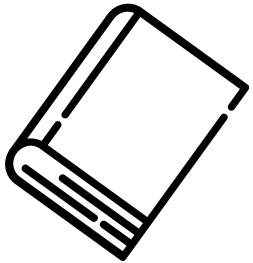
Formation Linux, administration avancée

Objectifs pédagogiques



- Maîtriser les différentes méthodes d'installation et déploiement Linux
- Dépanner des problèmes du système, matériel et du réseau
- Superviser la charge système et l'état du serveur avec Nagios
- Optimiser ses serveurs

UBSECURITY

01
Installation avancée et déploiement

Automatiser l'installation avec kickstart

Vous pouvez permettre à une installation de s'exécuter seule à l'aide de Kickstart.

Le fichier Kickstart spécifie les paramètres pour une installation. Une fois que le système d'installation démarre, il peut lire un fichier Kickstart et commencer le processus d'installation sans participation supplémentaire de la part d'un utilisateur.

Cette méthode est propre à RedHat , chez debian c’est Preseed qui est employé. Un dérivé permettant l’usage de Kickstart avec des options Pressed est disponible chez Ubuntu.

Automatiser l'installation avec kickstart

Il est possible d'effectuer des installations Kickstart à l'aide d'un DVD local, d'un disque dur local ou encore via NFS, FTP, HTTP, ou HTTPS.

Pour utiliser le mode Kickstart, vous devez :

1. Créer un fichier Kickstart.
2. Mettre le fichier Kickstart à disponibilité sur un support amovible, un disque dur ou un emplacement réseau.
3. Créer un support amovible, qui sera utilisé pour commencer l'installation.
4. Rendre la source d'installation disponible.
5. Lancer l'installation Kickstart.

Automatiser l'installation avec kickstart

Création d'un fichier Kickstart

Le fichier Kickstart est un fichier texte contenant des mots-clés qui servent à guider l'installation.

L'ensemble de la syntaxe peut-être consulté ici : Syntaxe Kickstart

Pour créer des fichiers Kickstart, RedHat recommande d'effectuer une installation sur un seul système en premier.
Une fois l'installation terminée, tous les choix effectués dans l'installateur sont enregistrés dans un fichier nommé **anaconda-ks.cfg**, situé dans le répertoire **/root/**

Un petit exemple avec .. LVM on RAID bien entendu : Kickstart LVM on RAID

Automatiser l'installation avec kickstart

Regardons un exemple complet ensemble :

```
# Red Hat | Oracle Solutions Kickstart Script
install
url --url=<place-distro-url-here>
lang en_US.UTF-8
keyboard us
network --onboot yes --device em1 --mtu=1500 --bootproto dhcp
rootpw <password-for-system>
# Reboot after installation
reboot
authconfig --enablemd5 --enableshadow
selinux --enforcing
timezone America/New_York
bootloader --location=mbr --driveorder=sda --append="crashkernel=auto rhgb
quiet"
# The following is the partition information you requested
# Note that any partitions you deleted are not expressed
# here so unless you clear all partitions first, this is
# not guaranteed to work
clearpart --all
volgroup myvg --pesize=32768 pv.008002
logvol /home --fstype=ext4 --name=home --vgname=myvg --size=8192
logvol / --fstype=ext4 --name=root --vgname=myvg --size=15360
logvol swap --name=swap --vgname=myvg --size=16400
logvol /tmp --fstype=ext4 --name=tmp --vgname=myvg --size=4096
logvol /u01 --fstype=ext4 --name=u01 --vgname=myvg --size=51200
logvol /usr --fstype=ext4 --name=usr --vgname=myvg --size=5120
logvol /var --fstype=ext4 --name=var --vgname=myvg --size=8192
part /boot --fstype=ext4 --size=256
part pv.008002 --grow –size=1000
%packages
@Base
@Core
```

Source : Github ou Doc. Redhat (Contenu identique)

Automatiser l'installation avec kickstart

Vérifier le fichier Kickstart

Lors de la création ou personnalisation de votre fichier Kickstart, il est recommandé de vérifier sa validité avant de tenter de l'utiliser dans une installation

Il existe un outil en python pour cela : **ksvalidator**
Cet outil est issue du paquet **: pykickstart**

Installation et usage :

```
yum install pykickstart
ksvalidator /path/to/kickstart.ks
```

Automatiser l'installation avec kickstart

/!\ Note : L'outil de validation possède des limites. Le fichier Kickstart peut être très compliqué.

ksvalidator s'assure que la syntaxe est correcte et que le fichier ne contient pas d'options désapprouvées, mais il ne peut garantir le succès de l'installation.

Il ne valide pas non plus les sections **%pre**, **%post** et **%packages** du fichier Kickstart.

Automatiser l'installation avec kickstart

Les deux étapes suivante vont dépendre de votre mode d’installation.
En effet il nous faut :

- Mettre à disposition le fichier Kickstart
- Mettre à disposition la source d'installation

Nous allons prendre, pour la suite, une installation via PXE.

En général, l'approche la plus couramment utilisée étant que l'administrateur dispose à la fois d'un serveur BOOTP / DHCP et d'un serveur NFS sur le réseau local.

Automatiser l'installation avec kickstart

Nous placerons notre fichier d'installation sur le serveur NFS ou se trouve aussi l'iso de la source d'installation.

Il faut modifier le fichier **pxelinux.cfg/default** pour lui préciser l'emplacement de ce fichier.

Exemple pour les système avec Grub :

label 1
kernel RHEL7/vmlinuz
append initrd=initrd.img
inst.ks=http://**<IP>**/mnt/archive/RHEL-7/7.x/Server/x86_64/kickstarts/ks.cfg

Automatiser l'installation avec kickstart

Exemple pour les système avec Grub2 :
Modifier le fichier grub.cfg et ajouter cette ligne à votre menuentry :

kernel vmlinuz
inst.ks=http://**IP**/mnt/archive/RHEL-7/7.x/Server/x86_64/kickstarts/ks.cfg

Installation ROOT-on LVM on RAID

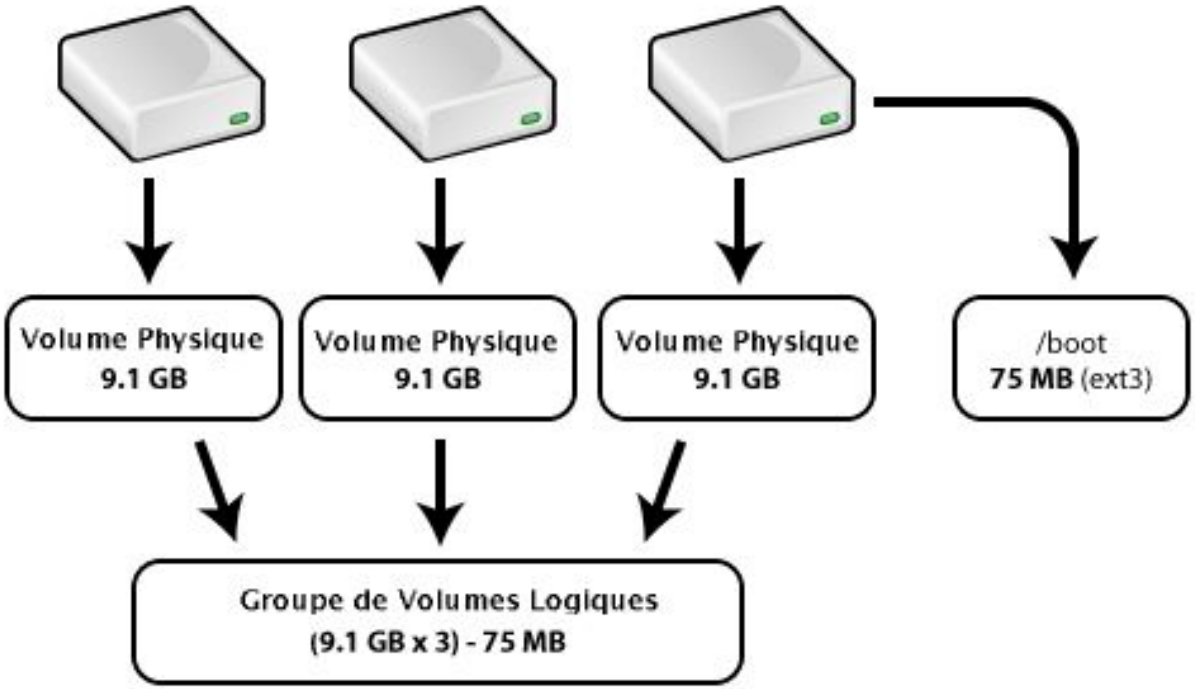
Qu'est-ce LVM ?

LVM est une méthode d'allocation d'espace de disque dur en volumes logiques, dont la taille peut facilement être modifiée, au lieu d'utiliser des partitions.

Avec LVM, le disque dur ou l'ensemble de disques durs est alloué à un ou plusieurs volumes physiques. Un volume physique ne peut pas s'étendre sur plusieurs disques.

Étant donné qu'un volume physique ne peut pas s'étendre sur plusieurs disques, si vous souhaitez tout de même le faire, vous devrez créer un ou plusieurs volumes physiques par disque.

Installation ROOT-on LVM on RAID



/!\ Note : La partition /boot/ ne peut pas se trouver sur un groupe de volumes logiques car le chargeur d'amorçage ne peut pas le lire

Installation ROOT-on LVM on RAID

Pour utiliser LVM avec du RAID, on utilise d'habitude LVM et MD, qui est l'implémentation du RAID logiciel de Linux.

Pour cela, on en agrège ses périphériques de stockage en un volume RAID, qu'on utilise ensuite comme un volume LVM physique

Il s'agit de LVM au-dessus de RAID

Installation ROOT-on LVM on RAID

Pour ce montage nous allons utiliser trois disque de taille identique en RAID5.

Un autre disque sda sera réservé au système racine qui héberge aussi le /boot

Nous avons donc sdb,sdc,sdd pour le montage en RAID 5

Sur les trois disques du raid nous créons à l'aide de fdisk une partition de type "fd"

Installation ROOT-on LVM on RAID

On créer le montage RAID via mdam

mdadm --create /dev/md0 --level=5 --assume-clean --raid-devices=3 /dev/sd[bcd]1

Le troisième disque est utilisé en disque de parité.

Configuration LVM

1. On créer le volume physique : pvcreate /dev/md0
   a. Pour observer le résultat : pvdisplay /dev/md0
2. Création du VG : vgcreate data1 /dev/md0
   a. Pour observer le résultat : vgdisplay data1
3. Création d'une partition de 4G : lvcreate -n Vol1 -L 4g data1
   a. Pour observer le résultat : lvdisplay /dev/data1/Vol1
4. On peut désormais créer un filesystem de son choix sur ce volume.

Sécuriser le système de démarrage

Sécurité du BIOS et du chargeur de démarrage

La protection par mots de passe associés au BIOS (ou un équivalent du BIOS) et au chargeur de démarrage peut empêcher que des utilisateurs non-autorisés ayant un accès physique à vos systèmes ne les démarrent à partir d'un média amovible ou ne s'octroient un accès super-utilisateur par le biais du mode mono-utilisateur

(Méthode que nous explorerons ensemble plus tard).

Sécuriser le système de démarrage

Mots de passe du BIOS

Ci-dessous figurent les deux raisons essentielles pour lesquelles des mots de passe sont utiles pour protéger le BIOS d'un ordinateur :

Empêcher toute modification des paramètres du BIOS — Si un intrus a accès au BIOS, il peut le configurer de manière à démarrer à partir d'une disquette ou d'un CD-ROM.

Il peut entrer dans un mode de secours ou un mode mono-utilisateur, lui permettant alors de lancer des processus arbitraires sur le système ou de copier des données confidentielles.

Sécuriser le système de démarrage

Mots de passe du BIOS

Empêcher le démarrage du système — Certains BIOS permettent de protéger le processus de démarrage à l'aide d'un mot de passe.

Lorsque cette fonctionnalité est activée, tout attaquant est obligé de saisir un mot de passe avant que le BIOS ne puisse lancer le chargeur de démarrage.

Les méthodes utilisées pour établir un mot de passe pour le BIOS varient selon les fabricants d'ordinateurs. Il est conseillé de consulter le manuel de votre ordinateur pour obtenir les instructions appropriées.

Sécuriser le système de démarrage

Mots de passe du BIOS

/!\ Note : il est possible de le réinitialiser soit en utilisant des cavaliers sur la carte mère, soit en débranchant la pile CMOS

L'ajout du chiffrement des disques avec stockage d'une partition boot sur une clé usb permet de se protéger de la récupération physique.

Sécuriser le système de démarrage

Mots de passe du chargeur de démarrage

Cela permet d’empêcher l'accès au mode mono-utilisateur — Si un attaquant peut démarrer en mode mono-utilisateur, il sera automatiquement connecté en tant que super-utilisateur sans avoir à saisir de mot de passe root.

Cela permet aussi d’empêcher l'accès à la console GRUB — Si l'ordinateur utilise GRUB comme chargeur de démarrage, un agresseur peut utiliser l'interface de l'éditeur de GRUB afin de changer sa configuration et de recueillir des informations à l'aide de la commande cat.

Sécuriser le système de démarrage

Mots de passe du chargeur de démarrage

Enfin cela permet d’empêcher l'accès à des systèmes d'exploitation non-sécurisés — Dans le cas d'un système à double démarrage, un attaquant peut, au moment du démarrage, choisir un système d'exploitation tel que DOS qui ne prend en compte ni les contrôles d'accès, ni les permissions de fichiers.

Sécuriser le système de démarrage

GRUB 2 donne deux types de protection de mot de passe :

- Un mot de passe est exigé pour pouvoir modifier les entrées de menu mais pas pour les entrées de menu boot existantes.
- Un mot de passe est requis pour pouvoir modifier les entrées de menu et pour pouvoir démarrer un, plusieurs ou toutes les entrées de menu.

Sécuriser le système de démarrage

Pour demander une authentification de mot de passe afin de modifier les entrées GRUB2, il faut exécutez la commande **grub2-setpassword** en tant qu'utilisateur root

En suivant cette procédure, vous créerez un fichier **/boot/grub2/user.cfg** qui contient le hash du mot de passe. L'utilisateur de ce mot de passe, root, est défini dans le fichier **/boot/grub2/grub.cfg**.

Avec ce changement, pour modifier une entrée boot au moment de l'amorçage, il faut maintenant que vous spécifiez le nom d'utilisateur root et votre mot de passe.

Sécuriser le système de démarrage

Configurer GRUB 2 pour qu'un mot de passe soit exigé pour les modification et le boot.

Pour ce faire, après avoir définit un mot de passe via la commande **grub2-setpassword**

1. Ouvrir un fichier /boot/grub2/grub.cfg .
2. Cherchez l'entrée boot que vous souhaitez protéger avec le mot de passe en cherchant des lignes commençant par menuentry.
3. Supprimer le paramètre --unrestricted du bloc d'entrée de menu
4. Enregistrer et fermer le fichier.

Sécuriser le système de démarrage

/!\ Note : Les changements au **/boot/grub2/grub.cfg** persistent quand de nouvelles versions de noyau sont installées, mais sont perdues quand on génère à nouveau **grub.cfg** par la commande **grub2-mkconfig**.

Ainsi, pour conserver la protection du mot de passe, il vous faudra utiliser la procédure ci-dessus à chaque fois que vous utiliserez **grub2-mkconfig**.

Sécuriser le système de démarrage

Et si je n’ai pas grub2-setpassword ?

Dans le cas ou cette utilitaire n’est pas présent voici la procédure :

Exécutez la commande **grub-mkpasswd-pbkdf2**.
Ensuite, ajustez les autorisations pour /etc/grub.d/40_custom et ajoutez ce qui suit à ce fichier:

/etc/grub.d/40_custom

set superusers = "nom d'utilisateur"
password_pbkdf2 “nom d'utilisateur” “mot de passe”

Régénérez votre fichier de configuration.

Sécuriser le système de démarrage

Pour les curieux voulant plus de détail et ceux voulant personnaliser plus finement , il est conseillé de lire :

https://www.gnu.org/software/grub/manual/grub/grub.html#Security

Créer un CD/DVD/Clé usb de recovery

Pour créer une image Live plusieurs méthodes sont possibles.
Nous verrons ici la méthode Fedora pour rester dans l'univers RedHat.

Pour créer une image en direct, l'outil **livecd-creator** est utilisé.
Pour l'installer nous avons d'abord besoin d'installer les repository EPEL

```
yum install epel-release
yum install livecd-tools
```

L'intérêt de l'outil est de prendre en charge des fichiers kickstart pour la construction de l'image.

L'outil vient d'ailleurs avec un fichier de base : **fedora-live-base.ks**

Créer un CD/DVD/Clé usb de recovery

Créer l'image

Pour créer l'image, exécutez simplement la commande suivante:

```
livecd-creator --verbose \
--config=/path/to/kickstart/file.ks \
--fslabel=Image-Label \
--cache =/var/cache/live
```

Pour tester l’iso le plus simple est de la lancé via qemu/kvm :

```
qemu-kvm -m 2048 -vga qxl -cdrom filename.iso
qemu-system-x86_64 -m 2048 -vga qxl -cdrom filename.iso
```

Créer un CD/DVD/Clé usb de recovery

Placer l'image sur une clé USB

La méthode la plus simple et rapide consiste à utiliser **dd** :

```
dd if=/path/to/image.iso of=/dev/sdX bs=8M status=progress oflag=direct
```

/dev/sdX étant votre périphérique USB.

/!\ Note : Cette méthode détruira toutes les données de la clé USB. Si vous avez besoin d'une méthode d'écriture non destructive, pour conserver les données existantes sur votre clé USB et / ou la prise en charge de data persistence, vous pouvez utiliser **livecd-iso-to-disk**.

Clonage d'une machine complète

Le clonage de disque est le processus de création d'une image d'une partition ou d'un disque dur entier. Cela peut être utile pour copier le lecteur sur d'autres ordinateurs ou à des fins de sauvegarde et de restauration.

Cela reste aussi une très bonne façon de changer de machine sans avoir à tout réinstaller.

Il existe une multitude de solution logiciel tout en un ou bien des livecd permettant de réaliser l'opération manuellement.
Nous allons voir ici comme réaliser l'opération via **dd** et **rsync**.

Clonage d'une machine complète

La première opération consiste à récupérer les informations de partition de la machine.

```
parted -l > disk.bck
cat /etc/fstab > fstab.bck
```

Nous avons maintenant les tailles de partition ainsi que l'ordre de montage.
Pour créer une image de chaque partition nous passons par dd :

```
dd if=/dev/sda5 of=/mnt/backup_drive/sdaX_root.img
```

On répète l'opération en fonction de vos partitions.

Clonage d'une machine complète

Vous remarquerez que la taille de vos partitions n'implique pas que tout l'espace soit utilisé. Il est possible de réduire la taille de celle-ci via la commande **resize2fs**

```
resize2fs -M sdX_root.img
```

L'utilitaire vous demandera de lancer **e2fsck** pour vérifier le disque. **e2fsck** produira nécessairement beaucoup d'erreurs ou de correctifs.

Cela est dû au fait que les informations du système de fichiers ne sont plus correctes en ce qui concerne le début et la fin des limites de partition.

Clonage d'une machine complète

Sur le nouveau disque , après avoir préparer les partitions d'accueil, il faut réaliser l'opération inverse avec dd.
Pas d'inquiétude si la partition d'accueil est plus grande que celle d'origine. Avec **resize2fs** il sera possible de l'étendre à nouveau.

Il reste à monter la partition **/root** (et la **/boot** si il y avait) pour réinstaller le grub sur l'entête du disk.
En admettant un montage de **/root** sur **/mnt/root/**

```
chroot /mnt/tmp && grub-install /dev/XXX
```

Les partitions ayant été copié va dd, elles conservent leurs UUID. Il ne vous restent plus qu'à vous assurer de leurs correspondance avec ce qui se trouve dans le **fstab**

Clonage d'une machine complète

/!\ Notes : au fil du temps, les systèmes de fichiers obtiennent de nouvelles fonctionnalités et les utilitaires **mkfs** modifient leurs valeurs par défaut, mais toutes les nouvelles fonctionnalités ne peuvent pas être activées sans reformatage.

Ainsi, lorsque vous déplacez des données vers un nouveau lecteur, au lieu de cloner les périphériques blocs ou les systèmes de fichiers, pensez à créer un nouveau système de fichiers et ne copiez que les fichiers (et leurs attributs, ACL, attributs étendus, etc.) avec par exemple **rsync** .

02
Maîtriser la configuration logicielle du système

Package RPM

Le gestionnaire de packages RPM (RPM) est un système de gestion de packages qui s'exécute sur Red Hat Enterprise Linux, CentOS et Fedora.

De nombreux éditeurs de logiciels distribuent leurs logiciels via un fichier d'archive conventionnel (comme une archive tar).

Cependant, il existe plusieurs avantages à empaqueter un logiciel dans des packages RPM.

Package RPM

Avantages des paquets RPM

- Installer, réinstaller, supprimer, mettre à niveau et vérifier les packages
- Utilisez une base de données de packages installés pour interroger et vérifier les packages
- Utilisez des métadonnées pour décrire les packages, leurs instructions d'installation, etc.
- Empaqueter les sources logicielles vierges en paquets source et binaires
- Signez numériquement vos paquets

Structure du fichier de package RPM

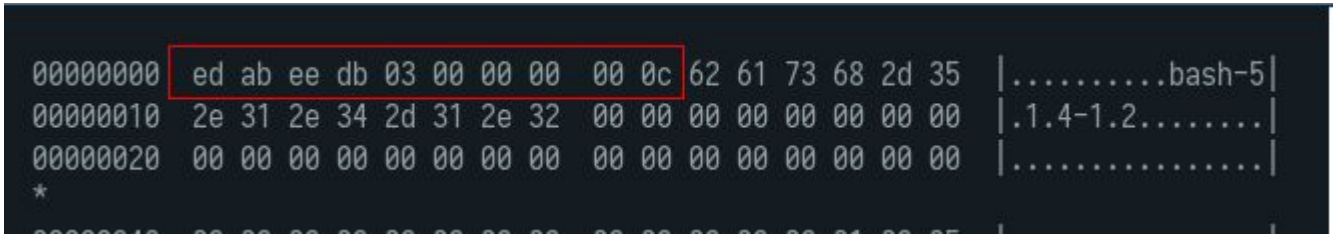
Les packages RPM sont livrés avec un fichier par package. Tous les fichiers RPM ont le format de base suivant de quatre sections:

- Un identifiant de fichier
- Une signature
- Informations d'en-tête
- Les fichiers à installer sous la forme d’une archive

Structure du fichier de package RPM

L'identifiant du fichier

Aussi appelé lead ou rpmlead, l'identifiant indique que ce fichier est un fichier RPM. Il contient un nombre magique que la commande **file** utilise pour détecter les fichiers RPM.
Il contient également des informations sur la version et l'architecture ainsi que la nature des fichiers contenu (Binaire ou Source)



Un exemple ici avec un fichier RPM pour architecture ARM
Ici ARM est représenté par **0c** en hexa ou **12** en décimal

Structure du fichier de package RPM

La structure de l’entête

La section identifiant ne contient plus suffisamment d'informations pour décrire les RPM modernes.

De plus, la section identifiant est loin d'être aussi flexible que l'exigent les packages actuels.

Pour contrer ces lacunes, la section d'en-tête a été introduite pour inclure plus d'informations sur le package.

Structure du fichier de package RPM

La signature

La signature apparaît après la section de l'identifiant. La signature RPM permet de vérifier l'intégrité du package et éventuellement l'authenticité.

La signature fonctionne en exécutant une fonction mathématique sur l'en-tête et la section d'archive du fichier. La fonction mathématique peut être un processus de chiffrement, tel que PGP (Pretty Good Privacy), ou un résumé de message au format MD5 (hash)

Structure du fichier de package RPM

La charge utile

La section payload, ou archive, contient les fichiers réellement utilisés dans le package. Ce sont les fichiers que la commande rpm installe lorsque vous installez le package.
Pour économiser de l'espace, les données de la section archive sont compressées au format GNU **gzip**.

Une fois décompressées, les données sont au format **cpio**, c'est ainsi que la commande **rpm2cpio** peut faire son travail. Au format cpio, la charge utile est constituée d'enregistrements, un par fichier.

Structure du fichier de package RPM

Pour aller plus loin dans le détail ..

Dans le but de ne pas surcharger cette présentation , voici deux documentations détaillant la structure des paquet RPM :

https://docs.fedoraproject.org/en-US/Fedora_Draft_Documentation/0.1/html/RPM_Guide/ch-package-structure.html

https://refspecs.linuxbase.org/LSB_5.0.0/LSB-Core-generic/LSB-Core-generic/pkgformat.html

https://rpm-packaging-guide.github.io/

Exécutable et librairies

Dans la charge utile d'un paquet RPM il faudra faire la différence entre les librairies et les binaires.

Le binaire contient le code utile et spécifique au programme quand les librairies contiennent un ensemble de fonctions et variable pouvant être commun à différent programme.

Un binaire peut être installé avec l'ensemble des librairies lui permettant de fonctionner ou bien de façon séparé. On parle alors de binaire utilisant des librairies partagés.

