

Administration GNU/Linux (Aspects avancés)





GNU/Linux Operating System

STOCKAGE ORGANISATION DES DONNEES

HELHa

Haute École
Louvain en Hainaut



Organisation des données

• Evolution

Le file system Ext2

C'est l'ancien standard Linux.

Il est basé sur la structure SYSTEM V d'un file system Unix.

Le file system Ext3

C'est une évolution de l'Ext2.

La gestion d'un journal a été ajoutée.

Le file system Ext4

C'est l'actuel standard des systèmes Linux (RHEL6).

C'est un système de transition.

Le file system Xfs

C'est l'actuel standard des systèmes Linux (RHEL7).

Il offre plus de fonctionnalités que l'Ext4.

Le file system Btrfs

Futur standard Linux (?) développé par Oracle (licence GPL).

En développement from 'scratch' depuis 2007.

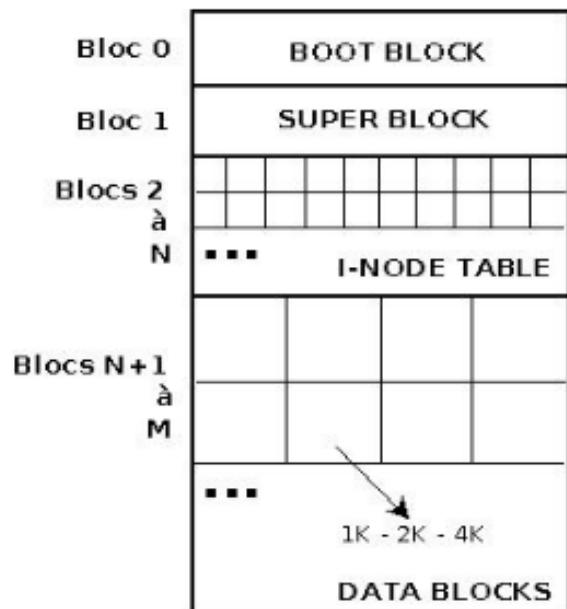
Il est à noter que, officiellement, ce fs est toujours en bêta.



Organisation des données

- **Le FS Ext2**

Structure



Il est vide si l'amorce du bootloader (GRUB) est installé dans le MBR ou si le FS est non bootable.
Il contient l'amorce du bootloader du FS bootable si le bootloader n'est pas installé dans le MBR.

Il est présent en Ram et est recopié périodiquement sur disque par la commande sync ou à l'arrêt du système. Il contient notamment:

- la taille des blocs
- nombre total d'i-nodes et de blocs
- le nombre et la liste des blocs libres
- le nombre et la liste des i-nodes libres
- le nombre de blocs réservés à root
- le nom du file system
- la date de la dernière vérification du fs
- temps maximal entre 2 vérifications
- dates du dernier montage et de la dernière écriture
- état du fs

