapper/vg01-lvvar
                          11796480 25648 11770832   1% /var
```

Supervision

Maintenant, pour éviter ce genre de problème, il reste une chose importante à réaliser : superviser le taux d'utilisation des inodes du serveur avec le mode inodes du module **os::linux::snmp::plugin** de `centreon_plugins.pl`.

Si on rencontre l'erreur **UNKNOWN: SNMP GET Request : Cant get a single value.**, il est probable que les autorisations données à l'agent SNMP soient trop restreintes.



Si cela se produit sur le mode Inodes, il est probable que le service SNMP du serveur Linux ne soit pas correctement configuré, il faut ajouter la directive ci-dessous dans le fichier de configuration SNMP (`/etc/snmp/snmpd.conf`) puis redémarrer le service:

```
includeAllDisks 10%
```

Première méthode

La commande vérifie tous les systèmes de fichier. Pour éviter un trop grand nombre de métriques, l'utilisation de l'option `filter-perfdata` est nécessaire. On utilisera la commande suivante :

```
./centreon_plugins.pl --plugin=os::linux::snmp::plugin --mode=inodes --snmp-community=public --hostname=xx.xx.xxx.xx --snmp-version=2 --filter-perfdata='^used_/|^used_/boot|^used_/home|^used_/tmp|^used_/var$' --warning-usage=70 --critical-usage=90
```

Dans le plugin MONITORING de GLPI la commande sera sous cette forme :



```
$PLUGINS_DIR$/centreon_plugins.pl --plugin=os::linux::snmp::plugin --mode=inodes --warning-usage=$ARG1$ --critical-usage=$ARG2L$ --hostname=$HOSTADDRESS$ --snmp1-version=$_HOSTSNMPVERSION$ --snmp-community=$_HOSTSNMPCOMMUNITY$ --filter-perfdata=$_SERVICEFILTER$
```

Il suffira de créer le modèle de service et de compléter chaque service avec la macro FILTER.

Deuxième méthode

Lorsque certains systèmes de fichiers spécialisés génère des faux positifs comme cet exemple :

```
./centreon_plugins.pl --plugin=os::linux::snmp::plugin --mode=inodes --snmp-community=public --hostname=192.168.0.163 --snmp-version=2 --filter-perfdata='^used_/|^used_/boot|^used_/home|^used_/tmp|^used_/var$' --warning-usage=70 --critical-usage=90
```

```
CRITICAL: Inode partition '/dev/pts' Used: 100 % - Inode partition '/proc' Used: 100 % - Inode partition '/sys' Used: 100 % - Inode partition '/var/lib/nfs/rpc_pipefs' Used: 100 % | 'used_/'=7%;0:70;0:90;0:100 'used_/boot'=0%;0:70;0:90;0:100 'used_/home'=0%;0:70;0:90;0:100 'used_/tmp'=0%;0:70;0:90;0:100 'used_/var'=6%;0:70;0:90;0:100
```

On va utiliser l'option diskpath mais comme on veut superviser plusieurs système de fichier en même temps, il faut utiliser la désignation numérique. Pour connaître la position numérique des Filesystem, lancer une première fois la commande en mode debug :

```
/var/lib/shinken/libexec/centreon_plugins.pl --plugin=os::linux::snmp::plugin --mode=inodes --snmp-community=public --hostname=xx.xx.xxx.xx --snmp-version=2 --debug
OK: All inode partitions are ok | 'used_/'=10%;;;0:100 'used_/boot'=0%;;;0:100 'used_/dev/pts'=100%;;;0:100 'used_/dev/shm'=0%;;;0:100 'used_/proc'=100%;;;0:100 'used_/proc/sys/fs/binfmt_misc'=100%;;;0:100 'used_/sys'=100%;;;0:100 'used_/var'=0%;;;0:100
disk path : '/', device : '/dev/mapper/vg_s013e13psnmp-lv_root'
disk path : '/proc', device : 'proc'
disk path : '/sys', device : 'sysfs'
disk path : '/dev/pts', device : 'devpts'
```

```
disk path : '/dev/shm', device : 'tmpfs'
disk path : '/boot', device : '/dev/sda1'
disk path : '/var', device : '/dev/mapper/vg_s013e13psnmp-lv_var'
disk path : '/proc/sys/fs/binfmt_misc', device : 'none'
Inode partition '/' Used: 10 %
Inode partition '/boot' Used: 0 %
Inode partition '/dev/pts' Used: 100 %
Inode partition '/dev/shm' Used: 0 %
Inode partition '/proc' Used: 100 %
Inode partition '/proc/sys/fs/binfmt_misc' Used: 100 %
Inode partition '/sys' Used: 100 %
Inode partition '/var' Used: 0 %
```

Ainsi on peut déterminer les FileSystem numériquement par ordre d'arrivée : 1 → /, 2 → /proc, 3 → /sys, 4 → /dev/pts, 5 → /dev/shm, 6 → /boot, 7 → /var

On peut alors créer une commande pour les disques 1 (root) et 7 (/var) comme ceci:

```
/var/lib/shinken/libexec/centreon_plugins.pl --
plugin=os::linux::snmp::plugin --mode=inodes --snmp-community=public --
hostname=192.168.0.163 --snmp-version=2 --diskpath=1,7 --warning-usage=70
--critical-usage=90
OK: All inode partitions are ok | 'used_/'=10%;0:70;0:90;0:100
'used_/var'=0%;0:70;0:90;0:100
```

Dans le plugin MONITORING de GLPI on peut compléter la commande avec l'option diskpath.



```
$PLUGINS$DIR/centreon_plugins.pl --plugin=os::linux::snmp::plugin -
-mode=inodes --warning-usage=$ARG1$ --critical-usage=$ARG2$ --
hostname=$HOSTADDRESS$ --snmp-version=$_HOSTSNMPVERSION$ --snmp-
community=$_HOSTSNMPCOMMUNITY$ --diskpath=$_SERVICEDISKPATH$
--filter-perfdata=$_SERVICEFILTER$
```

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=notes:linux-full-inodes>

Last update: **2025/02/19 10:59**