Exécutable et librairies

Concept de bibliothèques partagées

Semblables à leurs équivalents physiques, les bibliothèques de logiciels sont des collections de code qui sont destinées à être utilisées par de nombreux programmes différents; tout comme les bibliothèques physiques conservent des livres et d'autres ressources qui peuvent être utilisés par de nombreuses personnes différentes.

Pour créer un fichier exécutable à partir du code source d'un programme :

- **La compilation**
- **l’édition de liens**

Exécutable et librairies

Concept de bibliothèques partagées

Tout d'abord, le compilateur transforme le code source en code machine qui est stocké dans des fichiers objets .

Deuxièmement, l'éditeur de liens combine les fichiers objets et les lie à des bibliothèques afin de générer le fichier exécutable final.

Cette liaison peut se faire de manière statique ou dynamique

Exécutable et librairies

Conventions de dénomination des fichiers d'objets partagés

Le nom d'une bibliothèque partagée, également appelée soname , suit un modèle composé de trois éléments:

- Nom de la bibliothèque (normalement préfixé par lib)
- so (qui signifie «objet partagé»)
- Numéro de version de la bibliothèque

Voici un exemple: libpthread.so.0

Exécutable et librairies

Conventions de dénomination des fichiers d'objets partagés

La glibc(Bibliothèque GNU C) est un bon exemple de bibliothèque partagée.

Sur un système Debian GNU / Linux 9.9, son fichier est nommé libc.so.6.

Ces noms de fichiers plutôt généraux sont normalement des liens symboliques qui pointent vers le fichier réel contenant une bibliothèque, dont le nom contient le numéro de version exact. Dans le cas de glibc, ce lien symbolique ressemble à ceci:

```
$ ls -l /lib/x86_64-linux-gnu/libc.so.6
lrwxrwxrwx 1 root root 12 feb  6 22:17 /lib/x86_64-linux-gnu/libc.so.6 ->
libc-2.24.so
```

Exécutable et librairies

Configuration des chemins de bibliothèque partagés

lorsque le programme est exécuté. L'éditeur de liens dynamique (ld.so ou ld-linux.so) recherche des bibliothèques dans un certain nombre de répertoires.

Les emplacements communs pour les bibliothèques partagées dans un système Linux sont:

- /lib
- /lib32
- /lib64
- /usr/lib
- /usr/local/lib

Exécutable et librairies

Configuration des chemins de bibliothèque partagés

Comme la plupart des autres outils Linux, la liaison dynamique est également configurée à l'aide d'un fichier de configuration de texte.

Il se trouve dans **/etc/ld.so.conf** . Sur la plus part des distribution, il ne fait que pointer vers d'autres fichiers de configuration, **/etc/ld.so.conf.d/**

La commandes **ldconfig** traitent tous ces fichiers pour accélérer le chargement des bibliothèques. Cette commande crée **ld.so.cache** pour localiser les fichiers qui doivent être chargés et liés dynamiquement.

Exécutable et librairies

LD_LIBRARY_PATH

Parfois, vous devez remplacer les bibliothèques installées d'origine et utiliser la vôtre ou une bibliothèque spécifique. Les cas peuvent être:

- Vous exécutez un ancien logiciel qui nécessite une ancienne version d'une bibliothèque.
- Vous développez une bibliothèque partagée et souhaitez tester sans l'installer
- Vous exécutez un programme spécifique (par exemple depuis opt) qui doit accéder à ses propres bibliothèques

Exécutable et librairies

LD_LIBRARY_PATH

dans ces cas, vous devez utiliser la variable d'environnement **LD_LIBRARY_PATH** .

Une liste de répertoires séparés par collon (:) indiquera à votre programme où rechercher les bibliothèques nécessaires avant de vérifier les bibliothèques dans ld.so.cache.

Exemple :
LD_LIBRARY_PATH=/usr/local/mylib
(penser à lancer la commande avec export si vous souhaitez rendre la variable utilisable dans votre shell courant)

Exécutable et librairies

De quelles bibliothèques j'ai besoin ?

Pas de panique, nous n'allons pas explorer tous ikea pour cela.

La commande ldd vous aide à trouver:

- Si un programme est lié dynamiquement ou statiquement
- De quelles bibliothèques un programme a-t-il besoin?

Exemple :

```
ldd /usr/bin/ls
linux-vdso.so.1 (0x00007ffcbe37e000)
libcap.so.2 => /usr/lib/libcap.so.2 (0x00007fbb23b33000)
libc.so.6 => /usr/lib/libc.so.6 (0x00007fbb23966000)
/lib64/ld-linux-x86-64.so.2 => /usr/lib64/ld-linux-x86-64.so.2 (0x00007fbb23b8c000)
```

Construction d'un package RPM à partir des sources

La construction d’un package RPM permet d’intégrer un logiciel à un dépôt de paquet et ainsi bénéficier d’une intégration aux distribution visé.

Dans le but de ne pas alourdir cette présentation , la suite sera détaillé en suivant la documentation Fedora :

Construction d'un paquet RPM

Mise en place d'un miroir de paquets local

Objectif d’un dépôt local :

- Se passer d’un dépôt distant
- Diminuer le temps et la bande passante consommée par des mise-à-jour et des installations
- Offrir des dépôts de paquets supplémentaires

Cette opération est relativement simple , vous pouvez suivre les procédures suivante :

- Fedora
- Centos

Mises à jour du système et patchs de sécurité

Sur un serveur Linux, vous pouvez mettre à jour votre système en installant seulement les mises à jour de sécurité.

Cette possibilité est intéressante pour protéger votre système, d'autant plus s'il héberge une application critique, sans pour autant mettre à jour l'intégralité des paquets.

Les méthodes pour se faire sont relativement simple car gérée par les gestionnaires de paquets. Nous allons voir l’exemple de fedoras et centos.

Mises à jour du système et patchs de sécurité

Installer les patchs de sécurité sous fedoras

Il faut tout d'abord installer le greffon de DNF et mettre à jour votre version de Fedora actuelle.

```
dnf install dnf-plugin-system-upgrade
dnf upgrade && dnf clean all
```

Ensuite vous pouvez télécharger les paquets puis redémarrer la machine pour appliquer la mise à niveau.

```
dnf system-upgrade download --releasever=(Version)
dnf system-upgrade reboot
```

Mises à jour du système et patchs de sécurité

Installer les patchs de sécurité sous CentOS

la commande ci-dessous sert à lister les mises à jour de sécurité disponibles pour la machine CentOS :

(sudo) yum updateinfo list updates security

Puis pour n'installer que les package de sécurité :

(sudo) yum update --security

Ce sont les metadata contenu dans le paquet qui permette cette sélection.

03
Filesystems et unités de stockage

Avantages et inconv. de différents systèmes de fichiers

Avant de se jeter dans un comparatif il est bon de se demander l'usage que l'on vas avoir de ce système de fichier.

- Pc bureautique ?
- Serveur de donnée ?
- Serveur haut dispo ?
- Cluster ?

En fonction de ce point de vue il sera plus simple de définir le système de fichier qui convient.

Ainsi la première vrai grand question, avec ou sans journalisation ?

Avantages et inconv. de différents systèmes de fichiers

Journalisation

La journalisation est conçue pour empêcher la corruption des données due aux pannes et aux coupures de courant soudaines. Disons que votre système est à mi-chemin de l'écriture d'un fichier sur le disque et qu'il perd soudainement de l'énergie.

Sans journal, votre ordinateur n'aurait aucune idée si le fichier était entièrement écrit sur le disque. Le fichier resterait sur le disque mais corrompu.

Avantages et inconv. de différents systèmes de fichiers

Journalisation

Avec un journal, le système enregistre, dans le journal, qu'il allait écrire un certain fichier sur le disque. Une fois l’action en elle même réalisé l’information est mise à jour dans le journal.

Si l'alimentation est coupée pendant l'écriture du fichier, Linux vérifiera le journal du système de fichiers au démarrage et reprendra les travaux partiellement terminés.

Cela évite la perte de données et la corruption de fichiers.

Avantages et inconv. de différents systèmes de fichiers

Journalisation

La journalisation ralentit un peu les performances d'écriture sur disque, mais cela en vaut la peine sur un ordinateur de bureau ou un ordinateur portable.

Seules les métadonnées du fichier, l'inode (emplacement sur le disque) sont enregistrés dans le journal avant d'être écrits.

Les systèmes de fichiers qui n'offrent pas de journalisation sont disponibles pour une utilisation sur des serveurs hautes performance

Ils sont également idéaux pour les lecteurs flash amovibles.

Avantages et inconv. de différents systèmes de fichiers

Ext

Ext signifie «Système de fichiers étendu», et a été le premier créé spécifiquement pour Linux. Il a eu quatre révisions majeures.

«Ext» est la première version du système de fichiers, introduite en 1992.

Il s'agissait d'une mise à jour majeure du système de fichiers Minix utilisé à l'époque, mais il manque des fonctionnalités importantes.

De nombreuses distributions Linux ne prennent plus en charge Ext.

Avantages et inconv. de différents systèmes de fichiers

Ext2

Ext2 n'est pas un système de fichiers journalisé. Lors de son introduction, il s'agissait du premier système de fichiers à prendre en charge les attributs de fichier étendus et les lecteurs de 2 téraoctets.

Le manque de journal d'Ext2 signifie qu'il écrit moins sur le disque, ce qui le rend utile pour la mémoire flash comme les clés USB.

Cependant, les systèmes de fichiers comme exFAT et FAT32 n'utilisent pas non plus la journalisation et sont plus compatibles avec différents systèmes d'exploitation, nous vous recommandons donc d'éviter Ext2 à moins que vous ne sachiez que vous en avez besoin pour une raison quelconque (Compatibilité avec Openbsd, par exemple)

Avantages et inconv. de différents systèmes de fichiers

Ext3

Ext3 est fondamentalement juste Ext2 avec journalisation. Ext3 a été conçu pour être rétrocompatible avec Ext2, permettant aux partitions d'être converties entre Ext2 et Ext3 sans aucun formatage requis.

Il existe depuis plus longtemps qu'Ext4, mais Ext4 existe depuis 2008 et est largement testé. À ce stade, il vaut mieux utiliser Ext4.

Avantages et inconv. de différents systèmes de fichiers

Ext4

Ext4 a également été conçu pour être rétrocompatible. Vous pouvez monter un système de fichiers Ext4 en tant qu'ext3, ou monter un système de fichiers Ext2 ou Ext3 en tant qu'ext4.

Il inclut des fonctionnalités plus récentes qui réduisent la fragmentation des fichiers, permet des volumes et des fichiers plus importants et utilise l'allocation différée pour améliorer la durée de vie de la mémoire flash.

Il s'agit de la version la plus moderne du système de fichiers Ext et de la version par défaut sur la plupart des distributions Linux.

Avantages et inconv. de différents systèmes de fichiers

XFS

XFS a été développé par Silicon Graphics en 1994 pour le système d'exploitation SGI IRX, et a été porté sous Linux en 2001.

Il est similaire à Ext4 à certains égards, car il utilise également l'allocation différée pour aider à la fragmentation des fichiers et ne permet pas les instantanés montés.
Il peut être agrandi, mais pas rétréci, à la volée. XFS a de bonnes performances lorsqu'il s'agit de fichiers volumineux, mais a de moins bonnes performances que les autres systèmes de fichiers lorsqu'il s'agit de nombreux petits fichiers.
Cela peut être utile pour certains types de serveurs qui doivent principalement traiter des fichiers volumineux.

Avantages et inconv. de différents systèmes de fichiers

JFS

JFS , ou «Journaled File System», a été développé par IBM pour le système d'exploitation IBM AIX en 1990, puis porté sous Linux. Il bénéficie d'une faible utilisation du processeur et de bonnes performances pour les gros et petits fichiers.

Les partitions JFS peuvent être redimensionnées dynamiquement, mais pas réduites. Il a été extrêmement bien planifié et prend en charge la plupart des distributions majeures, mais ses tests de production sur des serveurs Linux ne sont pas aussi étendus que Ext, car il a été conçu pour AIX. Ext4 est plus couramment utilisé et est plus largement testé.

Avantages et inconv. de différents systèmes de fichiers

BtrFS

BtrFS , prononcé «Butter» ou «Better» FS, a été conçu à l'origine par Oracle. Il signifie «B-Tree File System» et permet le regroupement de lecteurs, des instantanés à la volée, une compression transparente et une défragmentation à chaud.

Il partage un certain nombre des mêmes idées que celles trouvées dans ReiserFS, un système de fichiers que certaines distributions Linux utilisent par défaut.

Ted Ts'o, le mainteneur du système de fichiers Ext4, considère Ext4 comme une solution à court terme et pense que BtrFS est la voie à suivre .

Avantages et inconv. de différents systèmes de fichiers

Conclusions

XFS - Est le meilleur choix pour le serveur et l'environnement de stockage de données (Et de large fichier) . Difficile d'avoir des données corrompues en raison d'une panne de courant, d'une panne système ou d'une panne matérielle.

JFS - Répond aux attentes. Utilise moins de ressources CPU que les autres systèmes de fichiers . Il est également difficile d'avoir des données corrompues en raison d'une panne de courant, d'un crash système ou d'une panne matérielle.

EXT4 - Stable et éprouvé. Bonnes performances et maniabilité . Bon choix pour une utilisation de bureau, même pour un environnement serveur dans certains cas.

Récupération des données perdues accidentellement

Avant de commencer

Ces slide sont essentiellement à des fins éducatives. Si vous avez accidentellement supprimé ou endommagé des données précieuses et irremplaçables sans aucune expérience en récupération de données, éteignez immédiatement l'ordinateur et préférer une récupération en chambre blanche (Manipulation physique du disque)

Il est aussi important de réaliser un backup du disque entier (Quand celui-ci n'a pas de soucis matériel) via l'utilitaire **ddrescue** ou **dd_rescue** en ayant redémarré sur un livecd

Toute manipulation (même de lecture) du disque directement peut aggraver votre situation mais surement pas l'améliorer.

Récupération des données perdues accidentellement

Les utilitaires **ddrescue** et **dd_rescue**, contrairement à **dd** qui agit en un seul passage, vont essayer de récupérer les erreurs à plusieurs reprises en lisant le support du début à la fin, puis de nouveau au début en essayant de récupérer les données.

Ils vont créer des fichiers logs pour que la récupération puisse être interrompue et reprise sans perdre les acquis.

Nous pourrons ensuite monté l'image en read-only et exploiter une récupération de données.

Récupération des données perdues accidentellement

Foremost

Foremost est un programme, en mode console, de récupération de fichiers selon les en-têtes et fins de fichier, et structures internes des données, ce qu'il est habituel de regrouper comme "data carving".

Foremost peut travailler sur des fichiers d'image disque (tels que ceux générés par dd) ou directement sur un lecteur.

Les types d'en-têtes (headers) et fins de fichiers (footers) peuvent être définis dans le fichier de configuration ou par des des commutateurs/options dans la ligne de commande, spécifiant des types de fichiers. Cette recherche sur des structures de données d'un format pré-défini, permet une récupération plus fiable et plus rapide.

Récupération des données perdues accidentellement

Scalpel

Scalpel est un programme, en mode console, de recherche via data-carving.

Basé initialement sur Foremost , quoique bien plus efficace. Initialement développé par Golden G. Richard III, il permet à un analyste de préciser des critères d'en-têtes et fins de fichiers pour recouvrer les fichiers de certains types à partir d'une partie d'un media.

Récupération des données perdues accidentellement

Extundelete

Extundelete est un utilitaire en mode console conçu pour récupérer des fichiers supprimés dans des partitions ext3 et ext4.

Il peut récupérer tous les fichiers récemment supprimés d'une partition donnée et/ou des fichiers spécifiques selon les chemins relatifs ou les valeurs d'inode.

Notez que cela fonctionne uniquement sur une partition démontée.

Les fichiers récupérés sont enregistrés dans le répertoire courant dans un dossier nommé RECOVERED_FILES/ .

Récupération des données perdues accidentellement

Testdisk et PhotoRec

TestDisk et PhotoRec sont les deux utilitaires de récupération de données open-source.

TestDisk est principalement conçu pour la récupération de partitions perdues et/ou rendre à nouveau amorçables (bootable) des disques ayant perdu cette qualité du fait d'un logiciel défectueux, de certains types de virus, ou d'une erreur humaine comme la suppression accidentelle des tables de partition.

PhotoRec est un logiciel de récupération des fichiers perdus, des photographies , des vidéos, des documents, des archives de disques durs ou d'ISO

Récupération des données perdues accidentellement

Récupération de fichier texte

Il est possible de trouver des fichiers texte supprimés sur un disque dur en les recherchant directement sur le périphérique. Il sera nécessaire de créer une chaîne de préférence unique à partir du fichier que vous essayez de récupérer.

Utiliser grep pour rechercher des chaînes fixes (-F) directement sur la partition:

```
grep -a -C 200 -F 'Une ligne du fichier' /dev/sdXN > OutputFile
```

Note : L'option -C 200 demande à grep d'afficher 200 lignes de contexte entourant (avant et après) chaque occurrence de la chaîne.

Remédier aux problèmes

Comme vus précédemment , les systèmes de fichiers Linux modernes sont journalisés. Cela signifie que chaque opération est enregistrée dans un journal interne avant son exécution. Si l'opération est interrompue en raison d'une erreur système (comme une panique du noyau, une panne de courant, etc.), elle peut être reconstruite en vérifiant le journal, évitant ainsi la corruption du système de fichiers et la perte de données.

Cela réduit considérablement le besoin de vérifications manuelles du système de fichiers, mais elles peuvent encore être nécessaires.

Remédier aux problèmes

Vérification de l'utilisation du disque

Il existe deux commandes qui peuvent être utilisées pour vérifier combien d'espace est utilisé et combien il reste sur un système de fichiers. Le premier est **du**, qui signifie «disk usage».

du est de nature récursive. Dans sa forme la plus basique, la commande montrera simplement la taille de chaque fichier dans chaque répertoire.

Pour obtenir un résumé de la taille global :
du -sh

Ou sans les sous-répertoire
du -Sh

Remédier aux problèmes

Vérification de l'espace libre

du fonctionne au niveau des fichiers. Il existe une autre commande qui peut vous montrer l'utilisation du disque et la quantité d'espace disponible au niveau du système de fichiers. Cette commande est **df**.

La commande df fournira une liste de tous les systèmes de fichiers disponibles (déjà montés) sur votre système, y compris leur taille totale, combien d'espace a été utilisé, combien d'espace est disponible, le pourcentage d'utilisation et où il est monté.

Remédier aux problèmes

Maintenance des systèmes de fichiers ext2, ext3 et ext4

Pour vérifier un système de fichiers pour les erreurs (et, espérons-le, les corriger), Linux fournit l'utilitaire **fsck** (pensez à «filesystem check» et vous n'oublierez jamais le nom).

Dans sa forme la plus basique, vous pouvez l'invoquer avec **fsck** suivi de l'emplacement du système de fichiers que vous souhaitez vérifier:

```
fsck /dev/sdb1
fsck from util-linux 2.33.1
e2fsck 1.44.6 (5-Mar-2019)
DT_2GB: clean, 20/121920 files, 369880/487680 blocks
```

Remédier aux problèmes

fsck lui-même ne vérifiera pas le système de fichiers, il appellera simplement l'utilitaire approprié pour le type de système de fichiers pour le faire.

Dans l'exemple ci-dessus, comme aucun type de système de fichiers n'a été spécifié, fsck a supposé un système de fichiers ext2/3/4 par défaut et appelé e2fsck.

Pour spécifier un système de fichiers, utilisez l'option **-t**, suivie du nom du système de fichiers, comme dans **fsck -t vfat /dev/sdc**.

Vous pouvez également appeler directement un utilitaire spécifique **fsck.msdos** au système de fichiers , comme pour les systèmes de fichiers FAT.

Remédier aux problèmes

Réglage fin d'un système de fichiers ext

Les systèmes de fichiers ext2, ext3 et ext4 ont un certain nombre de paramètres qui peuvent être ajustés, ou «réglés», par l'administrateur système pour mieux répondre aux besoins du système.

L'utilitaire utilisé pour afficher ou modifier ces paramètres est appelé tune2fs.

Pour voir les paramètres actuels d'un système de fichiers donné, utilisez le paramètre **-l** suivi du périphérique représentant la partition.

Remédier aux problèmes

Réglage fin d'un système de fichiers ext

Les systèmes de fichiers ext ont un nombre de montages . Le nombre est augmenté de 1 à chaque fois que le système de fichiers est monté, et lorsqu'une valeur seuil (le nombre maximal de montages) est atteinte, le système sera automatiquement vérifié **e2fsck** au prochain démarrage.

Ces une des options modifiable via **tune2fs**.

Les systèmes de fichiers ext3 sont essentiellement des systèmes de fichiers ext2 avec un journal. En utilisant, **tune2fs** vous pouvez ajouter un journal à un système de fichiers ext2, le convertissant ainsi en ext3. La procédure est simple, il suffit de passer le **-j** paramètre à tune2fs, suivi du périphérique contenant le système de fichiers.

Remédier aux problèmes

Maintenance des systèmes de fichiers XFS

Pour les systèmes de fichiers XFS, l'équivalent de fsck est xfs_repair. Si vous pensez que quelque chose ne va pas avec le système de fichiers, la première chose à faire est de le scanner pour détecter tout dommage.

Cela peut être fait en passant le -n paramètre à **xfs_repair**, suivi du périphérique contenant le système de fichiers. Le paramètre **-n** signifie «pas de modification»

le système de fichiers sera vérifié, des erreurs seront signalées mais aucune réparation ne sera effectuée.

Remédier aux problèmes

Maintenance des systèmes de fichiers XFS

Pour déboguer un système de fichiers XFS, l'utilitaire xfs_db peut être utilisé, comme dans xfs_db /dev/sdb1. Ceci est principalement utilisé pour inspecter divers éléments et paramètres du système de fichiers.

Cet utilitaire a une invite interactive, comme parted, avec de nombreuses commandes internes

Un autre utilitaire utile est xfs_fsr, qui peut être utilisé pour réorganiser («défragmenter») un système de fichiers XFS.

Copie d'un disque système complet à chaud

La duplication de disque possède une règle simple : Ne pas avoir d'activité sur un disque qui est en cours de copie.
Le soucie avec la copie du disque système c’est justement qu’il a de l’activité.

Ainsi réalisé un **dd** puis un **fsck** reste une méthode fonctionnel mais avec certains risques de corruption de données.

La méthode pour pouvoir réaliser une copie de **/root** sans avoir d’activité système dessus et de pivoter **/root** en ram le temps de la copie !

Et pour ça il existe une command du nom de … **pivot_root**.

Copie d'un disque système complet à chaud

Pour se faire nous allons donc migrer dans la mémoire le temps de l'opération.

On commence par créer un répertoire en mémoire et préparer l'environnement :

```
mkdir /tmp/tmproot
mount none /tmp/tmproot -t tmpfs
mkdir /tmp/tmproot/{proc,sys,usr,var,oldroot}
cp -ax /{bin,etc,mnt,sbin,lib} /tmp/tmproot/
cp -ax /usr/{bin,sbin,lib} /tmp/tmproot/usr/
cp -ax /var/{account,empty,lib,local,lock,nis,opt,preserve,run,spool,tmp,yp}
/tmp/tmproot/var/
cp -a /dev /tmp/tmproot/dev
```

Copie d'un disque système complet à chaud

Stopper tous les services tournant actuellement sur le système (sauf ssh si vous êtes à distance)

Réaliser le pivot :

```
pivot_root /tmp/tmproot/ /tmp/tmproot/oldroot
mount none /proc -t proc
mount none /sys -t sysfs
mount none /dev/pts -t devpts
```

Il vous reste à redémarrer ssh , vous y connecter et fermer l'ancienne connexion.
Vous pouvez maintenant réaliser une copie via dd de votre partition **/root**

LVM : modes linéaire, stripping, mirroring, les snapshots

Dans LVM, un groupe de volumes est divisé en volumes logiques.

Il y a trois types de volumes logiques LVM :

- Les volumes linéaires,
- Les volumes en mode stripe
- Les volumes en miroir.

Voyons tout cela ensemble.

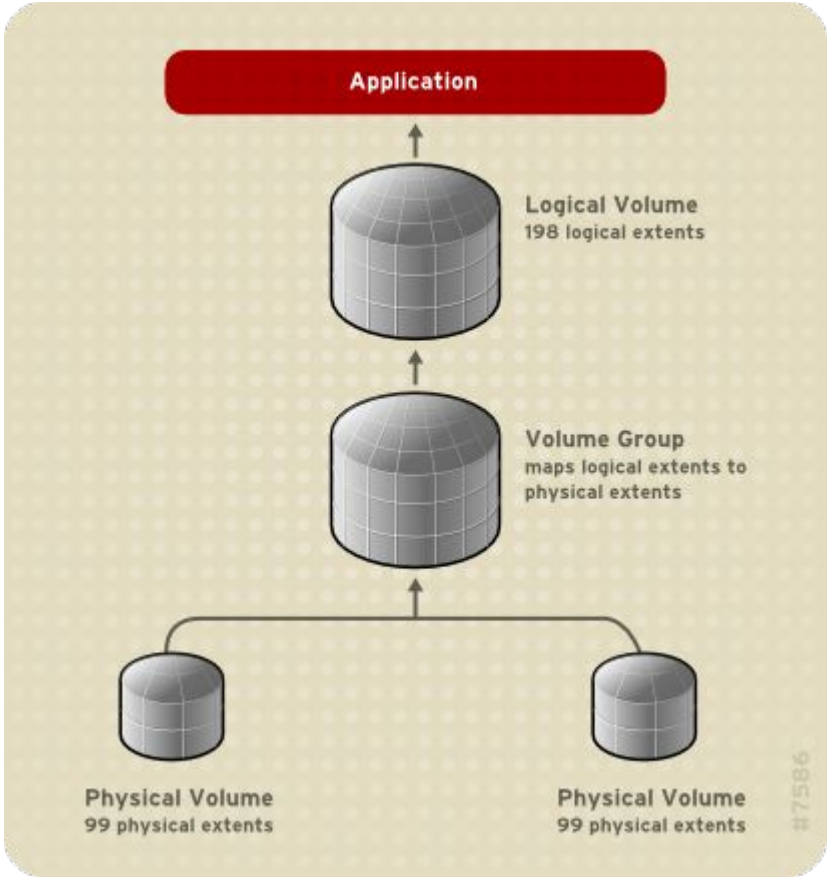
LVM : modes linéaire, stripping, mirroring, les snapshots

Les volumes linéaires

Un volume linéaire regroupe plusieurs volumes physiques dans un volume logique.