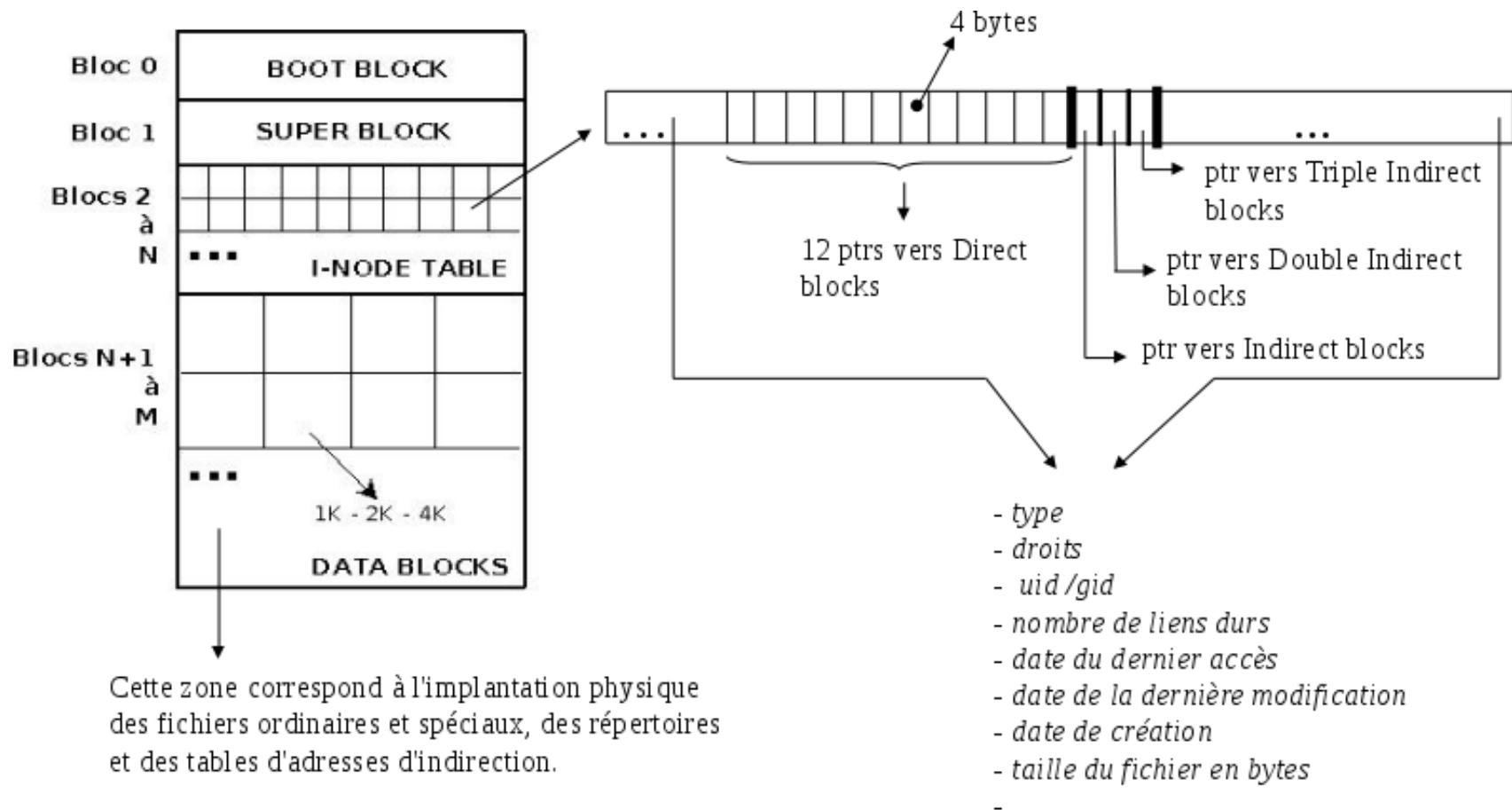
Il existe dans le fs en plusieurs exemplaires...

Blocs de données
Blocs d'indirection
Blocs libres
Blocs défectueux

Un i-node est identifié par un numéro unique qui correspond à sa position dans la table.
Il y a autant de i-nodes que de fichiers dans le FS.
La taille de la table des i-nodes est fixée au moment de la création du FS en fonction de sa taille globale.