Si vous avez par exemple deux disques de 60Go, vous pouvez créer un volume logique de 120Go.

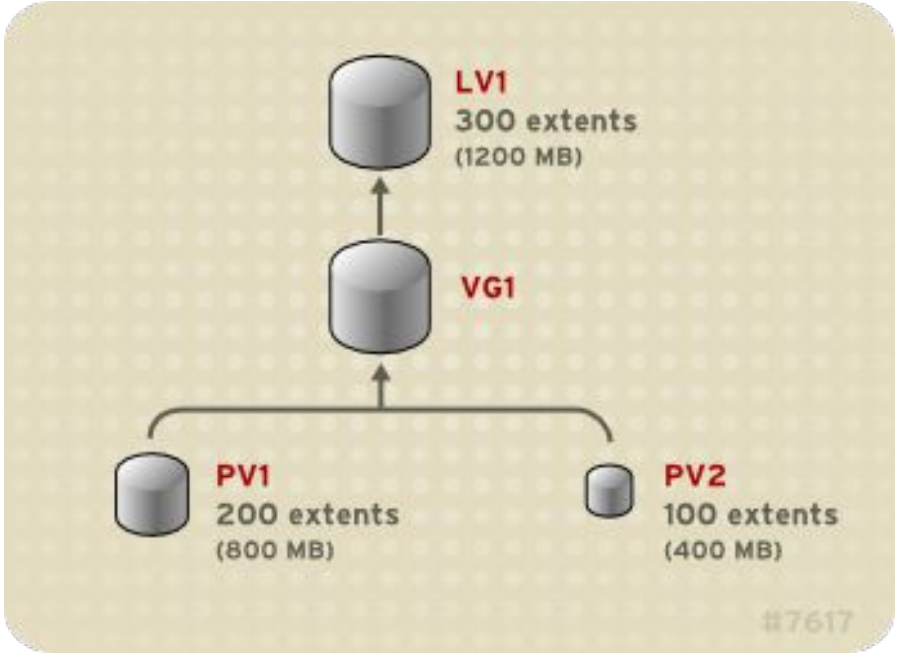
Le stockage physique est concaténé.



LVM : modes linéaire, stripping, mirroring, les snapshots

Les volumes linéaires

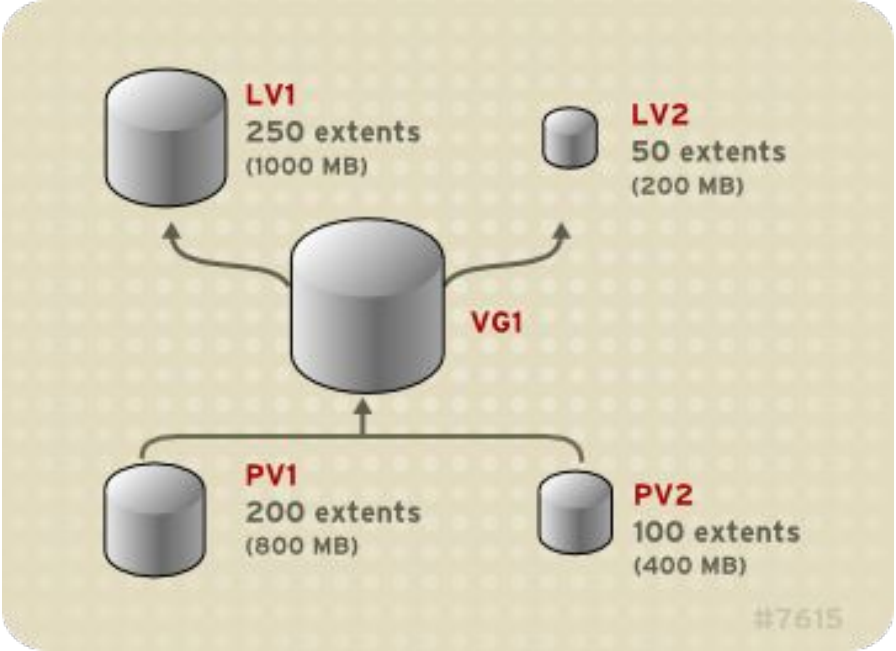
Les volumes physiques qui composent un volume logique ne doivent pas nécessairement être de la même taille.



LVM : modes linéaire, stripping, mirroring, les snapshots

Les volumes linéaires

Vous pouvez configurer plusieurs volumes logiques linéaires de la taille souhaitée à partir du pool d'extensions physiques.



LVM : modes linéaire, stripping, mirroring, les snapshots

Les volumes logiques en mode stripe

Lorsque vous écrivez des données sur un volume logique LVM, le système de fichiers disperse les données sur les volumes physiques sous-jacents.

Vous pouvez contrôler la façon dont les données sont écrites sur les volumes physiques en définissant un volume logique en mode stripe.

Pour les lectures et écritures séquentielles volumineuses, cela peut améliorer l'efficacité des E/S de données.

LVM : modes linéaire, stripping, mirroring, les snapshots

Les volumes logiques en mode stripe

Le striping améliore les performances en écrivant des données sur un nombre prédéterminé de volumes physiques avec une technique round-round.

Avec le striping, les E/S peuvent être faites en parallèle.

Dans certaines situations, cela peut résulter en un gain de performances quasiment linéaire pour chaque volume physique supplémentaire dans le stripe.

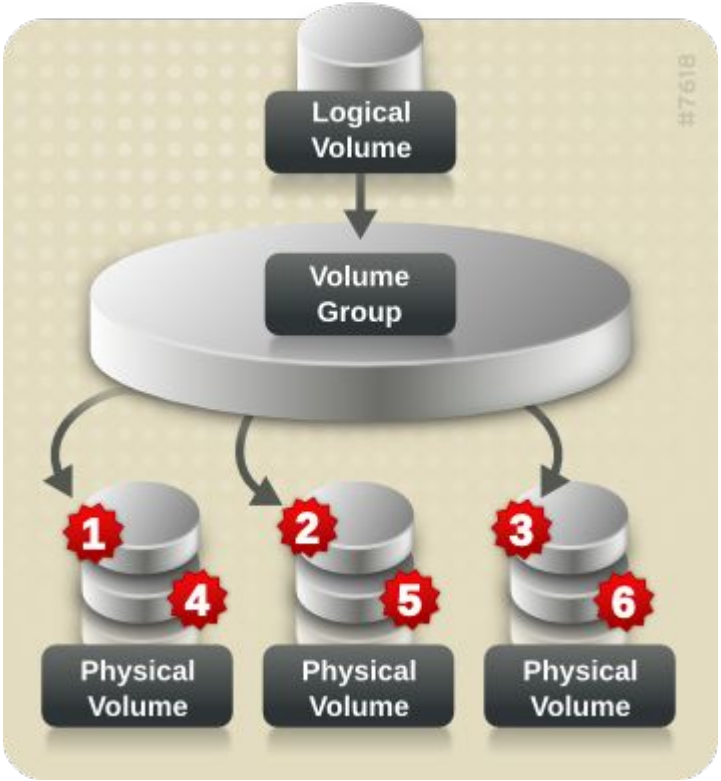
LVM : modes linéaire, stripping, mirroring, les snapshots

Les volumes logiques en mode stripe

Dans cette figure:

- le premier stripe de données est écrit sur PV1
- le second stripe de données est écrit sur PV2
- le troisième stripe de données est écrit sur PV3
- le quatrième stripe de données est écrit sur PV1

Dans un volume logique en mode stripe, la taille du stripe ne peut pas excéder la taille d'une extension.



LVM : modes linéaire, stripping, mirroring, les snapshots

Les volumes logiques en miroir

Un miroir gère des copies identiques de données sur différents périphériques. Lorsque les données sont écrites sur un périphérique, elles sont répliquées sur un deuxième périphérique, définissent ainsi un miroir.

Cela assure la protection des échecs de périphériques. Lorsqu'une section du miroir échoue, le volume logique devient un volume linéaire et reste accessible.

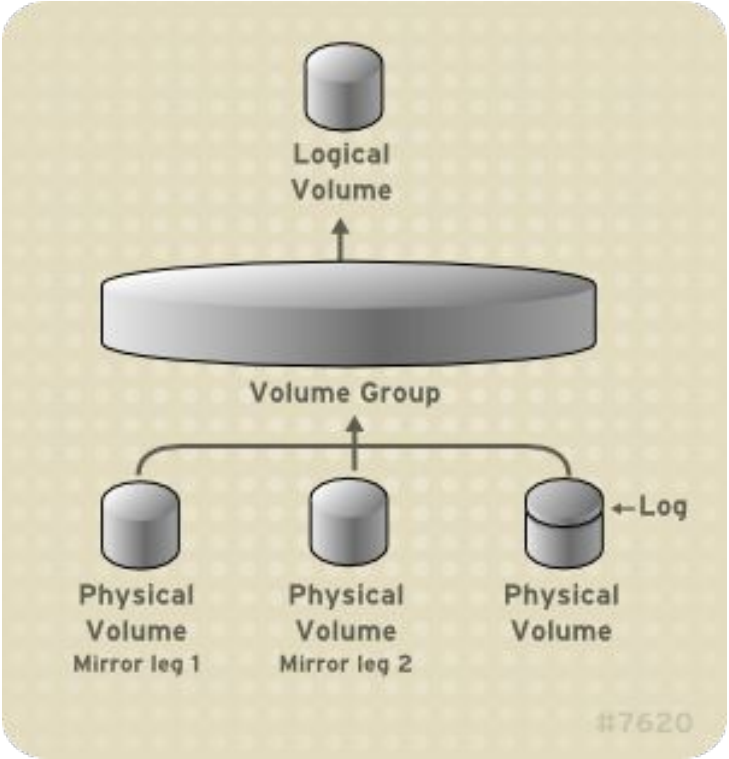
Un miroir LVM divise le périphérique qui est copié en zones qui ont généralement une taille de 512Mo.
LVM maintient un petit fichier journal afin d'assurer le suivi des zones synchronisées avec le ou les miroirs.

LVM : modes linéaire, stripping, mirroring, les snapshots

Les volumes logiques en miroir

Dans cette figure nous avons un PV qui est en miroir et un autre PV qui héberge le Log.

Si l'un des deux miroirs tombe alors le second passe en mode linéaire.



LVM : modes linéaire, stripping, mirroring, les snapshots

Les snapshot ou les instantannés

La fonctionnalité relative aux instantanés LVM permet de créer des images virtuelles de périphériques à un moment donné sans provoquer l'interruption d'un service.

Lorsqu'un changement est effectué sur le périphérique d'origine après qu'un instantané ait été pris, la fonctionnalité relative aux instantanés LVM effectue une copie de la zone de données modifiée, telle qu'elle était avant son changement, afin de pouvoir reconstruire l'état du périphérique.

LVM : modes linéaire, stripping, mirroring, les snapshots

Les snapshot ou les instantannés

Vous pouvez utiliser la fonctionnalité relative aux instantanés de différentes manières :

- Généralement, un instantané est pris afin d'effectuer une sauvegarde sur un volume logique sans arrêter le système en cours qui met à jour les données de façon continue.

- Vous pouvez exécuter la commande fsck sur l'instantané d'un système de fichiers pour vérifier l'intégrité du système de fichiers et déterminer si le système de fichiers d'origine doit être réparé.

LVM : modes linéaire, stripping, mirroring, les snapshots

Les snapshot ou les instantannés

Vous pouvez utiliser la fonctionnalité relative aux instantanés de différentes manières :

- Étant donné que l'instantané est en lecture-écriture, vous pouvez tester les applications avec des données de production en prenant un instantané et en exécutant des tests sur cet instantané, sans modifier les données réelles.

04
Noyau et périphériques

La représentation des périphériques pour le noyau

/dev

/dev est l'emplacement des fichiers spéciaux ou de périphérique. C'est un répertoire très intéressant qui met en évidence un aspect important du système de fichiers Linux - tout est un fichier ou un répertoire.

Regardez dans ce répertoire et vous devriez, espérons-le, voir sda1, sda2, etc. qui représentent les différentes partitions sur le premier lecteur maître du système.
/dev/cdrom représentent votre lecteur de CD-ROM. Cela peut sembler étrange, mais cela aura du sens si vous comparez les caractéristiques des fichiers à celles de votre matériel.

Les deux peuvent être lus et écrits.

La représentation des périphériques pour le noyau

Périphériques /dev

Les disques durs :

```
ls -l /dev/sd*
brw-rw---- 1 root disk 8, 0 déc 15 23:40 /dev/sda
brw-rw---- 1 root disk 8, 1 déc 15 23:40 /dev/sda1
brw-rw---- 1 root disk 8, 2 déc 15 23:40 /dev/sda2
brw-rw---- 1 root disk 8, 5 déc 15 23:40 /dev/sda5
brw-rw---- 1 root disk 8, 16 déc 15 23:40 /dev/sdb
```

Où nous avons des fichiers de type block (b) avec un numéro primaire 8 et un numéro secondaire qui identifie la partitions pour le noyau.

La représentation des périphériques pour le noyau

les autres périphériques /dev

Ce dossier contient tous les périphériques matériels comme un lecteur cdrom, une carte son, une carte réseau, etc...

Il contient également les pseudo-périphériques. Quelques exemples :

- **/dev/zero** génére des zéros
- **/dev/random** génère de l'aléatoire
- **/dev/null** constitue un trou noir à octets, et notamment utilisé pour se débarrasser des fichiers et des affichages
- **/dev/loop0** permet de créer de faux périphériques de type block (stockage) à partir de fichiers créés avec la commande dd

La représentation des périphériques pour le noyau

les autres périphériques /dev

- **/dev/ttyS0** (premier port de communication, COM1)
  - Premier port série (souris, modems).
- **/dev/psaux** (PS / 2)
  - Connexion souris PS / 2 (souris, claviers).
- **/dev/lp0** (premier port d'imprimante, LPT1)
  - Premier port parallèle (imprimantes, scanners, etc.).
- **/ dev/dsp** (premier périphérique audio)
- **/dev/usb** (Périphériques USB)
  - Ce sous-répertoire contient la plupart des nœuds de périphérique USB.

La détection automatique du matériel

udev

udev (userspace / dev) est un gestionnaire de périphériques pour le noyau Linux . En tant que successeur de devfsd et hotplug, udev gère principalement les nœuds de périphériques dans le répertoire **/dev**.

Dans le même temps, udev gère également tous les événements de l' espace utilisateur déclenchés lorsque des périphériques matériels sont ajoutés au système ou supprimés de celui-ci, y compris le chargement du micrologiciel comme requis par certains périphériques.

La détection automatique du matériel

Fonctionnement de udev

Lorsque udev est informé par le noyau de l'apparition d'un nouveau périphérique, il récupère de nombreuses informations sur le périphérique en question en consultant les entrées correspondantes dans **/sys/**, en particulier celles qui permettent de l'identifier de manière unique (adresse MAC pour une carte réseau, numéro de série pour certains périphériques USB, etc.).

Armé de toutes ces informations,udev consulte l'ensemble de règles contenu dans **/etc/udev/rules.d/** et **/lib/udev/rules.d/** et décide à partir de cela du nom à attribuer au périphérique, des liens symboliques à créer (pour offrir des noms alternatifs), ainsi que des commandes à exécuter.

La détection automatique du matériel

Fonctionnement de udev

udev reçoit les messages du noyau et les transmet aux démons de sous-système tels que Network Manager.
Les applications parlent à Network Manager via D-Bus.

$$\text{Kernel} \longrightarrow \text{udev} \longrightarrow \text{NetworkManager} \rightleftharpoons \text{D-Bus} \rightleftharpoons \text{Firefox}$$

La détection automatique du matériel

Fonctionnement de udev

La syntaxe des fichiers de règles est assez simple : chaque ligne contient des critères de sélection et des directives d'affectation. Les premiers permettent de sélectionner les événements sur lesquels il faudra réagir et les seconds définissent l'action à effectuer.

Exemple

```
KERNEL=="sd*[0-9]|dasd*[0-9]", ENV{ID_SERIAL}=="?*", \
        SYMLINK+="disk/by-id/$env{ID_BUS}-$env{ID_SERIAL}-part%n"
```

La détection automatique du matériel

Fonctionnement de udev

Les règles sont simplement séparés par des virgules et c'est l'opérateur qui désigne s'il s'agit d'un critère de sélection (pour les opérateurs de comparaison==ou!=) ou d'une directive d'affectation (pour les opérateurs=,+=ou:=).

La détection automatique du matériel

Fonctionnement de udev

Les opérateurs de comparaison s'emploient sur les variables suivantes :

- **KERNEL** : le nom que le noyau affecte au périphérique ;
- **ACTION** : l'action correspondant à l'événement (« add » pour l'ajout d'un périphérique, « remove » pour la suppression) ;
- **DEVPATH** : le chemin de l'entrée correspondant au périphérique dans /sys/ ;
- **SUBSYSTEM** : le sous-système du noyau à l'origine de la demande (ils sont nombreux mais citons par exemple « usb », « ide », « net », « firmware », etc.) ;

La détection automatique du matériel

Fonctionnement de udev

Les opérateurs de comparaison s'emploient sur les variables suivantes :

- **ATTR{attribut}** : contenu du fichier attribut dans le répertoire /sys/$devpath/ du périphérique. C'est ici que l'on va trouver les adresses MAC et autres identifiants spécifiques à chaque bus ;
- **PROGRAM** : délègue le test au programme indiqué(vrai s'il renvoie 0, faux sinon). Le contenu de la sortie standard du programme est stocké afin de pouvoir l'utiliser dans le cadre du test RESULT ;
- **RESULT** : effectue des tests sur la sortie standard du dernier appel à PROGRAM.

La détection automatique du matériel

Fonctionnement de udev

Voici les variables qui peuvent être modifiées :

- **NAME** : le nom du fichier de périphérique à créer dans /dev/.
  SYMLINK : la liste des noms symboliques qui pointeront sur le même périphérique ;
- **OWNER**,**GROUP** et **MODE** définissent l'utilisateur et le groupe propriétaire du périphérique ainsi que les permissions associées ;
- **RUN** : la liste des programmes à exécuter en réponse à cet événement.

La détection automatique du matériel

Exemple de règle udev

ici un exemple de règle qui crée un lien symbolique **/dev/video-cam** lorsqu'une webcam est connectée.

Supposons que cette caméra soit actuellement connectée et qu'elle ait été chargée avec le nom de l'appareil **/dev/video1**. La raison de l'écriture de cette règle est qu'au prochain démarrage, le périphérique pourrait apparaître sous un nom différent, comme **/dev/video0**.

La détection automatique du matériel

Exemple de règle udev

udevadm info --attribute-walk --path=$(udevadm info --query=path --name=/dev/video1)

Les informations collecté par **udevadm** commencent par le périphérique spécifié par le chemin dev, puis remontent la chaîne des périphériques parents.

Il imprime pour chaque périphérique trouvé, tous les attributs possibles dans le format de clé udev rules.

Une règle à associer peut être composée des attributs de l'appareil et des attributs d'un seul appareil parent.

La détection automatique du matériel

```
  looking at device
'/devices/pci0000:00/0000:00:14.0/usb1/1-6/1-6:1.0/video4linux/video1':
    KERNEL=="video1"
    SUBSYSTEM=="video4linux"
    DRIVER==""
    ATTR{dev_debug}=="0"
    ATTR{index}=="1"
    ATTR{name}=="TOSHIBA Web Camera - FHD: TOSHI"
    ATTR{power/control}=="auto"
    ATTR{power/runtime_active_time}=="0"
    ATTR{power/runtime_status}=="unsupported"
    ATTR{power/runtime_suspended_time}=="0"
```

La détection automatique du matériel

```
looking at parent device
'/devices/pci0000:00/0000:00:14.0/usb1/1-6/1-6:1.0':
    KERNELS=="1-6:1.0"
    SUBSYSTEMS=="usb"
    DRIVERS=="uvcvideo"
    [...]

looking at parent device '/devices/pci0000:00/0000:00:14.0/usb1/1-6':
    [...]
    ATTRS{idProduct}=="b537"
    ATTRS{idVendor}=="04f2"
    [...]
```

La détection automatique du matériel

Exemple de règle udev

Pour identifier la webcamera, à partir du périphérique video4linux que nous utilisons **KERNEL=="video1"**et **SUBSYSTEM=="video4linux"**, en remontant ensuite de deux niveaux au-dessus, nous faisons correspondre la webcamera en utilisant les ID de fournisseur et de produit du parent USB

En remontant ensuite de deux niveaux au-dessus, nous faisons correspondre la webcamera en utilisant les ID de fournisseur et de produit du parent **USB SUBSYSTEMS=="usb"**, **ATTRS{idVendor}=="04f2"**et **ATTRS{idProduct}=="b537"**

La détection automatique du matériel

Exemple de règle udev

Nous sommes maintenant en mesure de créer une correspondance de règle pour cet appareil comme suit:

/etc/udev/rules.d/83-webcam.rules
KERNEL=="video[0-9]*", SUBSYSTEM=="video4linux", SUBSYSTEMS=="usb", ATTRS{idVendor}=="04f2", ATTRS{idProduct}=="b537", SYMLINK+="video-cam"

/etc/udev/rules.d/83-webcam-removed.rules
ACTION=="remove", SUBSYSTEM=="usb", ENV{ID_VENDOR_ID}=="04f2", ENV{ID_MODEL_ID}=="b537", RUN+="/usr/script/webcam.sh"

Création d'un noyau personnalisé

Création d'un noyau personnalisé.

Les néophytes du système Linux se demandent souvent : "Pourquoi devrais-je construire mon propre noyau ?".

Étant donné les progrès réalisés en matière d'utilisation des modules de noyau, la meilleure réponse à cette question est sans doute : "Si vous ne savez pas pour quelle raison vous devriez construire votre propre noyau, vous n'avez probablement pas besoin de le faire."

Création d'un noyau personnalisé

Pourquoi recompiler le noyau ?

Le noyau qui tourne en ce moment sur votre Linux fraîchement installé est le noyau fourni en standard dans la distribution de votre choix. C'est un noyau assez gros qui est destiné à pouvoir fonctionner sur la quasi-totalité des ordinateurs.

Ce que nous pouvons faire c’est personnaliser le noyau pour qu'il supporte nos périphériques et eux seulement.
Plus le noyau est petit, plus le système d'exploitation est rapide. L'étape la plus difficile sera la configuration du noyau, pour qu'il supporte bien tous nos périphériques.

Création d'un noyau personnalisé

Comment faire ?

Les différentes distribution Linux viennent avec des outils d’aides à la récupération et l’installation du noyau. Nous allons voir ici une méthode générique qui s’appliquera à n’importe quel distribution.

Juste avant voyons quelques mots de vocabulaire.

Création d'un noyau personnalisé

Vocabulaire

Les sources sont le code du noyau (ou de tout autre logiciel) avant sa compilation sous forme binaire destiné à être exécuté. Le noyau Linux est écrit en différents langages de programmation (majoritairement “C”), et les deux programmes chargés de compiler le code source sous Linux sont généralement “**make**” et “**gcc**”

Compiler est l'action de transformer le code source “brute” d'un programme en une image binaire adaptée et personnalisée qui sera exécutée au sein de votre système GNU-Linux.
Un fichier souvent nommé “makefile” agit comme une super recette, il est utilisé par le programme “make” qui va réunir et lier les différents éléments nécessaires à la réalisation finale.
Pour y parvenir “make” va faire appel à un compilateur qui sur un système GNU-Linux est généralement “gcc”.

Création d'un noyau personnalisé

Vocabulaire

Module est parfois utilisé de manière interchangeable avec “pilote” (“driver” en anglais). Un module peut être un élément qui interface un matériel, le noyau, et les applications (c'est le cas des modules regroupés sous “kernel/drivers/” dans le répertoire des modules), mais de nombreux modules ne sont pas directement des “pilotes” de matériel et ce sens est extrêmement réducteur de la diversité des modules.

initrd ou **initramfs** est un système de fichier initial temporaire, chargé en mémoire vive par le chargeur d'amorçage (par exemple grub) à partir duquel le système de fichier racine définitif pourra être monté. Le rôle de l'initrd ne se borne pas à monter le système de fichier racine et à passer la main au noyau, il peut également inclure les modules nécessaires à initialiser les cartes réseau, initialiser les couches d'abstraction nécessaires pour monter la partition racine (md/raid, lvm) etc...