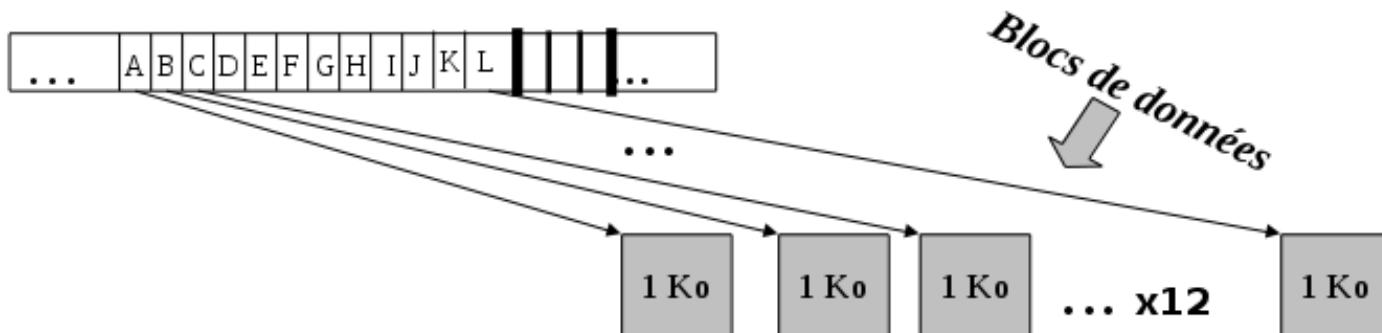
Organisation des données



Organisation des données

- Le FS Ext2 (suite)

Capacité d'adressage



Capacité maximale d'adressage

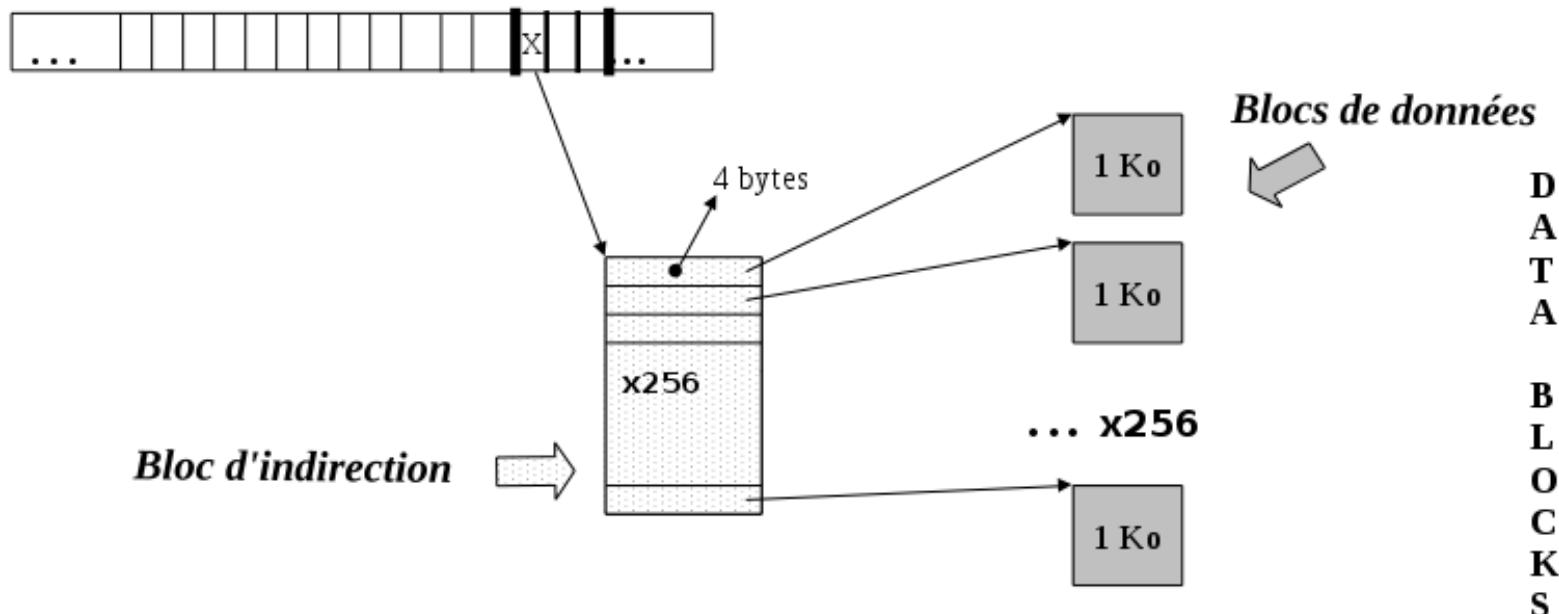
$$12 * 1 \text{ Ko} = \mathbf{12 \text{ Ko}}$$