Création d'un noyau personnalisé

Installer un environnement de compilation

Compiler un noyau nécessite comme toute compilation de disposer d'un certain nombre d'outils, en premier lieu un compilateur (**gcc**), le programme **make**, éventuellement le programme **patch** si vous désirez appliquer des patchs

Des dépendances supplémentaires peuvent être nécessaires pour utiliser des interfaces de configuration conviviales “menuconfig” ou mieux “nconfig” basées sur “ncurses” (paquets de développement de **libncurses**)

Création d'un noyau personnalisé

Récupérer les sources

Pour vérifier le kernel en cours sur votre machine :
uname -sr

Pour récupérer les sources du noyaux :

- **https://www.kernel.org/**
- **https://mirrors.edge.kernel.org/pub/linux/kernel/**

Création d'un noyau personnalisé

Copiez les sources du noyau dans un dossier que vous réserverez à la compilation des noyaux, par exemple :

```
cp linux-X.XX.tar.xz ~/kernelbuild/
```

Décompressez l'archive et entrez dans le dossier qui a été créé :

```
cd ~/kernelbuild|
tar -xvJf linux-X.XX.tar.xz
cd linux-X.XX
```

Création d'un noyau personnalisé

Préparez la compilation en lançant la commande de nettoyage suivante :

```
$ make mrproper
```

Ceci garantit que l'arbre du noyau est parfaitement propre. L'équipe du noyau Linux recommande de lancer cette commande avant toute compilation. Ne présupposez pas que l'arbre des sources du noyau est propre après la décompression.

Création d'un noyau personnalisé

Configuration de la compilation

Cette étape est la plus cruciale dans la personnalisation du noyau. Elle lui permettra de s'adapter au mieux aux spécificités de votre matériel.

En définissant correctement les options dans le fichier .config, votre noyau et votre ordinateur fonctionneront plus efficacement.

Astuce : Il est possible de configurer un noyau qui se passe d'initramfs dans des configurations simples. Assurez-vous que les modules requis pour la vidéo, les disques, les entrées et les systèmes de fichiers sont compilés dans le noyau, ainsi que, au grand minimum, la prise en charge de DEVTMPFS_MOUNT, TMPFS, AUTOFS4_FS.

Création d'un noyau personnalisé

Configuration de la compilation

The easy way

Récupérer le fichier de configuration (.config) du noyau sur lequel le système tourne

```
zcat /proc/config.gz > .config
```

Il vous faut ensuite brancher tous les périphériques que vous pensez utiliser sur le système puis rendez-vous dans votre dossier source et exécutez :

```
make localmodconfig
```

Ceci sélectionne seulement les options utilisées couramment.
La configuration résultante sera écrite dans **.config.**

Création d'un noyau personnalisé

Configuration de la compilation

La commande traditionnelle menuconfig

Une interface ncurses de configuration du noyau est disponible avec la commande
make menuconfig

En alternative, vous pouvez utiliser la commande plus moderne
make nconfig

Création d'un noyau personnalisé

Compilation

Le temps de compilation peut aller de 15 minutes à plusieurs heures. Ceci dépend du nombre d'options/modules sélectionnés et des capacités du processeur

Exécutez la commande :
make

Installation des modules
make modules_install

Copie du noyau dans le dossier /boot
cp -v arch/x86/boot/bzImage /boot/vmlinuz-leNomDeVotreNoyau

Création d'un noyau personnalisé

Construire un disque virtuel de démarrage — initramfs

mkinitcpio -k NomCompletDuNoyau -c /etc/mkinitcpio.conf -g /boot/initramfs-NomCompletDuNoyau.img

Vous êtes libres de nommer les fichiers de /boot comme vous le voulez. Néanmoins, utiliser le nommage **[kernel-major-minor-revision]** aide à garder les choses claires si vous:

- conservez plusieurs noyaux
- utilisez souvent mkinitcpio
- compilez des modules de tierces parties

Création d'un noyau personnalisé

mkinitcpio est un script shell utilisé pour créer un environnement qui se chargera en premier en mémoire :

- Il utilise BusyBox pour fournir un ensemble d'outils de base dans le ramdisk.
- Il peut utiliser udev pour la détection à chaud du matériel, ça évite ainsi le chargement d'une liste de modules par défaut.
- Il possède un système d'extensions (hooks) permettant d'effectuer des actions supplémentaires pendant le démarrage du noyau.
- Il supporte lvm2, dm-crypt pour par exemple, les volumes luks, raid, ainsi que le boot depuis des volumes usb.
- Plusieurs caractéristiques peuvent directement être configurées depuis la ligne du noyau sans avoir à re-générer l'image.
- mkinitcpio peut inclure l'image dans le noyau afin d'avoir un fichier unique contenant noyau et image.

Création d'une distribution Linux personnalisée

La création d’une distribution Linux implique la création d’une politique de maintenance et d’un plan sur le long terme avec une vision de ce que propose votre distribution. C’est une lourde charge de travail et ne peut être réalisé par un seul individu.

C’est d’ailleurs pour cela que beaucoup de distribution découle de distribution majeur comme Fedoras , Centos , Debian etc ..

Parfois certaines distribution font le choix de partir de 0 pour avoir les mains libre sur les choix techniques à long terme. On peut citer Archlinux par exemple.

Création d'une distribution Linux personnalisée

Il faut donc faire la différence entre créer une distribution depuis 0 et partir depuis une base.

Dans le cas d’une base existante, le plus évident reste d’utiliser les outils de personnalisation fourni par les mainteneurs ou la communauté de la distribution.

Chez Debian ont trouve Linux respin
Pour ubuntu (Un fork de Debian) ont trouve Ubuntu imager

Toutefois , pour ceux qui savent ce qu’ils font, ils toujours plus intéressant de passer par un fichier Kickstart ou Preseed.

Il reste toutefois une alternative qui est de réaliser un LFS

Création d'une distribution Linux personnalisée

LFS , Linux From Scratch

Qu'est-ce que Linux From Scratch?

Linux From Scratch (LFS) est un projet qui vous fournit des instructions étape par étape pour créer votre propre système Linux personnalisé entièrement à partir de la source.

Que puis-je faire avec mon système LFS?

Un système LFS à la main est assez minime, mais est conçu pour fournir une base solide sur laquelle vous pouvez ajouter les packages que vous souhaitez. Voir le projet BLFS pour une sélection de packages couramment utilisés.

Création d'une distribution Linux personnalisée

Suivre un LFS reste fastidieux passer la période de découverte et d'apprentissage. C'est pourquoi il existe **ALSF** pour **Automate Linux From Scratch**

Que puis-je faire avec ALFS ?

L'objectif d'ALFS est d' automatiser le processus de création d'un système LFS. Il cherche à suivre le livre le plus fidèlement possible en extrayant directement les instructions des sources XML.
C'est la raison pour laquelle il peut également être utilisé comme test des instructions actuelles du livre.

Création d'une distribution Linux personnalisée

NuTyX une distro LFS ?

Si vous souhaitez un exemple d’une distribution qui suit les concept du LFS et du BLFS vous pouvez découvrir cette distribution française.

NuTyX site web

Identifier le driver nécessaire à un composant

Les bon outils

Trouver le bon drivers pour son périphérique signifie aller récupérer des informations sur le périphérique en question.

Pour cela commençons par l'installation deux trois paquets qui nous seront bien utiles :

- pciutils
- dmidecode
- usbutils
- Smartmontools

Identifier le driver nécessaire à un composant

La carte mère

Pour avoir toutes les informations utiles sur sa carte mère et quelques informations diverses sur votre matériel.

dmidecode

```
Handle 0x0017, DMI type 2, 17 bytes
Base Board Information
        Manufacturer: TOSHIBA
        Product Name: PORTEGE Z30-C
        Version: Version A0
        Serial Number: C0414952C0H4700E
        [...]
```

Identifier le driver nécessaire à un composant

CPU

Pour connaître la version et afficher toutes les informations de son CPU ou processeur faites :

```
uname -p
```

Pour obtenir les infos complète vous pouvez ensuite employé les commandes suivante :

```
cat /proc/cpuinfo
lscpu
```

Identifier le driver nécessaire à un composant

La carte graphique

Pour connaître le modèle et le nom de sa carte graphique :

```
lspci | grep VGA
00:02.0 VGA compatible controller: Intel Corporation Skylake GT2 [HD Graphics 520] (rev 07)
```

Une fois les drivers Ati ou Nvidia installés, on peut vérifier la présence de l'accélération graphique par :

```
glxinfo | grep rendering
direct rendering: Yes
```

Identifier le driver nécessaire à un composant

Composant PCI

La commande lspci vous donnera la liste des périphérique PCI connecté à la carte. Pour plus de détail (Comme le nom du module kernel utilisé actuellement) il faudra utiliser l'option verbose

lspci -v

00:00.0 Host bridge: Intel Corporation Xeon E3-1200 v5/E3-1500 v5/6th Gen Core Processor Host Bridge/DRAM Registers (rev 08)
	Subsystem: Toshiba Corporation Device 0001
	Flags: bus master, fast devsel, latency 0
	Capabilities: <access denied>
	Kernel driver in use: skl_uncore

Identifier le driver nécessaire à un composant

/sys

Les commandes que nous avons vus avant ne vont finalement que vous aidez à parser les informations tiré de /sys

```
find /sys/devices/ -name net -print | xargs ls
'/sys/devices/pci0000:00/0000:00:14.0/usb1/1-2/1-2:2.12/net':
wwan0

'/sys/devices/pci0000:00/0000:00:1c.2/0000:02:00.0/net':
wlan0

'/sys/devices/pci0000:00/0000:00:1f.6/net':
enp0s31f6
```

Installation de drivers "exotiques"

L'installation de driver sous linux peut être réalisé selon deux méthodes.

Dans la première le driver se trouve dans un paquet issue de la distribution. Il suffit donc de récupérer ce paquet , l'installer et voilà.

Dans la seconde, celui-ci n'a pas était fourni dans un paquet , il faudra donc se rendre sur le site du constructeur pour retrouver les sources du drivers et sélectionner celui qui correspond à notre matériel ET notre architecture CPU.

La suite se trouvera dans le README fourni avec (normalement) et consistera dans l'enchainement des commandes :
make && make install (parfois il faudra un .configure avant pour la création du MakeFile)

Ajout d'un pilote spécifique dans initrd (mkinitrd).

Le noyau Linux est de nature modulaire. L'image du disque RAM / noyau utilisée pour démarrer Linux contient les modules initiaux qui sont utilisés pour démarrer le système; après le démarrage initial, d'autres modules sont chargés selon les besoins.

Dans certains cas, les modules du noyau doivent être inclus dans le disque RAM.

Les exemples incluent les cartes Fibre Channel, les contrôleurs de disque et les pilotes de système de fichiers.

Ajout d'un pilote spécifique dans initrd (mkinitrd).

En fonction des distributions, il est possible de configurer ce qui sera ajouter à l'initrd.

Toutefois il existe une méthode en ligne de commande permettant l'ajout d'un module.

```
mkinitrd -h
usage: mkinitrd [--version] [--help] [-v] [-f] [--preload <module>]
       [--image-version] [--with=<module>]
       [--nocompress]
       <initrd-image> <kernel-version>
```

Ajout d'un pilote spécifique dans initrd (mkinitrd).

Options importante

–f Permet à mkinitrd d'écraser un fichier image existant.
–Preload=module - Charge le module dans l'image initiale du disque virtuel. Le module est chargé avant tout module SCSI spécifié dans /etc/modprobe.conf. Cette option peut être utilisée autant de fois que nécessaire.
–With=module - Charge le module dans l'image initiale du disque virtuel. Le module est chargé après tous les modules SCSI spécifiés dans /etc/modprobe.conf. Cette option peut être utilisée autant de fois que nécessaire.

Les paramètres du noyau

Paramètres d'amorçage

Les paramètres d'amorçage sont des paramètres passés au noyau Linux pour s'assurer que les périphériques seront correctement pris en compte. Dans la plupart des cas le noyau détecte les périphériques, mais parfois vous devez l'aider un peu.

Si c'est la première fois que vous démarrez le système, essayez les paramètres par défaut ; autrement dit, ne donnez pas de paramètre et vérifiez que cela fonctionne correctement. Ce devrait être le cas. Sinon, vous pouvez redémarrer et donner les paramètres nécessaires à votre matériel.

Les paramètres du noyau

Paramètres d'amorçage

La liste des paramètre d'amorçage est relativement longue , nous allons donc nous référer à la bible dans le domaine :

https://tldp.org/HOWTO/BootPrompt-HOWTO.html

Les paramètres du noyau

sysctl

sysctl est un outil pour examiner et modifier les paramètres du noyau dynamiquement
sysctl est implémenté dans procfs , le système de fichiers de processus virtuel à /proc/.

Les paramètres disponibles sont ceux listés sous /proc/sys/.
Par exemple, le paramètre **kernel.sysrq** fait référence au fichier /proc/sys/kernel/sysrq sur le système de fichiers.

La commande sysctl --all peut être utilisée pour afficher toutes les valeurs actuellement disponibles.

Les paramètres du noyau

sysctl

Le fichier de préchargement/configuration **sysctl** peut être créé à l' adresse /etc/sysctl.d/99-sysctl.conf (Pour systemd , /etc/sysctl.d/et /usr/lib/sysctl.d/)

La dénomination et le répertoire source décident de l'ordre de traitement, ce qui est important car le dernier paramètre traité peut remplacer les précédents.

Par exemple, les paramètres de /usr/lib/sysctl.d/50-default.conf seront remplacés par des paramètres égaux dans /etc/sysctl.d/50-default.conf

Les paramètres du noyau

sysctl

Les paramètres peuvent être modifiés par la manipulation de fichiers ou à l'aide de l'utilitaire.

Par exemple, pour activer temporairement la clé magique SysRq :

```
sysctl kernel.sysrq = 1
```

Mais la modification directement dans le fichier concerné produira le même effet :

```
echo "1" > /proc/sys/kernel/sysrq
```

05
Maintenance et métrologie sur des serveurs Linux

Collecte, centralisation et analyse des logs système

Pourquoi journaliser?

Un système Linux a de nombreux sous-systèmes et applications en cours d'exécution. Nous utilisons la journalisation du système pour collecter des données sur notre système en cours d'exécution à partir du moment où il démarre.

Parfois, nous avons juste besoin de savoir que tout va bien. À d'autres moments, nous utilisons ces données pour auditer, déboguer, savoir quand un disque ou une autre ressource est à court de capacité, et à bien d'autres fins.

Collecte, centralisation et analyse des logs système

syslog

Syslog est une norme pour l'envoi et la réception de messages de notification - dans un format particulier - à partir de divers périphériques réseau.

Les messages incluent les horodatages, les messages d'événement, la gravité, les adresses IP de l'hôte, les diagnostics, etc.

Collecte, centralisation et analyse des logs système

syslogd

Le démon syslog est un processus serveur qui fournit une fonction de journalisation des messages pour les processus d'application et système..

La fonction traditionnelle syslog et son démon syslogd ont été complétées par d'autres fonctions de journalisation telles que rsyslog, syslog-ng et le sous-système de journal systemd.

Collecte, centralisation et analyse des logs système

syslog.conf

Le fichier syslog.conf est le fichier de configuration principal de syslogd. Chaque fois que syslogd reçoit un message de journal, il agit en fonction du type (ou de la fonction) du message et de sa priorité (appelés champs de sélection).

type (facility).priority (severity) destination(where to send the log)

Collecte, centralisation et analyse des logs système

facility

Les facility sont simplement des catégories.

- **auth**: messages de sécurité / d'authentification
- **utilisateur**: messages au niveau de l'utilisateur
- **kern**: messages du noyau
- **maïs**: démon d'horloge
- **démon**: démons système
- **mail**: système de messagerie
- **local0 – local7**: facility utilisées localement

Collecte, centralisation et analyse des logs système

Action

Sur le champ d'action, nous pouvons avoir des choses comme :

/var/log/messages	Ecris les logs dans le fichier spécifié.
user2	Envoyer une notification dans le terminal de la personne.
@192.168.10.42	Envoie les journaux au serveur de journaux spécifié et ce serveur décide comment traiter les journaux en fonction de ses configurations.

Collecte, centralisation et analyse des logs système

Surveillance des logs avec logcheck

Le programme logcheck scrute par défaut les fichiers de logs toutes les heures et envoie par courrier électronique à **root** les messages les plus inhabituels pour aider à détecter tout nouveau problème.

La liste des fichiers scrutés se trouve dans le fichier **/etc/logcheck/logcheck.logfiles** ;

les choix par défaut conviendront si le fichier **/etc/rsyslog.conf** n'a pas été complètement remodelé.

Collecte, centralisation et analyse des logs système

Surveillance des logs avec logcheck

logcheck peut fonctionner en 3 modes plus ou moins détaillés :

- paranoid (paranoïaque)
- server (serveur)
- workstation (station de travail)

Le premier étant le plus verbeux, on le réservera aux serveurs spécialisés (comme les pare-feu). Le deuxième mode, choisi par défaut, est recommandé pour les serveurs.

Le dernier, prévu pour les stations de travail, élimine encore plus de messages.

Collecte, centralisation et analyse des logs système

Surveillance des logs avec logcheck

Dans tous les cas, il faudra probablement paramétrer logcheck pour exclure des messages supplémentaires (selon les services installés) sous peine d'être envahi chaque heure par une multitude de messages inintéressants.

Leur mécanisme de sélection étant relativement complexe, il faut lire à tête reposée le document /usr/share/doc/logcheck-database/README.logcheck-database.gz pour bien le comprendre.

Collecte, centralisation et analyse des logs système

Surveillance des logs avec logcheck

Plusieurs types de règles sont appliqués :

- celles qui qualifient un message comme résultant d'une tentative d'attaque (elles sont stockées dans un fichier du répertoire /etc/logcheck/cracking.d/)
- celles qui annulent cette qualification (/etc/logcheck/cracking.ignore.d/) ;
- celles qui qualifient un message comme une alerte de sécurité (/etc/logcheck/violations.d/) ;
- celles qui annulent cette qualification (/etc/logcheck/violations.ignore.d/) ;
- et enfin celles qui s'appliquent à tous les messages restants (les System Events, ou événements système).

Collecte, centralisation et analyse des logs système

Vérification de l'intégrité du système avec AIDE

AIDE (Advanced Intrusion Detection Environment) est un vérificateur d'intégrité des fichiers et des répertoires.

Il crée une base de données à partir des règles d'expressions régulières qu'il trouve dans le ou les fichiers de configuration. Une fois cette base de données initialisée, elle peut être utilisée pour vérifier l'intégrité des fichiers. Il a plusieurs algorithmes de résumé de message qui peuvent être utilisés pour vérifier l'intégrité du fichier. Tous les attributs de fichier habituels peuvent également être vérifiés pour les incohérences.

Collecte, centralisation et analyse des logs système

Vérification de l'intégrité du système avec AIDE

AIDE prend un « instantané » de l'état du système, enregistre les fragmentations, les moments liés à des modifications et toute autre donnée concernant les fichiers définis par l'administrateur.

Lorsque l'administrateur souhaite exécuter un test d'intégrité, l'administrateur place la base de données précédemment générée en un lieu accessible et commande AIDE afin de comparer la base de données avec l'état réel du système. Toute modification qui se serait produite sur l'ordinateur entre la création de l'instantané et le test sera détectée par AIDE et sera signalée à l'administrateur.

Collecte, centralisation et analyse des logs système

Vérification de l'intégrité du système avec AIDE

Il existe plusieurs système similaire permettant de réaliser un hash des fichiers présent sur le système. Toutefois AIDE va plus loin grâce à certaines fonctionnalité comme :

- Attributs de fichier pris en charge: type de fichier, autorisations, inode, Uid, Gid, nom du lien, taille, nombre de blocs, nombre de liens, Mtime, Ctime et Atime
- Support pour les ACL, SELinux, XAttrs et les attributs des système de fichier étendu.

Collecte, centralisation et analyse des logs système

Suivi de l'activité des processus et du système (lsof, vmstat, sysstat)

Quand un système Unix manque de mémoire, cela a des conséquences souvent catastrophiques : l'ensemble du système se met à fonctionner au ralenti jusqu'à la paralysie complète de la machine !

Le manque de mémoire est donc un problème majeur à traiter sans délai. Il faut donc avant tout être capable de repérer ce phénomène.

Collecte, centralisation et analyse des logs système

Suivi de l'activité des processus et du système (lsof, vmstat, sysstat)

La commande free

Pour mesurer la consommation de la mémoire, il existe plusieurs commandes, chacune avec sa spécificité. La commande free fournit des informations détaillées sur la façon dont la mémoire est consommée :

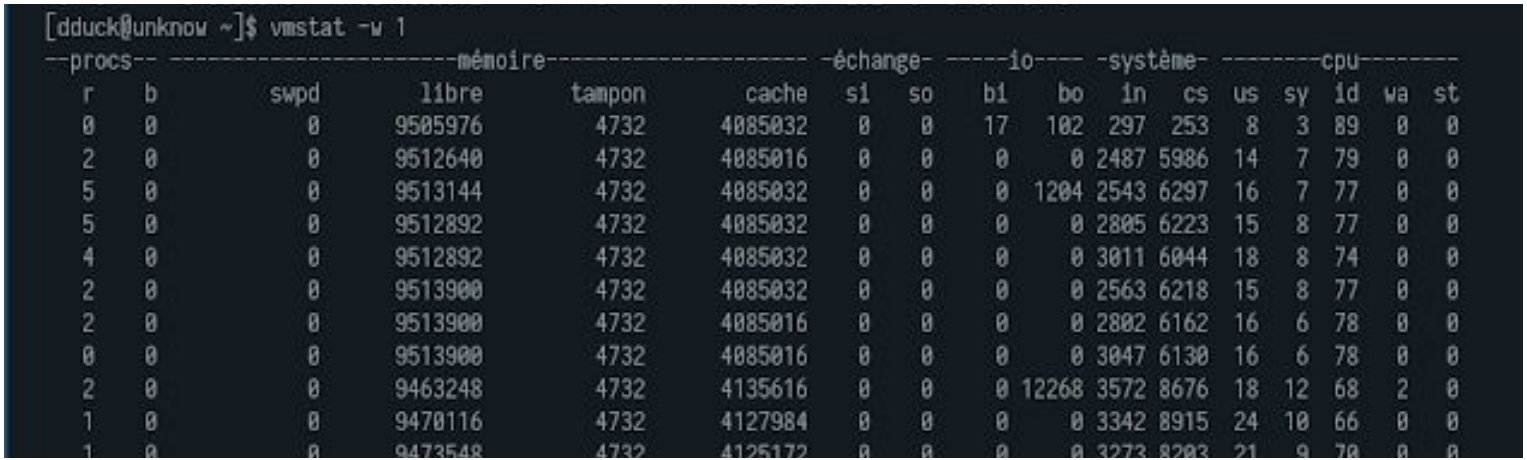
```
[dduck@unknow ~]$ free -m
               total        utilisé      libre     partagé tamp/cache   disponible
Mem:           15939         2748         9370         735         3820        12162
Partition d'échange:      20479            0        20479
[dduck@unknow ~]$ free -mh
               total        utilisé      libre     partagé tamp/cache   disponible
Mem:            15Gi        2,7Gi       9,0Gi       812Mi       3,9Gi        11Gi
Partition d'échange:       19Gi          0B        19Gi
[dduck@unknow ~]$
```

Collecte, centralisation et analyse des logs système

Suivi de l'activité des processus et du système (lsof, vmstat, sysstat)

La commande vmstat

La commande vmstat permet de suivre en temps réel l'utilisation de la mémoire :



```
[dduck@unknow ~]$ vmstat -w 1
--procs-- -----------------------mémoire---------------------- -échange- -----io---- -système- --------cpu--------
   r    b         swpd         libre       tampon        cache   si   so    bi    bo   in   cs us sy id wa st
   0    0            0       9505976         4732      4085032    0    0    17   102  297  253  8  3 89  0  0
   2    0            0       9512640         4732      4085016    0    0     0     0 2487 5986 14  7 79  0  0
   5    0            0       9513144         4732      4085032    0    0     0  1204 2543 6297 16  7 77  0  0
   5    0            0       9512892         4732      4085032    0    0     0     0 2805 6223 15  8 77  0  0
   4    0            0       9512892         4732      4085032    0    0     0     0 3011 6044 18  8 74  0  0
   2    0            0       9513900         4732      4085032    0    0     0     0 2563 6218 15  8 77  0  0
   2    0            0       9513900         4732      4085016    0    0     0     0 2802 6162 16  6 78  0  0
   0    0            0       9513900         4732      4085016    0    0     0     0 3047 6130 16  6 78  0  0
   2    0            0       9463248         4732      4135616    0    0     0 12268 3572 8676 18 12 68  2  0
   1    0            0       9470116         4732      4127984    0    0     0     0 3342 8915 24 10 66  0  0
   1    0            0       9473548         4732      4125172    0    0     0     0 3273 8203 21  9 70  0  0
```

Collecte, centralisation et analyse des logs système

Suivi de l'activité des processus et du système (lsof, vmstat, sysstat)

Le paquet Sysstat

Sysstat est un package qui contient le binaire **sar** et **iostat**. Le dernier sert à monitorer uniquement les IO disque, tandis que sar sert à monitorer à peu prêt tout.

06
Blocage, crash et dépannage d'urgence

Fonctionnement détaillé du boot

GRUB (acronyme signifiant en anglais « GRand Unified Bootloader ») est un programme d'amorçage de micro-ordinateur.

Il s'exécute à la mise sous tension de l'ordinateur, après les séquences de contrôle interne et avant le système d'exploitation proprement dit, puisque son rôle est justement d'en organiser le chargement.

Lorsque l'ordinateur héberge plusieurs systèmes (on parle alors de multi-amorçage), il permet à l'utilisateur de choisir quel système démarrer.

Fonctionnement détaillé du boot

Fonctionnement

Quand l'ordinateur est allumé, le BIOS cherche le premier périphérique bootable (habituellement le disque dur), charge le secteur d'amorçage ou Master Boot Record (MBR), correspondant aux 512 premiers octets de ce disque, puis transfère le contrôle à ce code.

Fonctionnement détaillé du boot

Version 0 (Grub legacy)

GRUB 0 suit une approche en deux étapes. Le Master Boot Record (MBR) contient généralement l' **étape 1** de GRUB , ou peut contenir une implémentation MBR standard qui est en charge de lancer l' étape 1 de GRUB à partir du secteur de démarrage de la partition active .

Compte tenu de la petite taille d'un secteur de démarrage (512 octets), l' étape 1 peut faire un peu plus que charger l'étape suivante de GRUB en chargeant quelques secteurs de disque à partir d'un emplacement fixe près du début du disque (dans ses 1024 premiers cylindres).

Fonctionnement détaillé du boot

L'**étape 1** peut charger l' **étape 2** directement, mais il est normalement configuré pour charger l'**étape 1.5**. , situé dans les 30 premiers Kio du disque dur immédiatement après le MBR et avant la première partition.

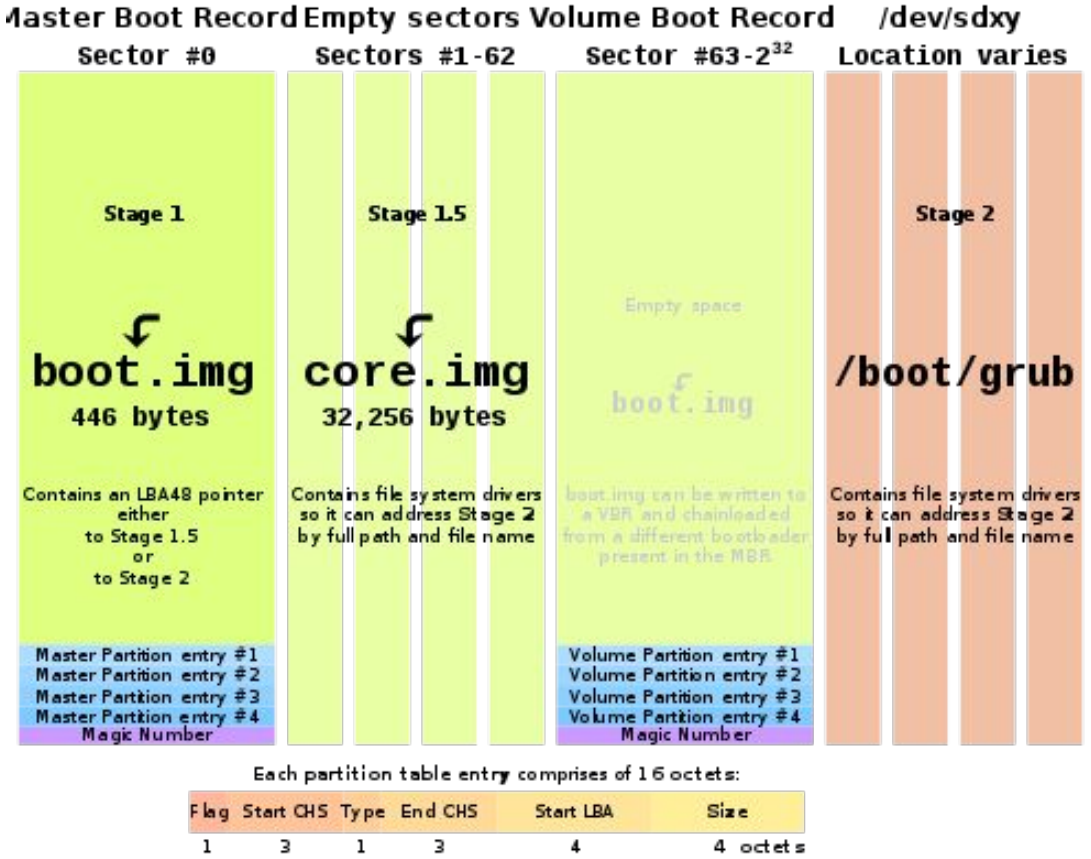
L' image de l'**étape 1.5** contient des pilotes de système de fichiers, lui permettant de charger directement l'**étage 2** à partir de n'importe quel emplacement connu du système de fichiers, par exemple à partir de **/boot/grub**.

L'**étape 2** chargera ensuite le fichier de configuration par défaut et tous les autres modules nécessaires.

Fonctionnement détaillé du boot

GRUB version 2

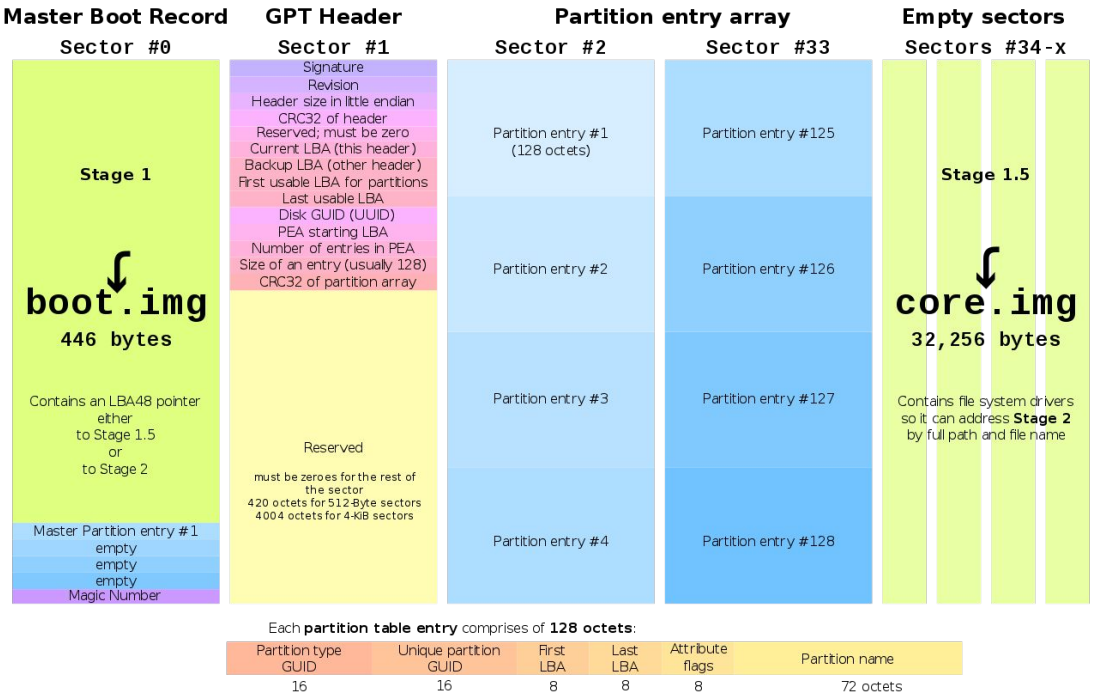
Démarrage sur les systèmes à l' aide du BIOS firmware



Fonctionnement détaillé du boot

GRUB version 2

Démarrage sur les systèmes à l' aide du BIOS firmware

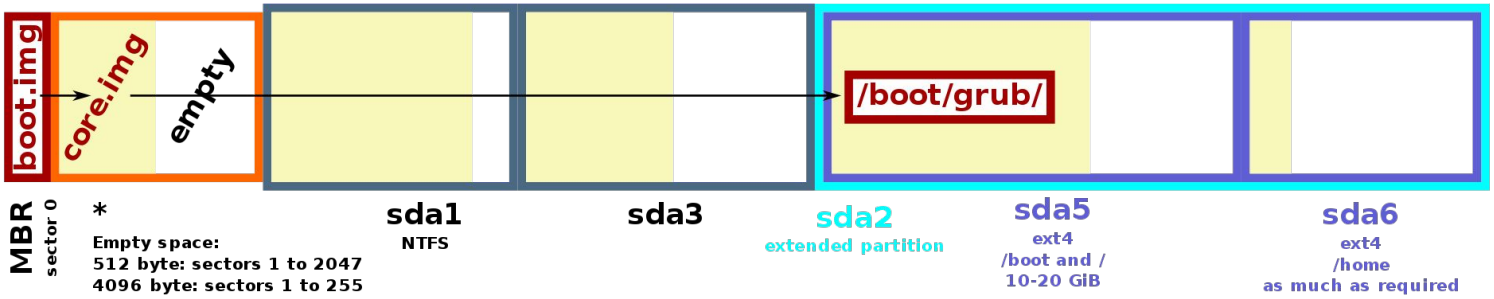


Fonctionnement détaillé du boot

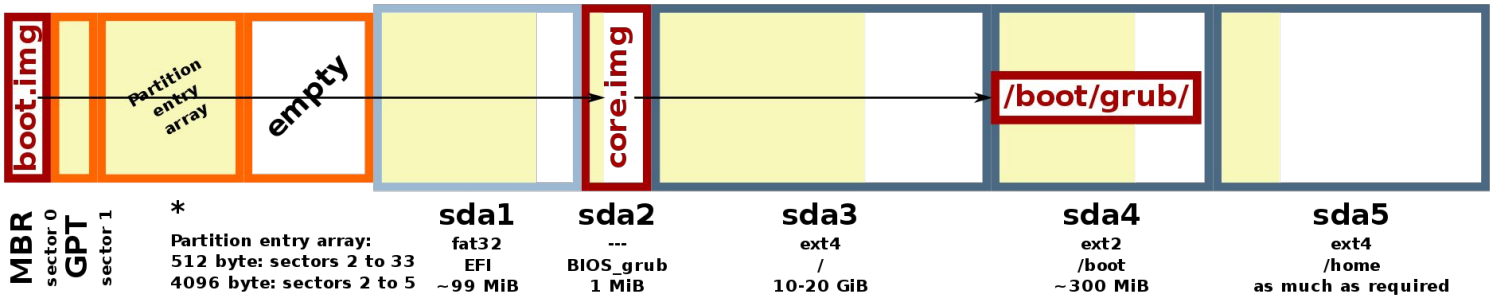
GNU GRUB 2

Locations of *boot.img*, *core.img* and the */boot/grub* directory

Example 1: An MBR-partitioned hard disk with sector size of 512 or 4096 bytes



Example 2: A GPT-partitioned hard disk with sector size of 512 or 4096 bytes



Fonctionnement détaillé du boot

Boot.img (étape 1) est écrit dans les 440 premiers octets du Master Boot Record (code de démarrage MBR dans le secteur 0), ou éventuellement dans un secteur de démarrage de partition (PBR).

Il charge **diskboot.img** via une adresse LBA de 64 bits.

Le numéro de secteur réel est écrit par grub-install. **diskboot.img** est le premier secteur de core.img dans le seul but de charger le reste des core.img via les numéros de secteur identifiés via les secteurs de l'addresse LBA également écrits par grub-install.

Fonctionnement détaillé du boot

Sur les disques partitionnés en MBR, **core.img** (étape 1.5) est stocké dans les secteurs vides (si disponibles) entre le MBR et la première partition.

Les systèmes d'exploitation récents suggèrent ici un écart de 1 Mio pour l'alignement (secteurs de 2047 * 512 octets ou 255 * 4KiB).

Cet écart était de 62 secteurs (31 KiB) pour rappeler la limite du nombre de secteurs de l' adressage Cylinder-Head-Sector (C / H / S) utilisé par le BIOS avant 1996, **core.img** est donc conçu pour être inférieur à 32 KiB

Fonctionnement détaillé du boot

Sur les disques partitionnés GPT, les partitions primaires ne sont pas limitées à 4, **core.img** est donc écrite sur sa propre partition de démarrage BIOS minuscule (1 Mio), sans système de fichiers.

Fonctionnement détaillé du boot

étape 2: core.img charge /boot/grub/<arch>/normal.mod depuis la partition configurée par grub-install.

Si l'index de partition a changé, GRUB ne pourra pas trouver le normal.mod, et présente à l'utilisateur l'invite GRUB Rescue.

Selon la façon dont GRUB2 a été installé, le /boot/grub/ se trouve soit dans la partition racine de la distribution Linux, soit dans la partition séparée /boot .

normal.mod analyse /boot/grub/grub.cfg, charge éventuellement des modules (par exemple pour l'interface utilisateur graphique et le support du système de fichiers) et affiche le menu.

Fonctionnement détaillé du boot

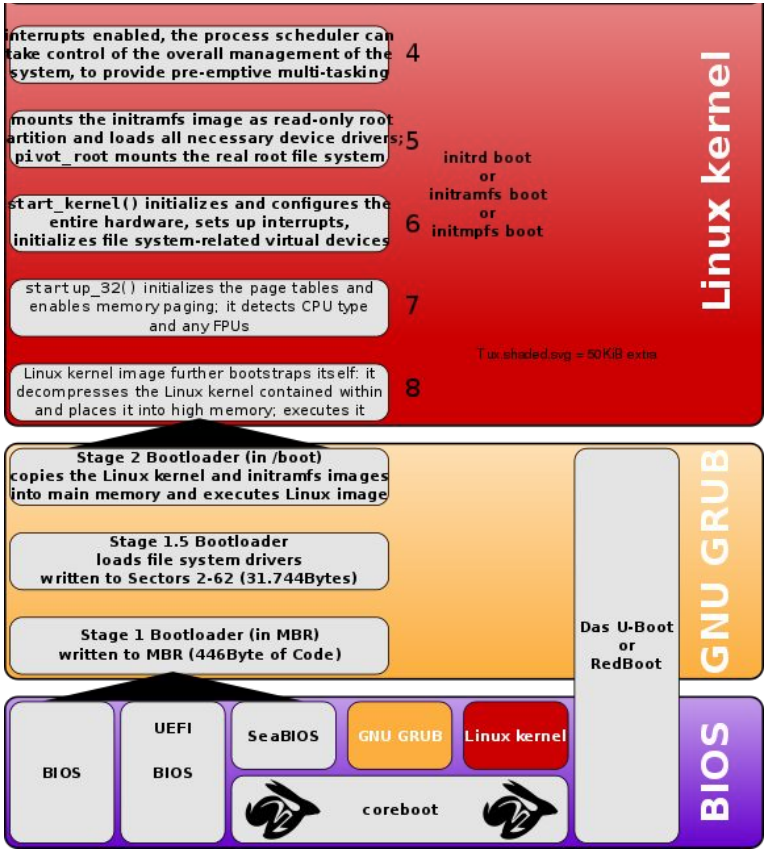
Démarrage sur les systèmes utilisant UEFI firmware

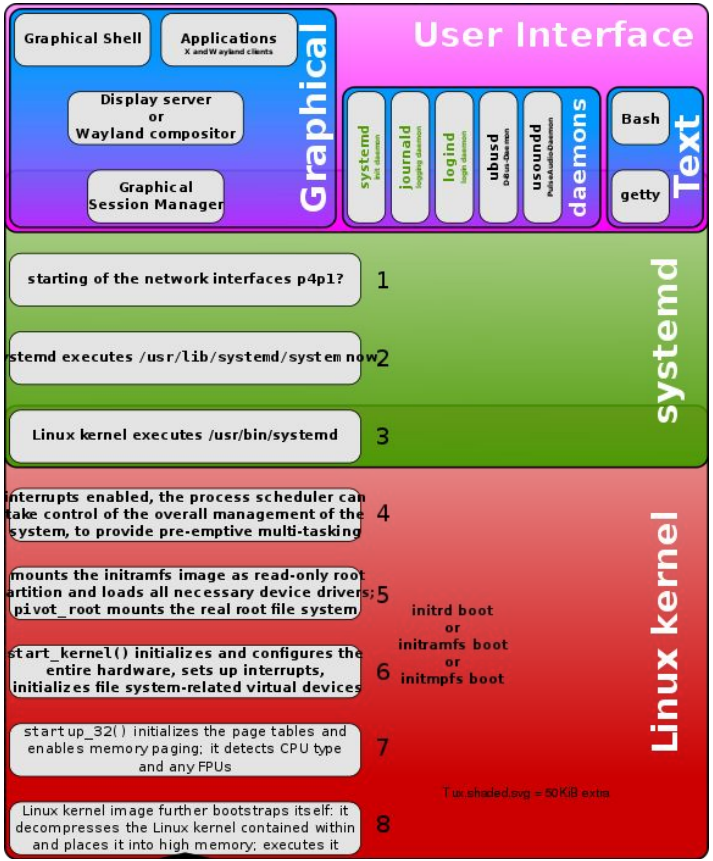
/efi/<distro>/grubx64.efi (pour les systèmes UEFI x64) est installé en tant que fichier dans la partition système EFI , et démarré directement par le micrologiciel, sans un **boot.img** sur le secteur MBR. Ce fichier est l' équivalent de **stage1** et **stage1.5**.

/boot/grub/ peut être installé sur la partition système EFI ou sur la partition séparée **/boot**

Pour les systèmes UEFI x64, stage2 correspond au fichier **/boot/grub/x86_64-efi/normal.mod** et aux autres fichiers de **/boot/grub/**.

Fonctionnement détaillé du boot





Fonctionnement détaillé du boot

Fichiers Grub2

La configuration de GRUB2 est composé de trois principales dans des fichiers inclus :

/etc/default/grub - le fichier contenant les paramètres du menu de GRUB 2,
/etc/grub.d/ - le répertoire contenant les scripts de création du menu GRUB 2, permettant notamment de personnaliser le menu de démarrage,
/boot/grub2/grub.cfg - le fichier de configuration final de GRUB 2, non modifiable. (**/boot/grub/grub.cfg** sous Debian).

Reconstruction du MBR.

On sauve le MBR du disque sda dans un fichier nommé boot.mbr à l'aide de la commande dd :
dd if=/dev/sda of=boot.mbr bs=512 count=1

On le restaure de cette manière (boot.mbr est le fichier qui a été sauvegardé ci-dessus) :
dd if=boot.mbr of=/dev/sda bs=512 count=1

Pour ne sauver que le programme et le désassembler :
dd if=/dev/sda of=pg.mbr bs=1 count=440
objdump -D -b binary -mi386 -Maddr16,data16 pg.mbr
Si la table de partition n'a pas changé, on peut très bien ne recharger que les 446 premiers octets du MBR (en indiquant count=1 bs=446).

Passage d'argument au boot.

bootparam - Introduction aux paramètres de démarrage du noyau Linux

Le noyau Linux accepte un certain nombre d'options en ligne de commandes, également appelées paramètres de démarrage, au moment où il est chargé. En général, c'est principalement utilisé pour fournir au noyau des informations sur les paramètres matériels, qu'il serait incapable de déterminer seul, ou pour éviter/remplacer les valeurs qu'il détecterait normalement.

Pour une liste des options rendons-nous directement dans le man de bootparam !

Modifier le mot de passe "perdu" de root.

Étape 1: démarrer en mode de récupération

Redémarrez votre système. Une fois que vous voyez l'écran de démarrage du fabricant de l'ordinateur, maintenez la touche Maj enfoncée. Le système devrait proposer un GRUB noir et blanc , ou menu de démarrage, avec différentes versions du noyau Linux affichées.

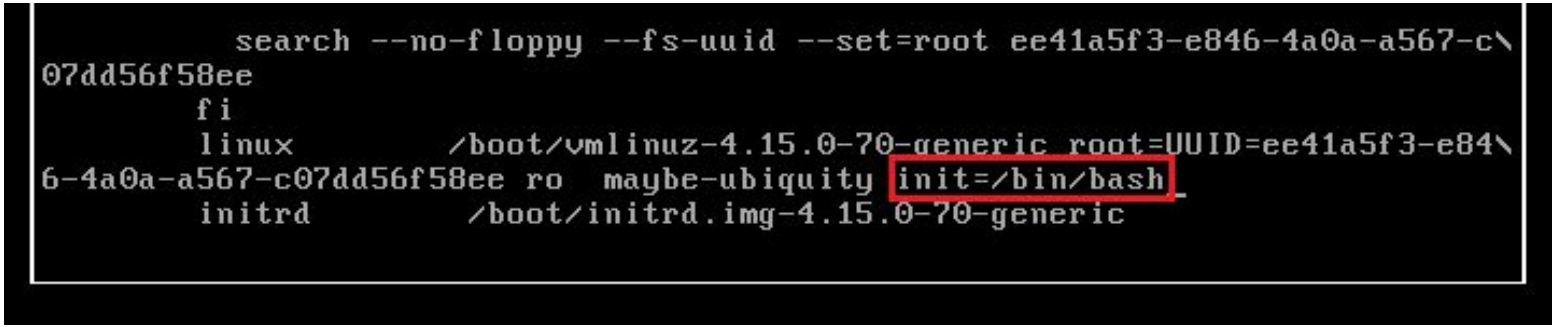
Sélectionnez la deuxième en partant du haut - la révision la plus élevée, suivie de (mode de récupération) .

Modifier le mot de passe "perdu" de root.

Étape 1: démarrer un shell depuis grub

Si il n'existe pas de mode de récupération, il est nécessaire de modifier le comportement de démarrage du système.

L'objectif est de faire exécuter **/bin/bash** en lieu de init lors du démarrage.



Modifier le mot de passe "perdu" de root.

Étape 2: remonter le système de fichiers avec des autorisations d'écriture

À l'heure actuelle, votre système n'a qu'un accès en lecture seule à votre système. Cela signifie qu'il peut consulter les données, mais ne peut apporter aucune modification. Mais nous avons besoin d'un accès en écriture pour modifier le mot de passe, nous devrons donc remonter le lecteur avec les autorisations appropriées.

À l'invite, tapez:
mount –o rw,remount /

07
Optimisation des performances

Analyse détaillée de l'occupation mémoire.

Il existe de nombreuses façons d'avoir un aperçu de l'utilisation de la mémoire vive sur un système Linux depuis la lecture directe dans /proc/ jusqu'aux outils de débogage spécialisés comme valgrind, en passant par top et htop, vmstat, pmap, memstat ou smem.

Analyse détaillée de l'occupation mémoire.

Lire les informations à la source : /proc/

/proc, il s'agit d'une système de fichiers virtuels qui sont des interfaces avec le noyau Linux, c'est ici que de nombreux utilitaires comme “top” viennent chercher leurs informations. Ici vous pouvez lire directement ces informations, mais le plus souvent sous une forme brute et peu compréhensible. C'est pour cette raison qu'on utilise souvent un programme qui se chargera de la mise en forme du résultat présenté.

Pour notre sujet à savoir l'utilisation de la mémoire vive par un programme donné, c'est /proc/$PID/maps qui va nous intéresser. “$PID” représente le numéro de processus (PID) du programme visé.

Analyse détaillée de l'occupation mémoire.

Lire les informations à la source : /proc/ && memstat

Le fonctionnement de memstat n'a rien de compliqué, et il ne fait rien d'autre que récupérer, compiler et mettre en forme a minima les données accessibles dans /proc.

Deux options sont pratiques, “-w” permet d'éviter le “line wrapping” (les lignes de résultat coupées à 80 caractères), “-p” permet de donner en argument un PID, sans cette option c'est l'ensemble de l'utilisation de la mémoire du système qui sera reportée

Analyse détaillée de l'occupation mémoire.

Smem

Pour analyser rapidement l'utilisation mémoire d'un programme smem est un outil idéal.

En plus de reporter des valeurs véritablement significatives (la colonne “rss” pour “Resident Set Size” est sans doute la valeur la plus représentative) il prend en compte l'ensemble des processus appartenant au programme.

C'est un gros avantage par rapport à un simple coup d'œil à “top” qui ne reportera que les processus à la plus grosse consommation de ressources à un moment donné.

Analyse détaillée de l'occupation mémoire.

Smem

smem sait également produire des diagrammes pour rendre compte de l'utilisation mémoire, pour cela il est nécessaire d'installer “python-matplotlib”.

L'option --pie permet de créer un diagramme circulaire (“camembert”), elle sera suivi de la valeur de référence pour les noms des processus (pid (par défaut), name), et éventuellement l'option “-s permettant de classer par valeur (rss, pss (par défaut), vss,...etc).

$ smem --pie name -s rss

Choisir le filesystem approprié

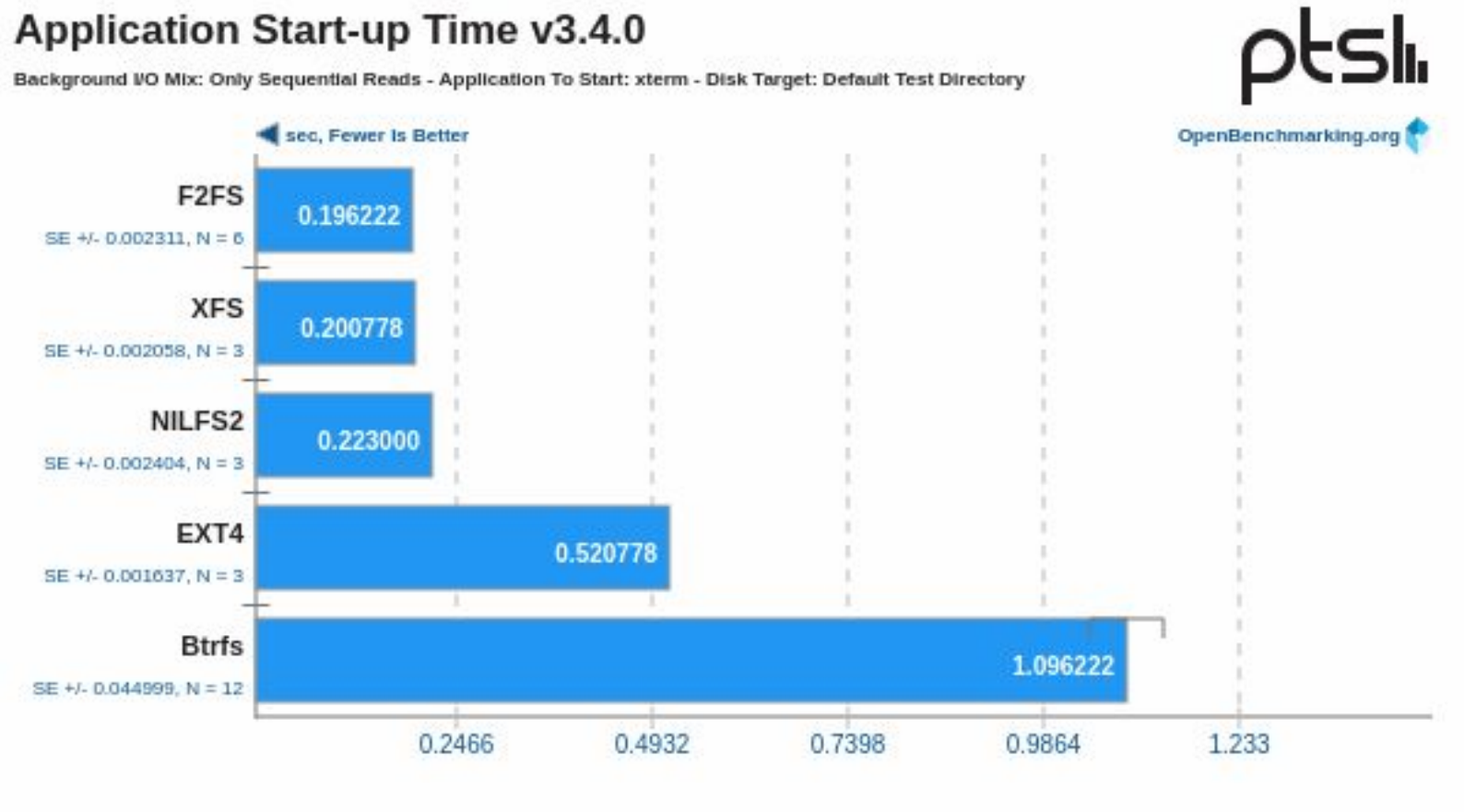
Etudes de benchmarks

Nous allons observer un benchmark réalisé par Phoronix via leur logiciel **Phoronix Test Suite**

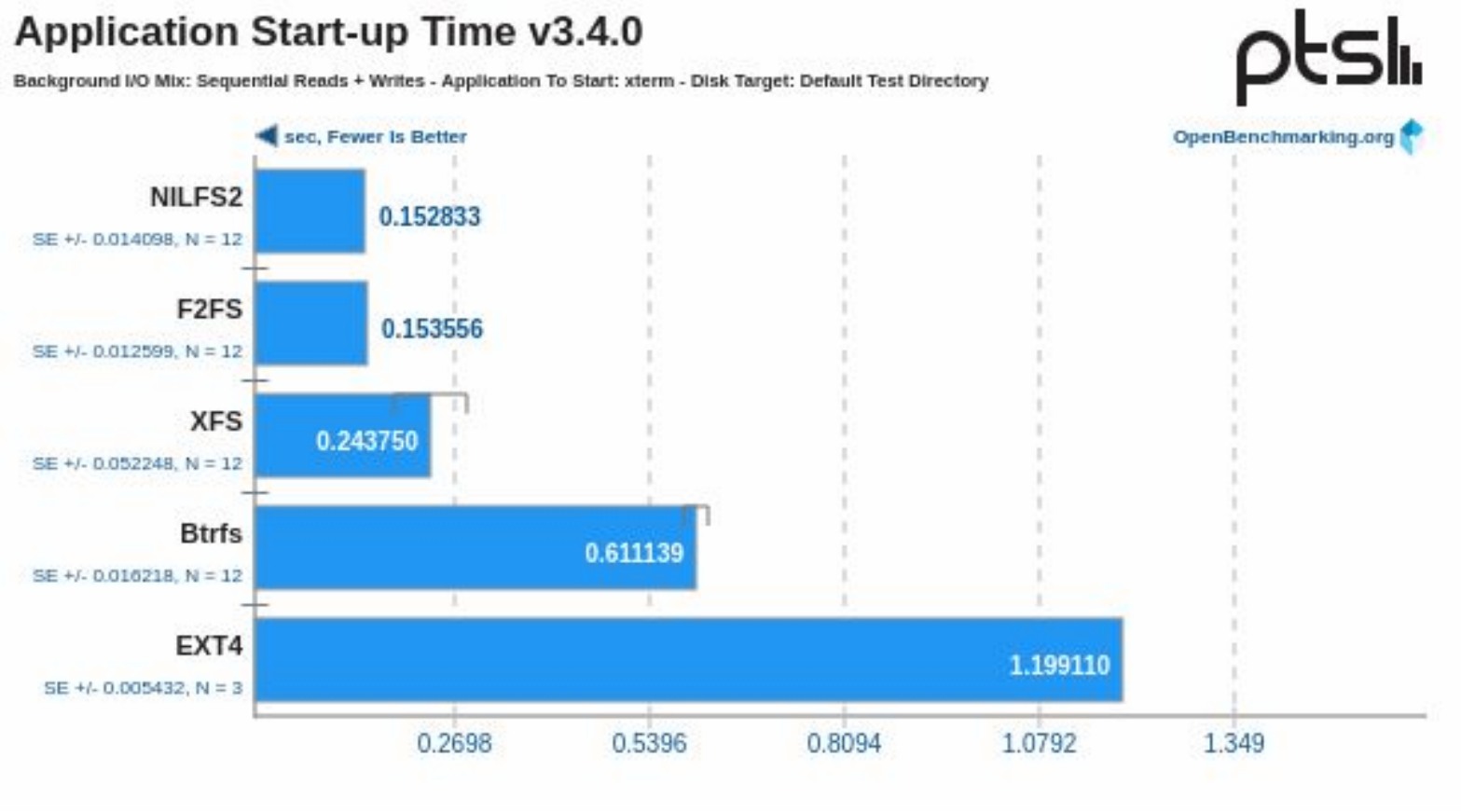
Nous allons pouvoir observer les réel résultats de EXT4/BTRFS/XFS ainsi que deux autres filesystem moins connu F2FS et le nouveau NILFS2

Les test sont réalisé sur un disque dur SDD pour comparer l'usage de ces filesysteme sur un système desktop standard.

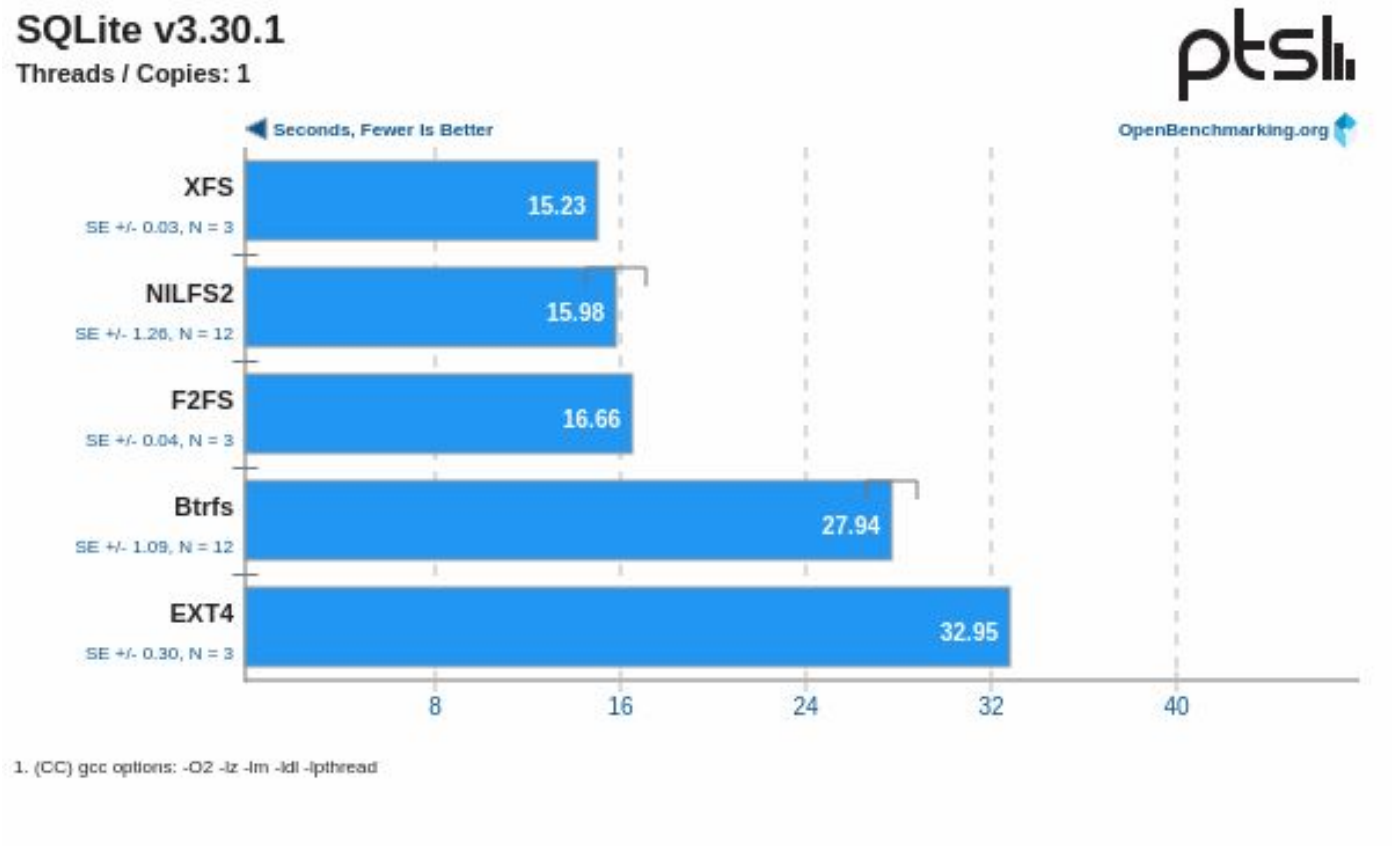
Choisir le filesystem approprié



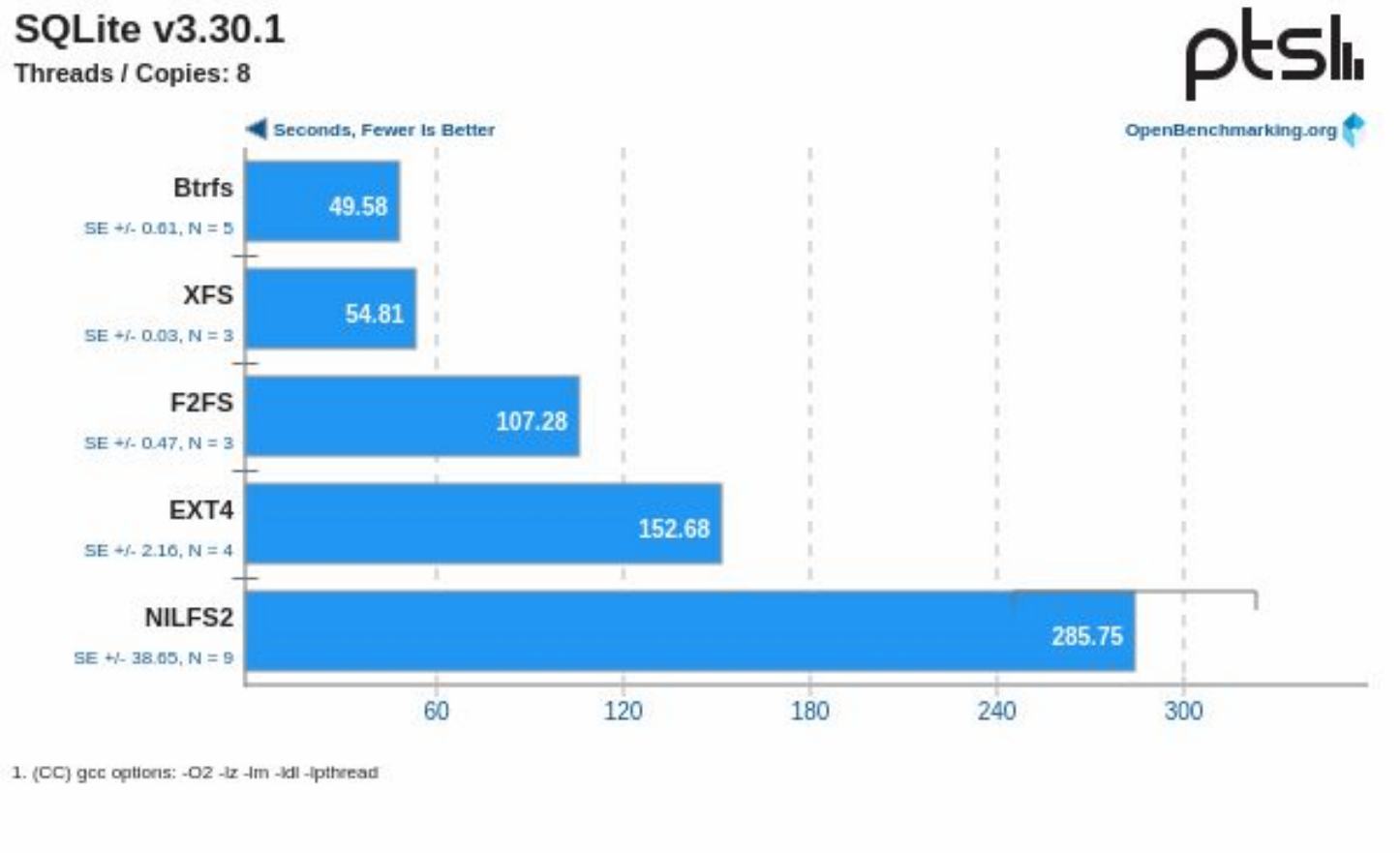
Choisir le filesystem approprié



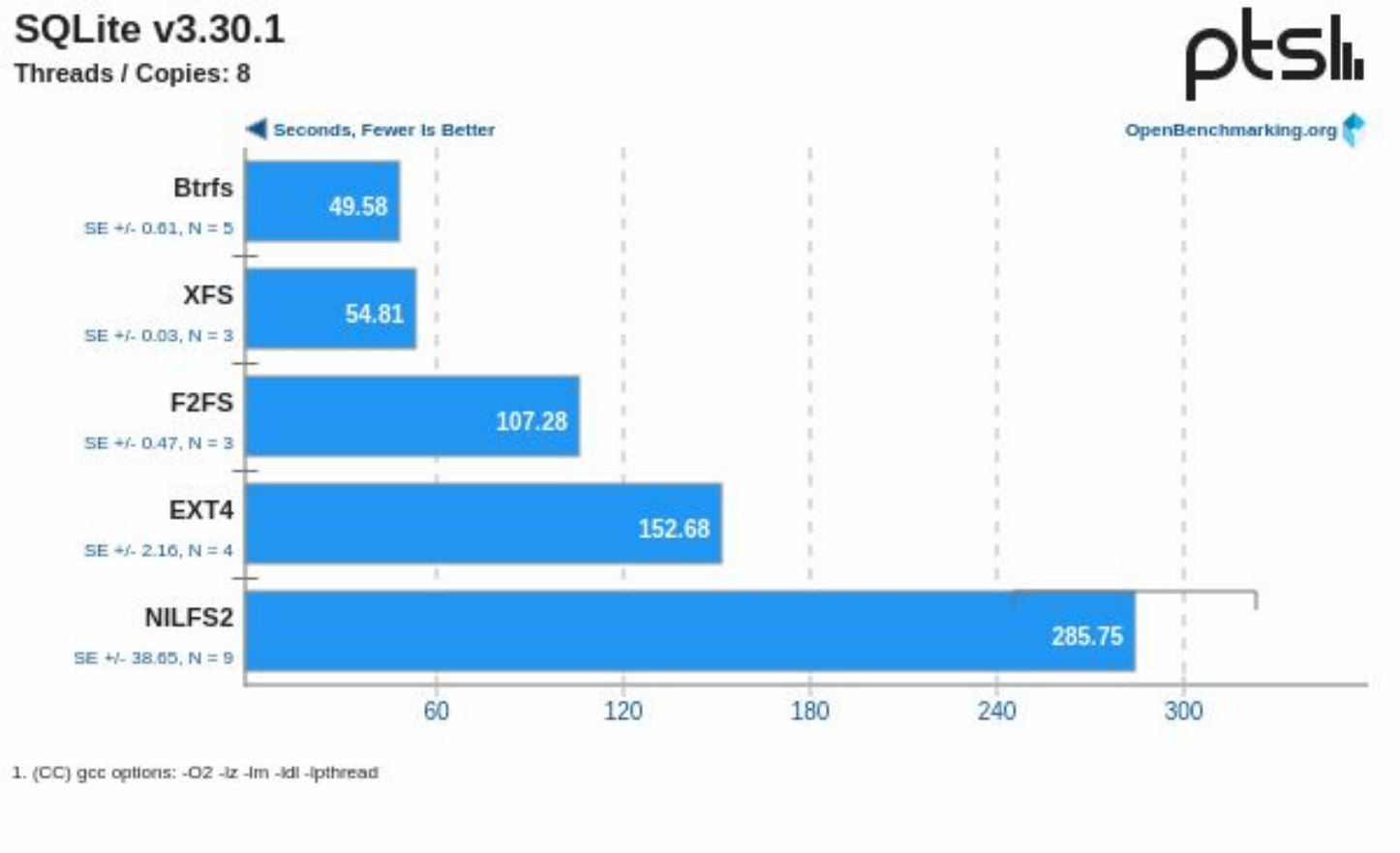
Choisir le filesystem approprié



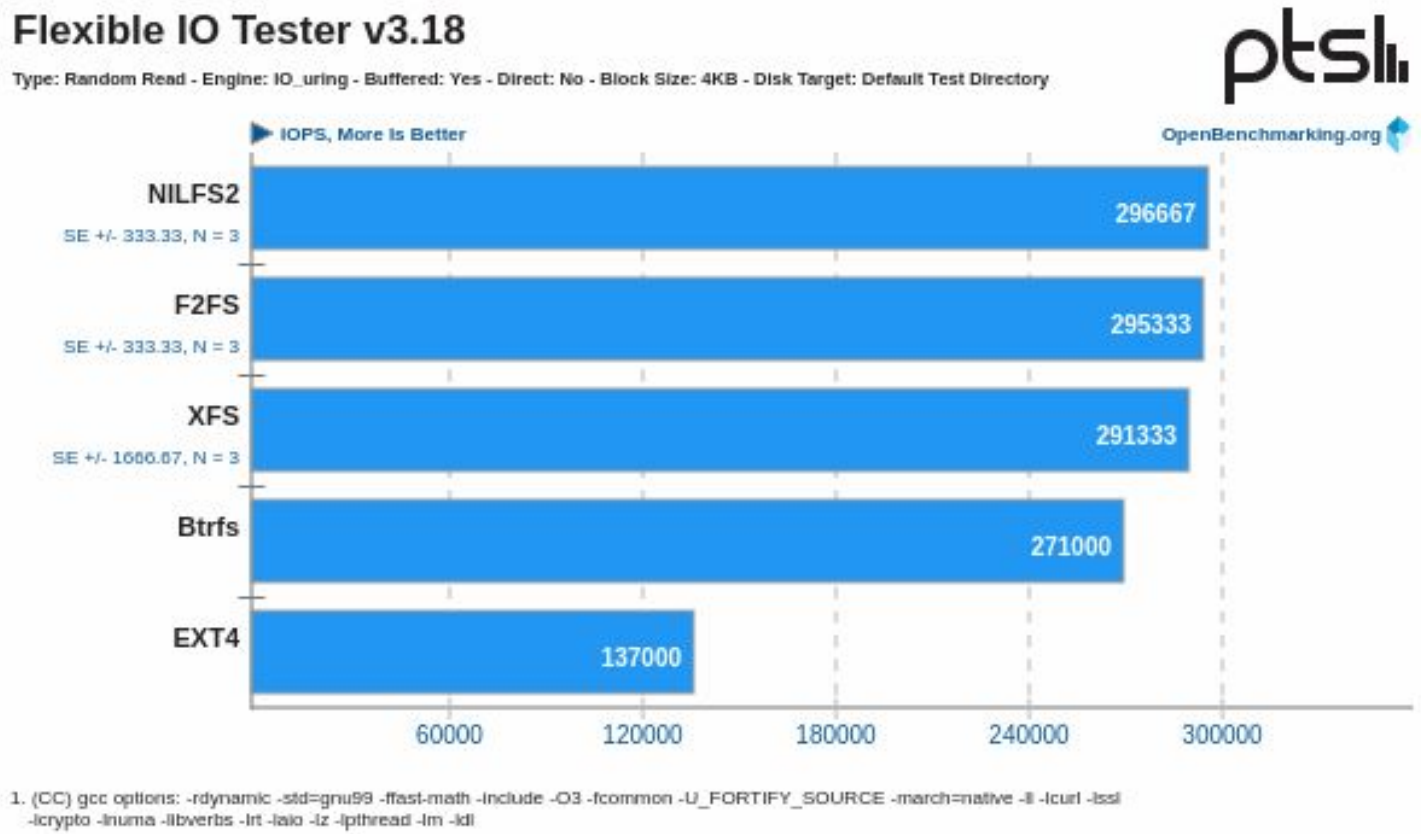
Choisir le filesystem approprié



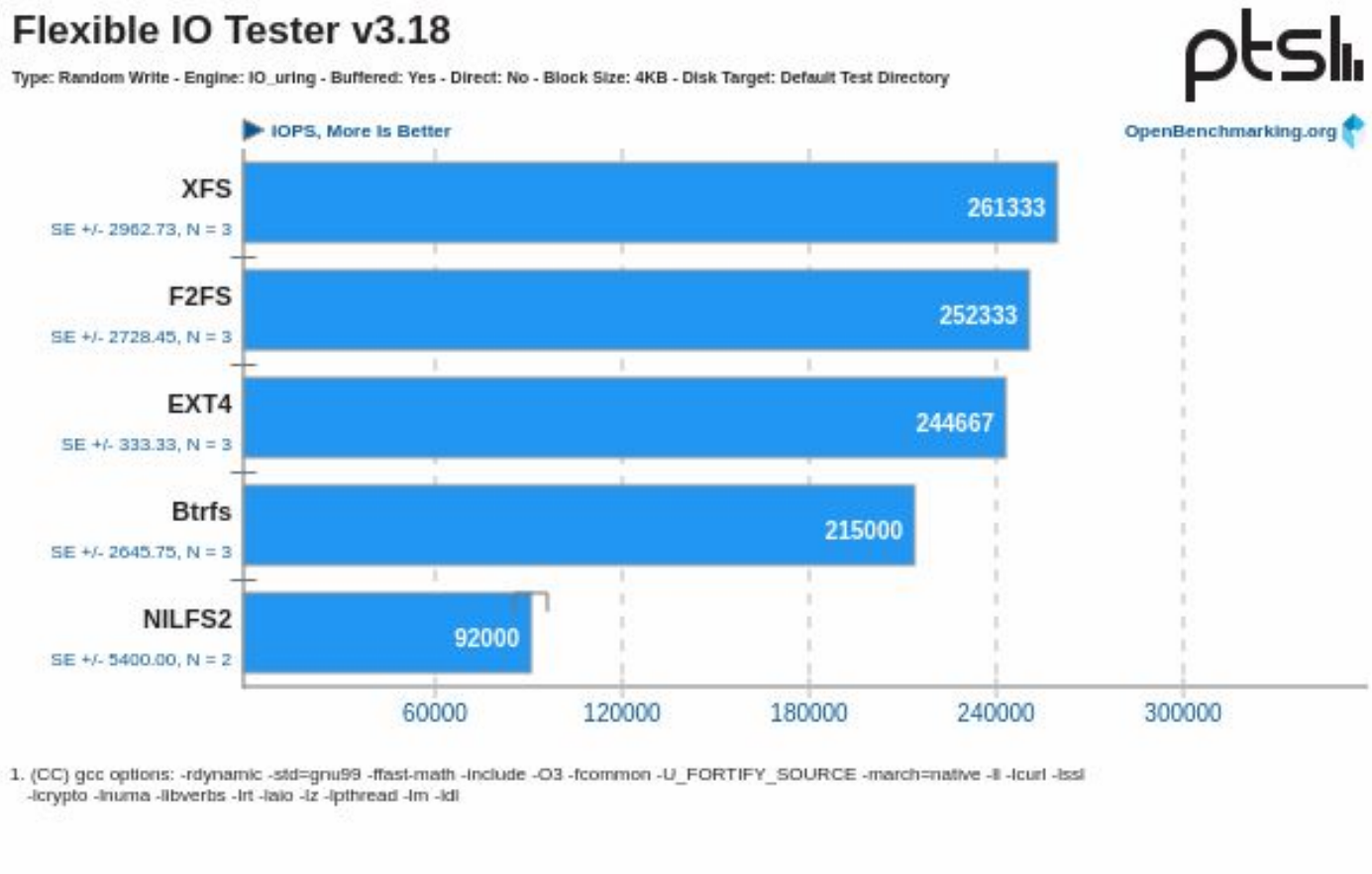
Choisir le filesystem approprié



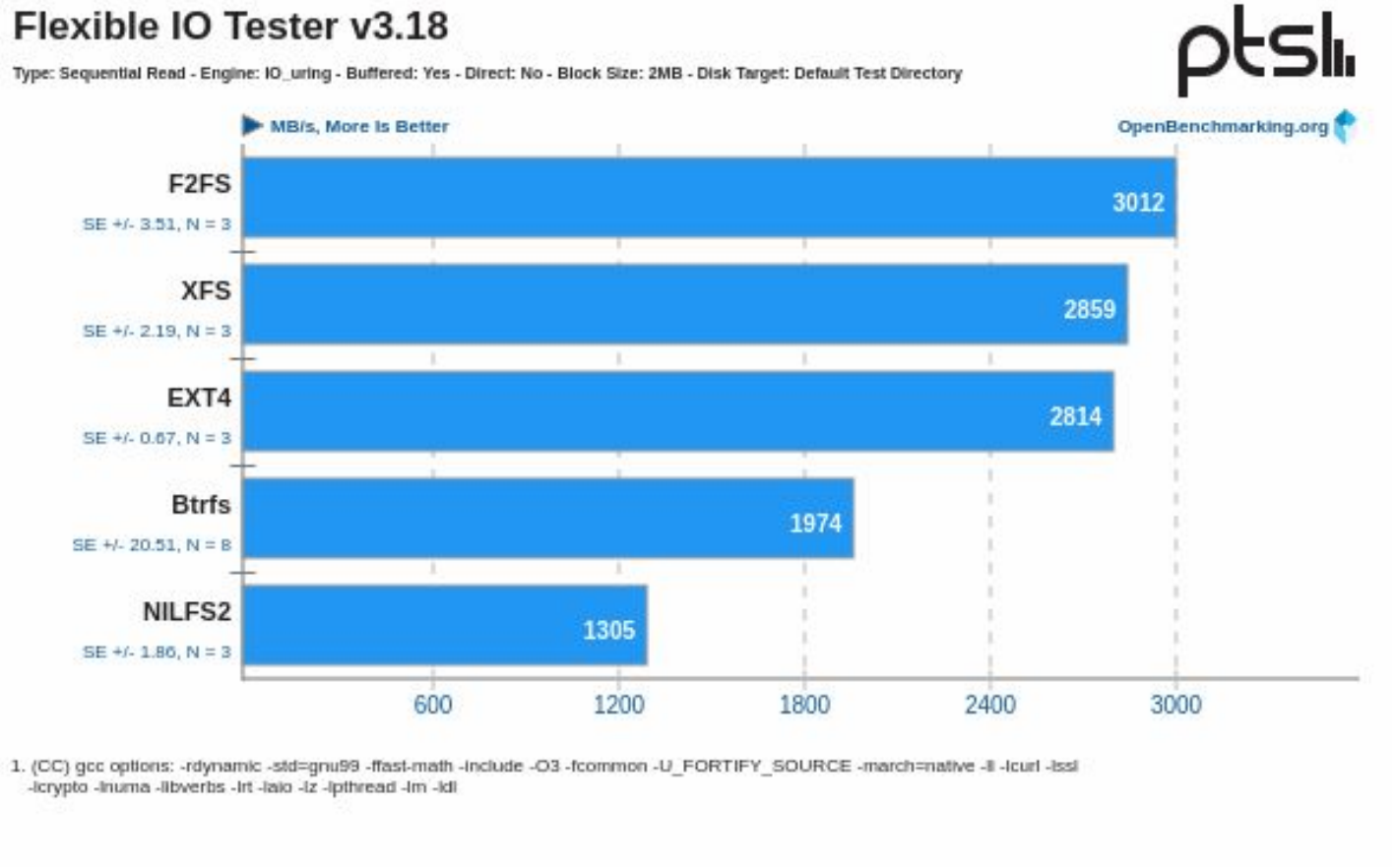
Choisir le filesystem approprié



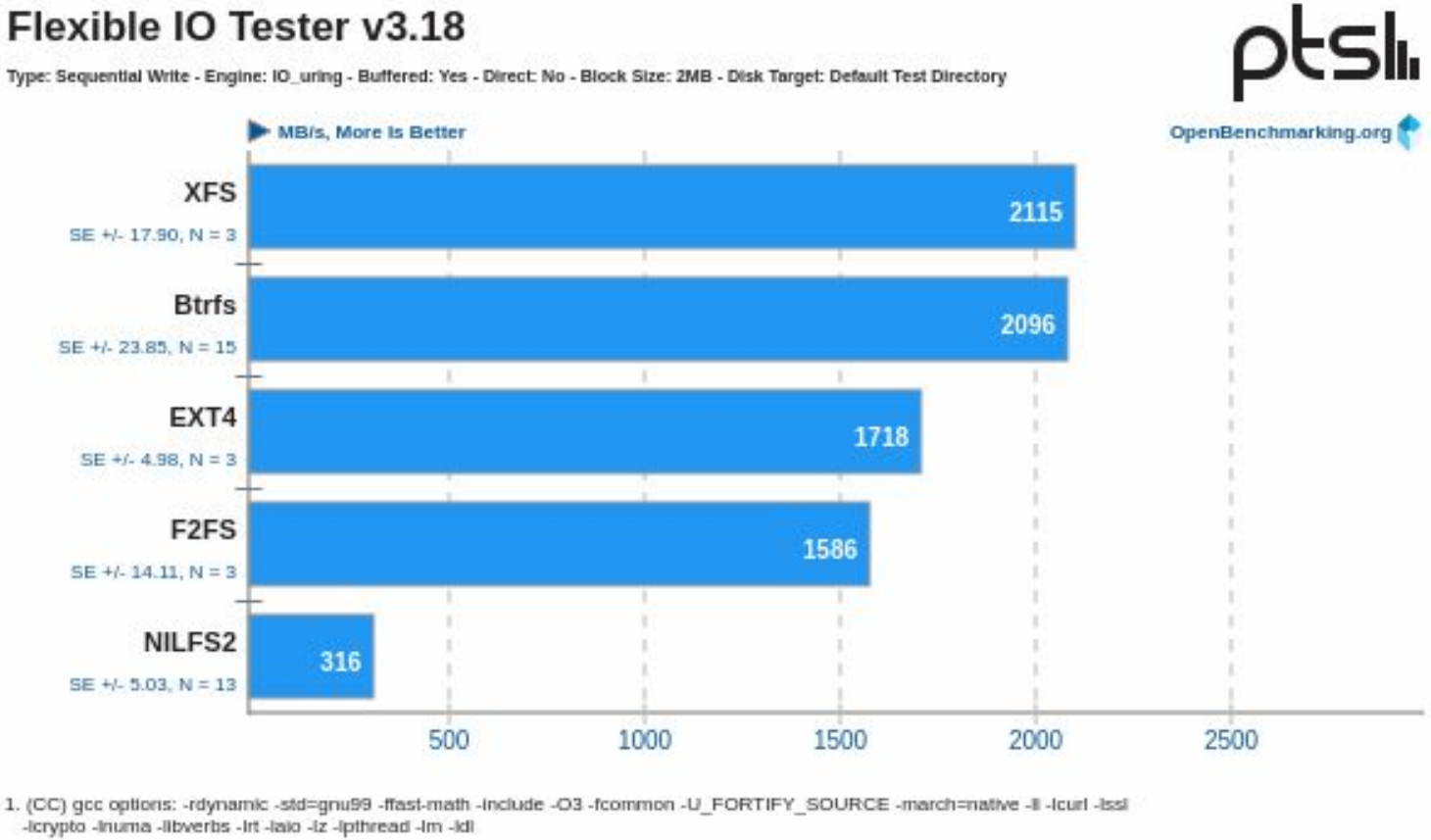
Choisir le filesystem approprié



Choisir le filesystem approprié



Choisir le filesystem approprié



Tuning des filesystems.

La durée pendant laquelle un système utilise les tâches de disque IO (In / Out) a un impact significatif sur les performances globales du système.

Le système de fichiers utilisé par un disque et la manière dont il est réglé peuvent considérablement influencer le temps qu'un système utilise pour attendre que les données soient écrites ou lues sur le disque.

Quel système de fichiers dois-je choisir?

On lit souvent :

- xfs est très efficace pour stocker principalement des fichiers très volumineux. xfs peut être un bon choix pour un lecteur de 200 Go dédié au stockage de votre collection XviD (principalement des fichiers de 350 Mo et 700 Mo).
- jfs est également très efficace pour les lecteurs qui stockent principalement de petits fichiers.
- ext3 ne semble être recommandé par personne pour ses performances. Cependant, il est dit "stable" et "non sujet à la perte de données". Ces propriétés sont dans de nombreux cas bien plus importantes que les performances.

Quel système de fichiers dois-je choisir?

Avez-vous besoin d'enregistrer les temps d'accès?

GNU / Linux enregistre les temps d'accès , appelés atime , lorsque les fichiers sont écrits. Cela peut être une bonne chose , cela vous permet de rechercher les derniers fichiers consultés dans un intervalle donné, de trier les fichiers en fonction des temps d'accès, etc.

Mais avez-vous vraiment besoin de ces informations? L'enregistrement occasionnel entraîne une charge système importante sur la plupart des systèmes de fichiers. Le désactiver donne un gain de performances sur la plupart des systèmes de fichiers,

Quel système de fichiers dois-je choisir?

Ce que vous devez faire pour le désactiver est d'ajouter noatime à votre fichier / etc / fstab .

/dev/hda5/home btrfs noatime, notail, rw, user, exec, auto 1 1
/dev / hda7/pub btrfs noatime, rw, exec, auto 1 1

Quel système de fichiers dois-je choisir?

Tailles de bloc

Une grande taille de bloc peut donner des gains de performances significatifs sur les systèmes de fichiers qui stockent principalement des fichiers très volumineux.

La taille de bloc par défaut est de **1024** octets. Les débats sur la liste de diffusion du noyau Linux au cours des dernières années sont remplis d'arguments selon lesquels une taille de bloc pour 4096 devrait être la valeur par défaut car elle contient moins de fragmentation de fichiers, un fsck plus rapide, une suppression de fichier plus rapide et une vitesse de lecture plus rapide car une taille de bloc plus grande entraîne moins de recherches.

Quel système de fichiers dois-je choisir?

Tailles de bloc

La modification de la taille du bloc pour augmenter les performances n'est pas une option dans la plupart des cas car elle nécessite de reformater le système de fichiers.

Cependant, c'est quelque chose que vous voudrez peut-être prendre en compte lors de l'installation de nouveaux disques et systèmes.

La commande format vous permet de changer la taille du bloc avec -b size :

```
mke2fs -b 4096 /dev/hdx
```

l'intérêt de Xinetd

xinetd est le remplaçant de TCP Wrappers (inetd). Il est installé maintenant par défaut sur RedHat.

Il gère la connexion de certains protocoles (telnet, pop, ftp,....) et permet d'autoriser ou d'interdire les connexions sur votre machine en fonction de votre adresse IP, du nom d'hôte du client ou de son domaine, de l'heure, du taux de charge, du nombre de connexions simultanées, du nombre de connexions entrantes par seconde, ce qui permet de limiter les tentatives de Deny of Service.

l'intérêt de Xinetd

Le fichier xinetd.conf

Ce fichier se trouve dans le répertoire /etc. Il permet d’indiquer la configuration par défaut qui s'applique à tous les services et l’emplacement des fichiers de configuration propre à chaque service qui utilisent xinetd.

Donnez dans ce fichier la politique par défaut que vous allez appliquer à toutes les applications qui utilisent xinetd

l'intérêt de Xinetd

Le fichier xinetd.conf

La syntaxe est la suivante :

defaults
{
disable = yes tout est désactivé par défaut
instances = 20 Nombre de requêtes simultanées que xinetd peut gérer.
no_access = 0.0.0.0/0 par défaut aucun réseau ne peut se connecter. On peut à la place utiliser no_access.
log_type = SYSLOG authpriv Envoyé à syslog comme authpriv.info.
log_on_failure = HOST RECORD Enregistre lorsque la connexion échoue HOST enregistre le nom, RECORD tout ce qui peut être enregistré sur le client.
log_on_success = HOST USERID DURATION PID Enregistre en cas de succès de la connexion, HOST le nom, USERID l'utilisateur, DURATION la durée de connexion, PID le pid du serveur.
per_source = 4 on n'autorise que 4 connexions en provenances de la même machine.

}
includedir /etc/xinetd.d Indique le répertoire ou se trouvent les fichiers de configuration par service.

l'intérêt de Xinetd

Les fichiers de configuration par service

service telnet
{
disable = no On suppose ici que vous souhaitez l'activer
flags = REUSE
instances = UNLIMITED Pas de limitation sur le nombre de requêtes possibles, à éviter
only_from = 172.16.0.0/16 n'autorise la connexion que depuis les adresses 172.16.0.0 avec un masque de 255.255.0.0
only_from = .ac-creteil.fr idem mais depuis des machines du domaine ac-creteil.fr
only_from = 10.0.0.{10,11,12} idem mais seulement depuis 10.0.0.11, 10.0.0.12, 10.0.0.13.
socket_type = stream type de service de transport de données (il existe stream pour tcp, dgram pour udp, raw pour IP)
wait = no état d'attente, si l'état est wait xinetd doit attendre que le serveur ait restitué la socket avant de se remettre à l'écoute. On utilise wait plutôt avec les types dgram et raw. l'autre possibilité est nowait qui permet d'allouer dynamiquement des sockets à utiliser avec le type stream.
user = root Nom de l'utilisateur sous lequel le daemon tourne
server = /usr/sbin/in.telnetd Chemin d'accès au programme in.ftpd lancé par inetd (il est possible ici d'ajouter les options de démarrage du programme.
}

Booter rapidement son système.

L'amélioration des performances de démarrage d'un système peut réduire les temps d'attente au démarrage et permet d'en savoir plus sur la manière dont certains fichiers système et scripts interagissent les uns avec les autres.

Analyse du processus de démarrage

systemd fournit un outil appelé **systemd-analyze** qui peut être utilisé pour afficher des détails de synchronisation sur le processus de démarrage, y compris un diagramme svg montrant les unités en attente de leurs dépendances. Vous pouvez voir quels fichiers unitaires ralentissent votre processus de démarrage.
Vous pouvez alors optimiser votre système en conséquence.

Booter rapidement son système.

Compiler un noyau personnalisé

La compilation d'un noyau personnalisé peut réduire le temps de démarrage et l'utilisation de la mémoire. Bien qu'avec la standardisation de l'architecture 64 bits et la nature modulaire du noyau Linux, ces avantages peuvent ne pas être aussi importants que prévu.

Initramfs

Dans une approche similaire à la compilation d'un noyau personnalisé , les initramfs peuvent être allégées.

Booter rapidement son système.

Démarrage précoce des services

Une caractéristique centrale de systemd est l' activation du D-Bus et du socket.
Cela provoque le démarrage des services lors de leur premier accès et c'est généralement une bonne chose.

Cependant, si vous savez qu'un service (comme upower) sera toujours démarré pendant le démarrage, alors le temps de démarrage global peut être réduit en le démarrant le plus tôt possible.
Ceci peut être réalisé (si le fichier de service est configuré pour lui, ce qui dans la plupart des cas c'est le cas) en émettant:

systemctl enable upower

08
Supervision

La supervision.

La supervision regroupe un ensemble d’outils et de pratique permettant de remonter et d’agir sur les éléments technique d’un réseau.

L'intérêt pour l’entreprise est de garder un niveau de disponibilité maximum au niveau de ses service.

Pour l’administrateur en charge de ce réseau c’est la possibilité de visualiser l’état de santé de son réseau , de prévenir les pannes et rapidement pouvoir identifier un incident et ses impacts.

La supervision.

Définir le périmètre de supervision

La supervision est un domaine vaste qui englobent de multiple domaine de compétence. Il est vital de comprendre ce que l'on veut tirer de la mise en place d'une supervision.

Est-ce que l'on veut :
Visualiser , Surveiller ou Agir

Même si aujourd'hui les outils permettent de regrouper ses actions , nous allons voir que les méthodes diffères en fonction de notre mode de fonctionnement et des besoins de l'entreprise.

La supervision.

Visualiser c'est mettre en évidence les points sensible de son architecture. On peut voir à tout instant le niveau de santé de notre réseau. L'ajout de fonction de cartographie dans la supervision aide la prise de décision.

Avec une bonne cartographie d'un réseau nous pouvons suivre l'évolution d'une intervention sur celui-ci par exemple. Suivre la montée en charge d'une nouvelle application. Comme on le verra ensuite , cela donne notamment la possibilité de vérifier une remontée d'alerte.

Le point le plus important est la capacité à prévoir les mutations de notre architecture. Voir les points d'engorgement du réseau et évité une perte de service à cause de débit trop important sur un lien faible.

La supervision.

La surveillance d'un réseau peut se faire de deux manière :

Activement : C'est notre solution de supervision qui va à la recherche d'information. On vient donc interroger un indicateur sur une information bien précise ou pour récupérer le total des informations disponibles.

Passivement : Dans ce cas c'est une fonctionnalité externe à notre superviseur qui va lui renvoyer une information. Dans la plupart des solutions celle-ci sera enregistrée par un collecteur puis classifié comme un contrôle actif.

Les résultats des contrôles qu'ils soient actifs ou passifs seront ensuite analysé par le superviseur qui gérera ensuite les alertes en fonction des alarmes que nous définissons.

La supervision.

En fonction de la méthode utilisée pour superviser un équipement , il est possible de pousser des modifications sur les équipements.
Cette méthode demande de connaître parfaitement la configuration de l’ équipement.

Mais son avantage majeur est la possibilité de scripter une réponse à une alerte. En fonction d’un événement reçu le serveur de supervision peut envoyer des instructions à un équipement.

La supervision.

Vous verrez que bien souvent la supervision est comparée au système nerveux de l'humain. Et ce pour la bonne raison que si chaque connexion neuronale est important , chaque noeud d'un réseau est important.

Tout cela pour dire qu'il est nécessaire de bien connaître son réseau et les systèmes qui y vivent.

On regroupe les différents partis d'un SI comme tel :

- Supervision réseau et matérielle
- Supervision système
- Supervision des services et des applications

Il n'y a malheureusement pas de supervision des utilisateurs ...

La supervision.

Choix des indicateurs :

Nous nous concentrerons sur la supervision réseau et matérielle dans notre cas.

Toujours en connaissance du but à atteindre nous allons jouer sur les innombrables indicateurs disponible dans les commutateurs , les routeurs , les serveurs etc ..

Un indicateurs correspond à une information. Comme connaître la disponibilité d'un service , les débits d'un lien , la charge cpu d'un serveur etc ..

La supervision.

Même s'il n'y a pas de recette magique et que tous est conditionné par le but de votre supervision , nous retiendrons les indicateurs vitaux suivant :

État des liens et des ports (up , down , disable , block)
Débit des liens (Reperer les gouleaux d'étranglement)

Ligne de vie des :

- Cpu (charge processeur)
- Disque dur (espace libre)
- Ventilateurs (Médianne de la température)
- Mémoire (charge mémoire)

La supervision.

Principale solution de supervision sur le marché :
Solution libre : Nagios

Nagios, créé à l'origine sous le nom de NetSaint, a été écrit et est actuellement maintenu par Ethan Galstad avec un groupe de développeurs qui maintiennent activement les plugins officiels et non officiels.

NetSaint est devenue Nagios (pour “Nagios Ain't Gonna Insist On Sainthood”) suite à une contestation par les propriétaire de la marque NetSaint.

La supervision.

Vue d'ensemble

- Suivi des ressources de l'hôte (charge processeur, l'utilisation du disque, les journaux système) sur la majorité des systèmes d'exploitation de réseau, y compris Microsoft Windows avec le NSClient + + plugin ou Vérifier MK.
- Suivi de toute autre chose comme les sondes (température, alarmes, etc) qui ont la capacité d'envoyer des données collectées via un réseau de plugins écrits spécifiquement
- Surveillance via des scripts à distance gérées via Nagios exécuteur Plugin distance

La supervision.

- La surveillance à distance via SSH ou par un tunnel cryptés en SSL.
- Un design simple de plugin qui permet aux utilisateurs de développer facilement leurs propres contrôles de service en fonction des besoins, en utilisant leurs outils de choix (scripts shell, C + +, Perl, Ruby, Python, PHP, C #, etc)
- La gestion et création de graphique
- Les contrôles de service parallélisés
- La possibilité de définir des hiérarchies d'accueil du réseau en utilisant des hôtes "parents", permettant la détection et la distinction entre les hôtes qui sont en panne ou inaccessible

La supervision.

- Notifications lorsque des problèmes de service ou de l'hôte se produisent et se résolve (via e-mail, Bippeur, SMS, ou toute autre méthode définie par l'utilisateur par le système de plugin)
- La possibilité de définir des gestionnaires d'événements pour être exécuté pendant les événements de service ou d'hôte pour la résolution pro-active des problèmes
- Rotation automatique des log
- Une interface web optionnelle pour visualiser l'état actuel du réseau, les notifications, etc...
- Stockage de données via des fichiers de texte plutôt que des bases de données

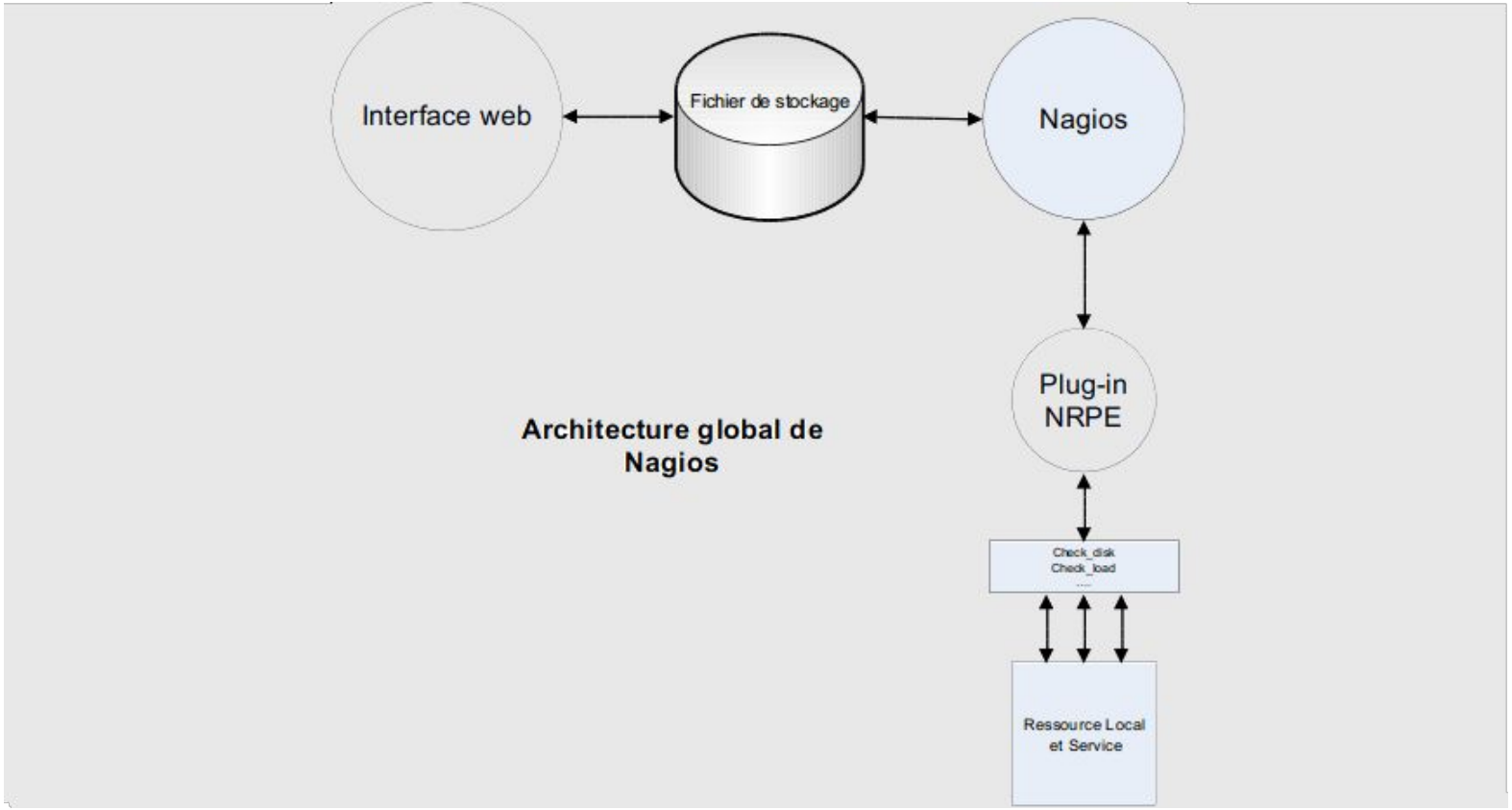
La supervision.

Architecture Global

L'architecture globale du système se compose de cinq entités :

- Le serveur récupère les informations des plugins et agit en conséquences (alertes, ..)
- Les plugins effectuent des actions en vue d'obtenir une valeur sur l'objet distant
- La base de données ou fichiers de stockages qui permettent de conserver les informations collectés
- Un objet distant fournit les informations que souhaite observer l'utilisateur
- Une interface web qui permet d'observer les activités

La supervision.



La supervision.

Contrôle Actif ou contrôle passif ?

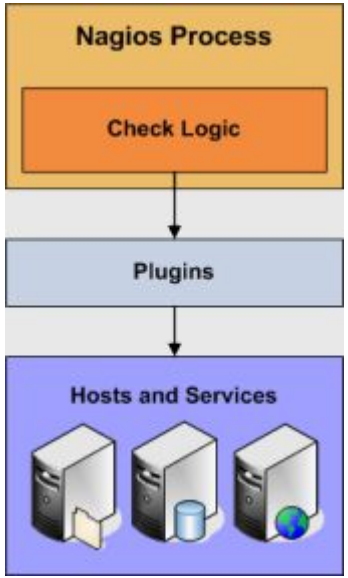
Nagios peut récupérer les informations des postes qu'il supervise par deux moyens. Soit par le mode passif, soit par le mode actif.

- Le mode passif : l'action est réalisée par une application externe venant de l'équipement supervisé. La charge processeur pour Nagios est quasi nulle.

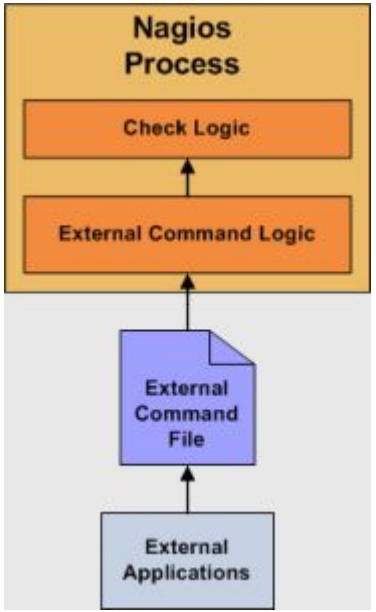
- Le mode actif : l'action est réalisée par Nagios. Quand Nagios a besoin de connaître l'état d'un hôte ou d'un service, il va exécuter un plugin et lui donner les informations sur ce qu'il a besoin de contrôler. Le plugin contrôle alors l'état opérationnel de l'hôte ou du service et renvoie les résultats en retour au daemon Nagios.

La supervision.

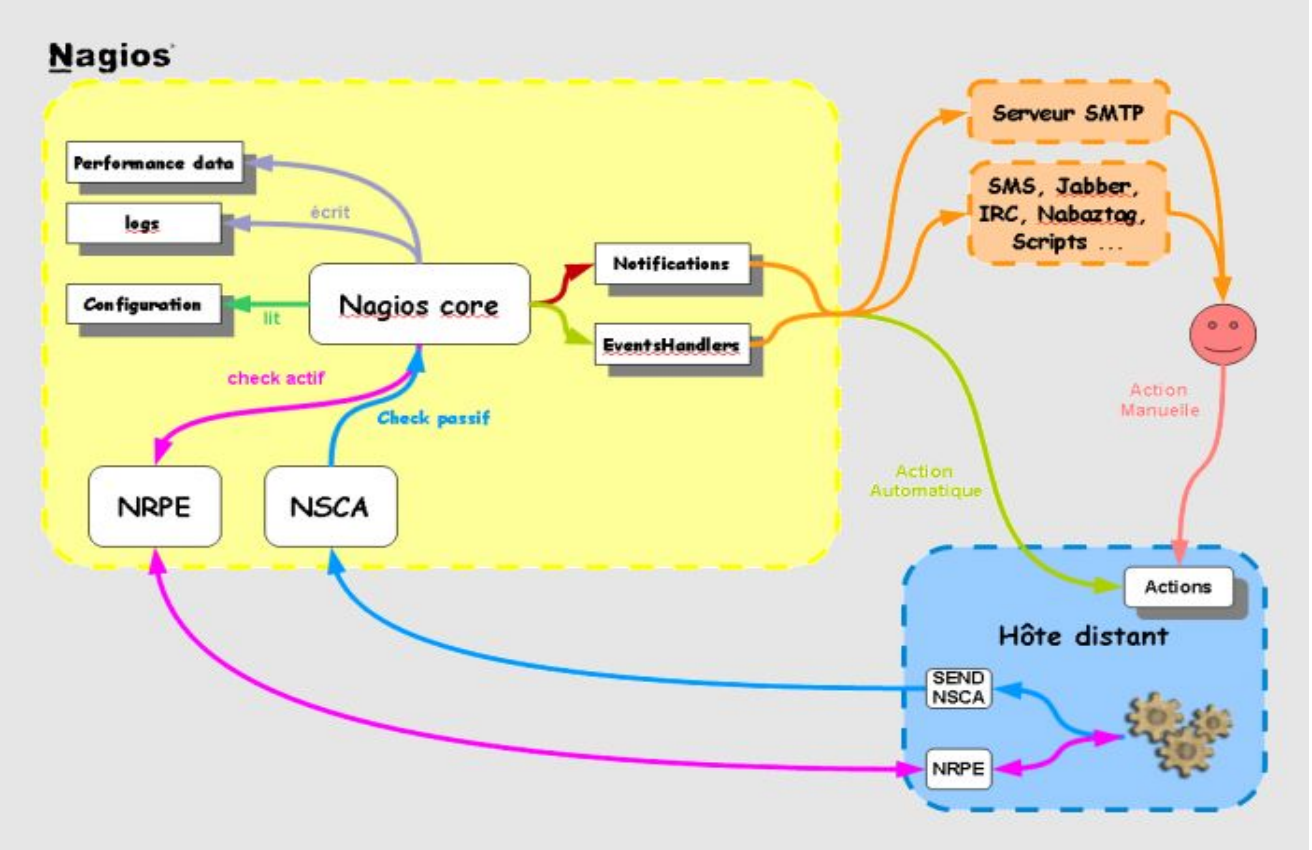
Mode actif



Mode passif



La supervision.



La supervision.

Les plugins

Les plugins (un programme exécutable ou un script qui se lance en ligne de commande) vont permettre de tester un poste ou un service. C'est le résultat de l'exécution du plugin qui est interprété par Nagios pour déterminer l'état du service ou de la station testée.

Tous les plugins fonctionnent avec les conventions suivantes :

- 0 : OK — Le service fonctionne correctement
- 1 : Warning, — Le service est dégradé
- 2 : Critical — Le service ne fonctionne plus
- 3 : Unknown — État du service indisponible

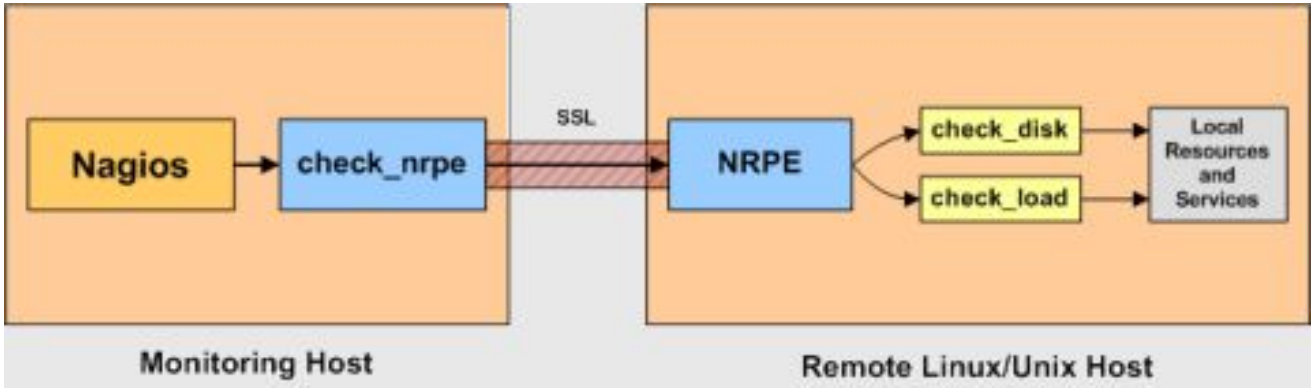
La supervision.

Les principaux plugins utilisés par Nagios sont :

- check_disk : Vérifie l'espace occupé d'un disque dur
- check_http : Vérifie le service "http" d'un hôte
- check_mysql : Vérifie l'état d'une base de données MYSQL
- check_nt : Vérifie différentes informations (disque dur, processeur ...) sur un système d'exploitation Windows
- check_nrpe: Permet de récupérer différentes informations sur les hôtes
- check_ping: Vérifie la présence d'un équipement, ainsi que sa durée de réponse
- check_pop: Vérifie l'état d'un service POP (serveur mail)
- check_snmp : Récupère diverses informations sur un équipement grâce au protocole SNMP (Simple Network Management Protocol)

La supervision.

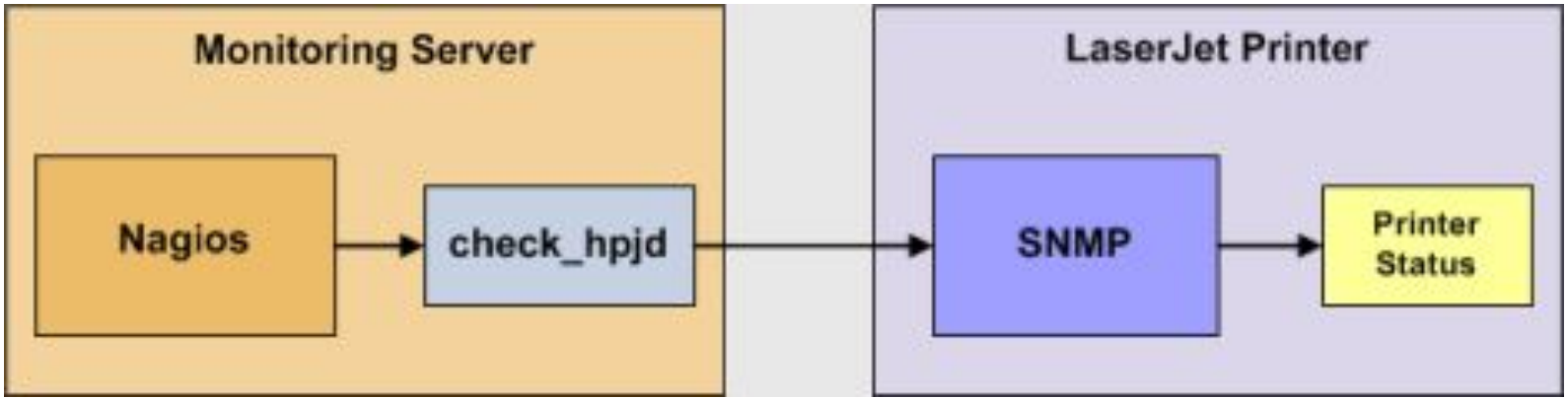
Nous utiliserons le plugin NRPE pour superviser le serveur Nagios. NRPE permet la remontée de nombreuse informations sur un poste UNIX/LINUX. Nous surveillerons donc la charge CPU, la taille des partitions, le nombre de processus (Ainsi que les processus Zombie.)



La supervision.

Check_hpjd

Plugins à la base développée pour les imprimantes HP il se généralise à toutes les imprimantes disposant d'une carte externe/interne Jet Direct permettant la communication via le protocole SNMP



La supervision.

Configuration de Nagios :
Dans le dossier /opt/nagios/etc/ on trouve les fichiers :

cgi.cfg
- Définition des paramètres des scripts CGI. Vous pouvez utiliser le fichier fourni par défaut par Nagios.

nagios.cfg
- Fichier de configuration de Nagios. A modifier par vos soins selon votre configuration et l'arborescence choisie. Vous pouvez partir du fichier fourni en standard par Nagios et le modifier selon votre configuration.

resource.cfg
- Définition des ressources externes. Vous pouvez utiliser le fichier fourni par défaut par Nagios.

objects/
- C'est dans ce sous-répertoire que sont centralisé les définitions des machines et services à surveiller part votre serveur Nagios.

La supervision.

objects/commands.cfg

- C'est là que nous allons définir les commandes utilisées par Nagios pour interroger vos machines. Vous pouvez partir du fichier fourni en standard par Nagios et le modifier selon votre configuration.

objects/contacts.cfg

- Dans ce fichier, il faut configurer les contacts pouvant être prévenu en cas d'alerte. Vous pouvez partir du fichier fourni en standard par Nagios.

objects/hostclients.cfg

- Ce fichier est à créer, il comportera la définition de toutes vos machines clients à surveiller (c'est à dire les postes utilisateurs). Ce fichier n'existe pas dans la structure de base de Nagios.

objects/hostservers.cfg

- Ce fichier est à créer, il comportera la définition de toutes vos machines serveurs à surveiller (c'est à dire les serveurs Web, DNS, DB...). Ce fichier n'existe pas dans la structure de base de Nagios.

La supervision.

objects/localhost.cfg

- Ce fichier est là pour que Nagios puisse surveiller le serveur sur lequel il est installé (localhost). Vous pouvez partir du fichier fourni en standard par Nagios.

objects/templates.cfg

- C'est le fichier ou se trouve la définition des "templates". Vous pouvez partir du fichier fourni en standard par Nagios.

objects/timeperiods.cfg

- Ce fichier défini les périodes de temps. Vous pouvez partir du fichier fourni en standard par Nagios.

objects/network.cfg

- Ce fichier est à créer, il comporte la définition de toutes les machines composant l'infrastructure de votre réseau (routeur, switch ou hub, borne Wifi ...)

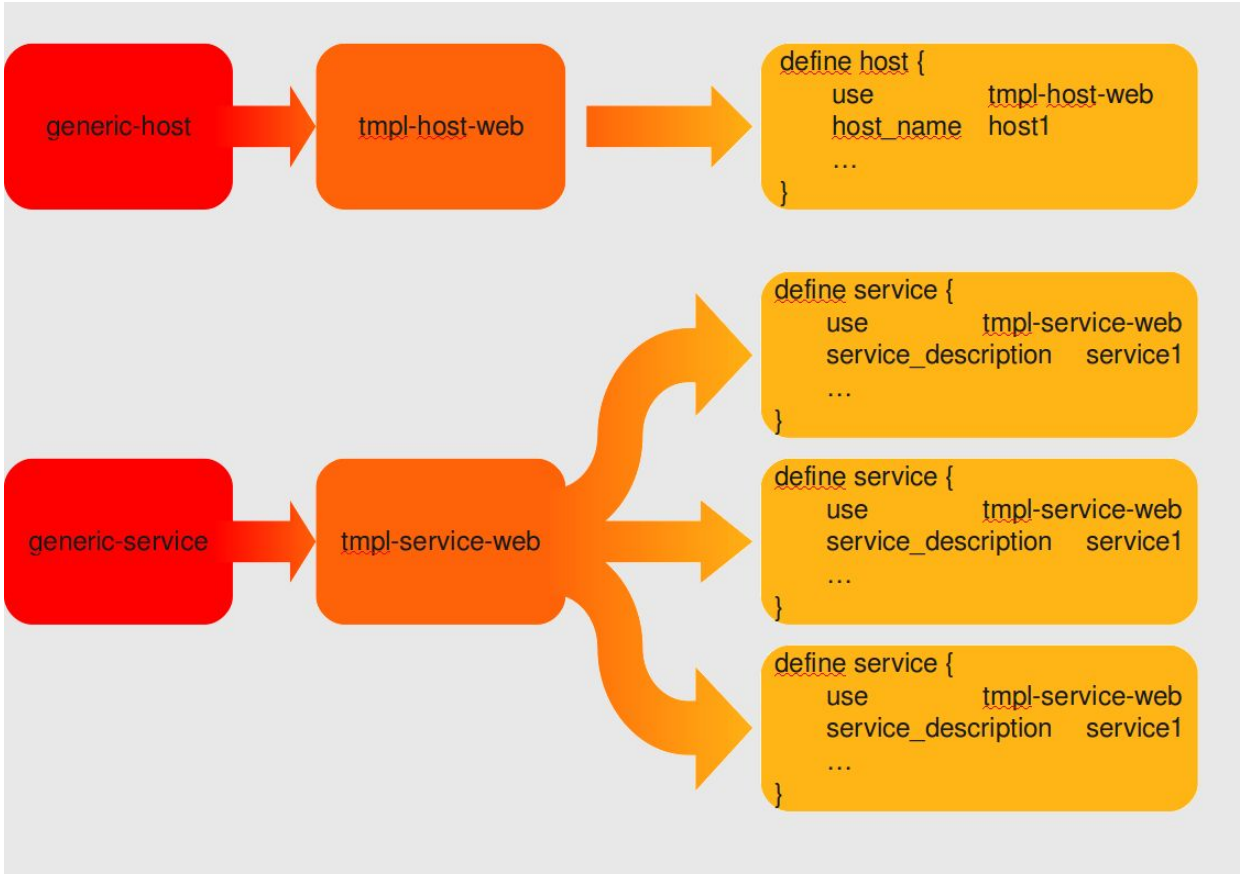
La supervision.

Nagios comprend le système de Template.

Un template permet de transmettre un ensemble de variable commune aux fichiers de configuration. Cela évite de redéfinir des informations pour chaque machine ou service.

Il est possible de fonctionner sans cette fonctionnalité. Mais le but d’un administrateur étant dans faire le moins ce serait dommage de s’en passer.

La supervision.



La supervision.

Exemple de template :

```
define host{
        name                    generic-host
        initial_state           o
        active_checks_enabled   1
        passive_checks_enabled  1
        notifications_enabled   1
        event_handler_enabled   0
        check_command           check-host-alive
        flap_detection_enabled  1
        failure_prediction_enabled  1
        failure_prediction_options  d,u,r
        process_perf_data       1
        check_freshness         0
        obsess_over_host        0
        check_period            24x7
        check_interval          0
        retry_interval          1
        stalking_options        u,d
        max_check_attempts      10
        retain_status_information   1
        retain_nonstatus_information    1
        notification_period     24x7
        first_notification_delay    0
        contact_groups          admins
        notification_options    d,u,r
        notification_interval   0
        register                0
        notes                   generic-host
        }
```

```
define service{
        name                    generic-service
        active_checks_enabled   1
        passive_checks_enabled  1
        initial_state           o
        parallelize_check       1
        obsess_over_service     0
        check_freshness         0
        notifications_enabled   1
        event_handler_enabled   0
        flap_detection_enabled  1
        failure_prediction_enabled  1
        failure_prediction_options  w,c,u
        process_perf_data       1
        retain_status_information   1
        retain_nonstatus_information    1
        is_volatile             0
        check_period            24x7
        flap_detection_options  o,u,c,w
        max_check_attempts      5
        normal_check_interval   5
        retry_check_interval    2
        contact_groups          admins
        notification_options    w,u,c,r
        notification_interval   0
        notification_period     24x7
        first_notification_delay    0
        notes                   generic-service
        register                0
        }
```

La supervision.

```
define host{
        use                     generic-host
        name                    tmpl-host-web
        check_command
check-host-alive
        check_period            24x7
        max_check_attempts      5
        notification_period     24x7
        contact_groups          support
        notification_interval   60
        register                0
        notes                   generic-host
        }
```

```
define service{
        use                     generic-service
        name                    tmpl-service-web
        check_period            24x7
        max_check_attempts      5
        normal_check_interval   15
        retry_check_interval    2
        contact_groups          support
        notification_interval   60
        notification_period     24x7
        register                0
        }
```

Nous pouvons déclarer un template pour les serveurs web. Celui-ci va sera appelé par nos hôtes.

La supervision.

```
define host{
        use                     tmpl-host-web
        host_name               Rainette
        alias                   Serveur Web Rainette
        address                 xx.xx.xx.xx
        contact_groups          support,admins
        }
```

```
define service{
        use                     tmpl-service-web
        host_name               Rainette
        service_description     Reponse
interface Web Nagios
        check_command
check_http!"http://xx.xx.xx.xx/nagios"
        }
```

Et enfin nous déclarons nos hôtes et le service associé.

La supervision.

```
define host{
        use                     tmpl-host-web
        host_name               Rainette
        alias                   Serveur Web Rainette
        address                 xx.xx.xx.xx
        contact_groups          support,admins
        }
```

```
define service{
        use                     tmpl-service-web
        host_name               Rainette
        service_description     Reponse
interface Web Nagios
        check_command
check_http!"http://xx.xx.xx.xx/nagios"
        }
```

Et enfin nous déclarons nos hôtes et le service associé.

La supervision.

Nous trouvons un dernier système de gestion des templates via les pivots.

Le pivot correspondra à un fichier de configuration central qui renvoie les hosts vers les templates associés.

Concrètement :

Nous avons X serveurs web et nous désirons leurs définir des vérifications de base. Pour ne pas avoir à intervenir sur chaque hosts , nous allons déployer un hostgroup qui comprendra la liste de nos hosts.

La supervision.

On crée un ou plusieurs serveurs sur le modèle suivant :

```
define host {
        use                     generic-host
        host_name               <serveurname>
        alias                   <serveurnamealias>
        address                 xx.xx.xx.xx
        contact_groups          support
        }
```

Puis le fichier hostgroups.cfg

```
define hostgroup {
        hostgroup_name          Serveur_web
        alias                   Groupe des Serveurs WEB
        members                 <serveurname>
```

La supervision.

Il nous reste à ajouter des services spécifiques (qui ne sont donc pas générés par le template de base générique service)

Nous allons ajouter un fichier de configuration comme :

service-acces-http.cfg

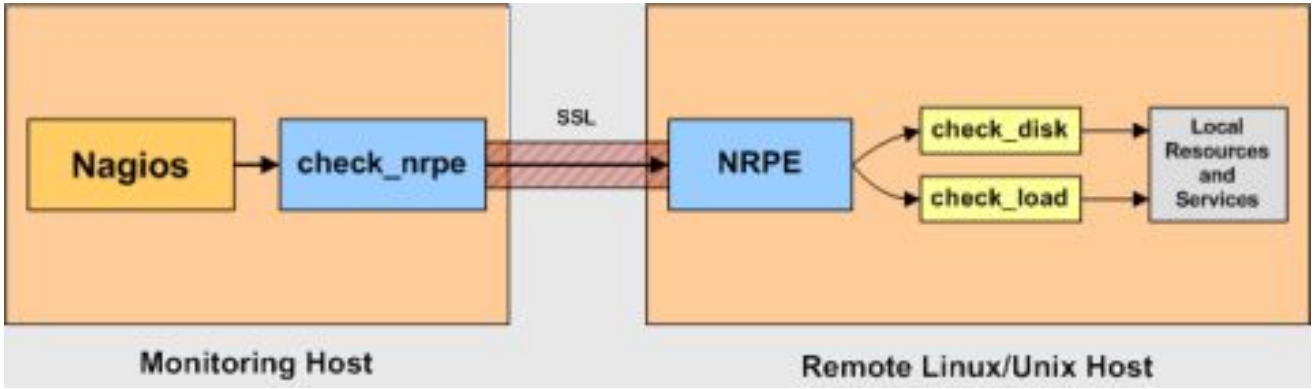
```
# Definition du service de controle d'url Web
define service{
        use                     generic-service
        hostgroup_name          Serveur_web
        service_description     Reponse HTTP
        check_command           check_http
        }
```

La supervision.

Mise en place du service NRPE

Le plugin NRPE permet de lancer des check sur une machine distante.

Le serveur exécute le plugin NRPE avec en argument le check à réaliser en local sur la machine distante.



La supervision.

Mise en place du client nrpe sur le serveur nagios

On installe le client :
apt-get install nagios-nrpe-plugin

Le fichier **/etc/nagios-plugins/check_nrpe.cfg** contient deux définitions de la commande nrpe :

check_nrpe et check_nrpe_1arg

La supervision.

Mise en place du serveur nrpe sur la serveur distant

Il nous faut modifier quelques lignes dans le fichier /etc/nagios/nrpe.cfg

allowed_hosts = Ip sur serveur Nagios

On relance le service :

/etc/init.d/nagios-nrpe-server start

La supervision.

Pour activer des plugins sur le serveur distant, il faut ajouter la commande dans le fichier de configuration nrpe.

Si on désire ajouter une commande pour vérifier l'espace disque :

```
command[check_sda1]=/usr/lib/nagios/plugins/check_disk -w 20 -c 10 -p /dev/sda1
```

Le nom de la commande sera **check_sda1** et sera utilisé avec **check_nrpe_1arg**

La supervision.

Sur le serveur nagios nous pouvons maintenant configurer un check nrpe :

```
# Vérification disque
define service{
use generic-service
host_name hera
service_description disk check
check_command check_nrpe_1arg!check_sda1
}
```