Organisation des données

- Le FS Ext2 (suite)

Capacité d'adressage



Capacité maximale d'adressage

Capacité précédente = 12 Ko

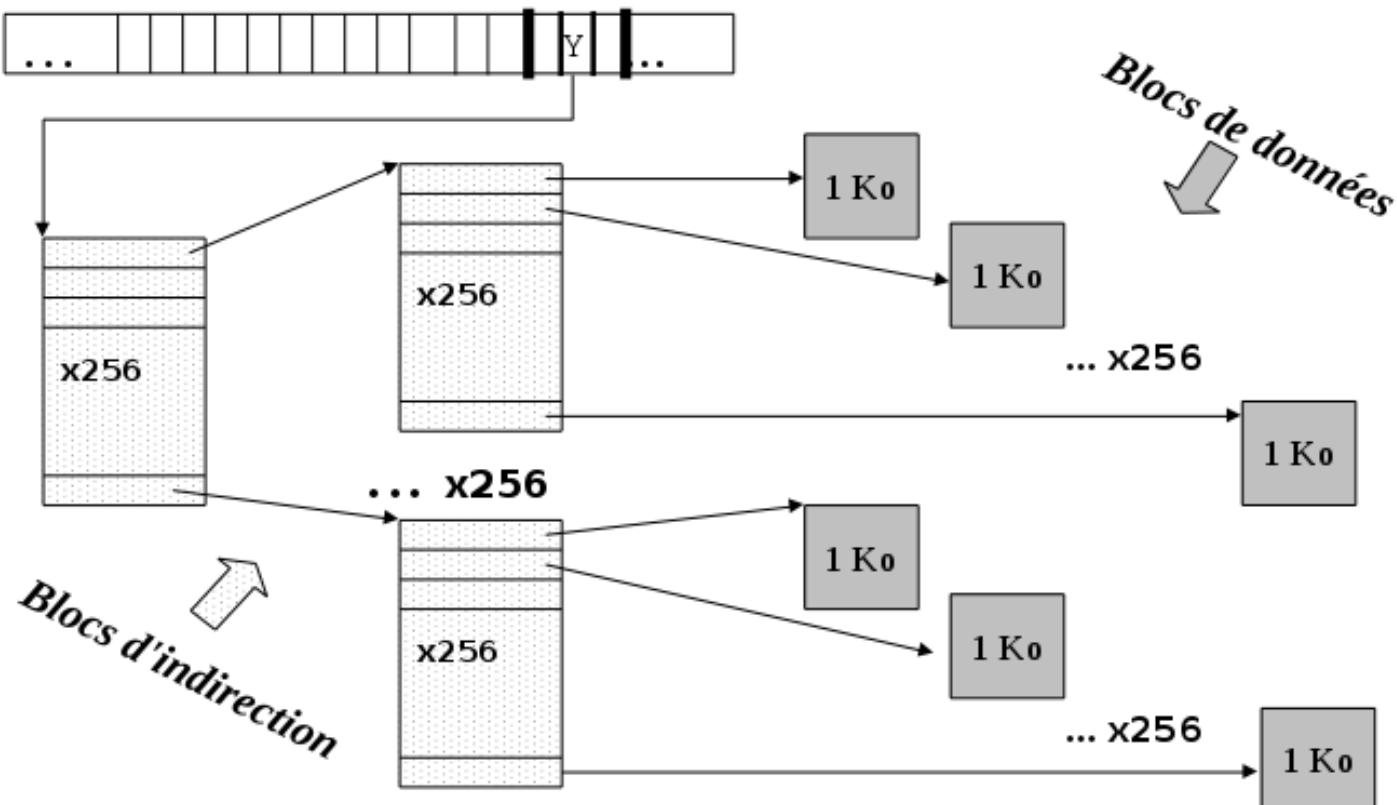
Indirect Blocks = $256 * 1 \text{ Ko} = 256 \text{ Ko}$

Total= 268 Ko



Organisation des données

- Le FS Ext2 (suite)



Capacité d'adressage



Capacité maximale d'adressage

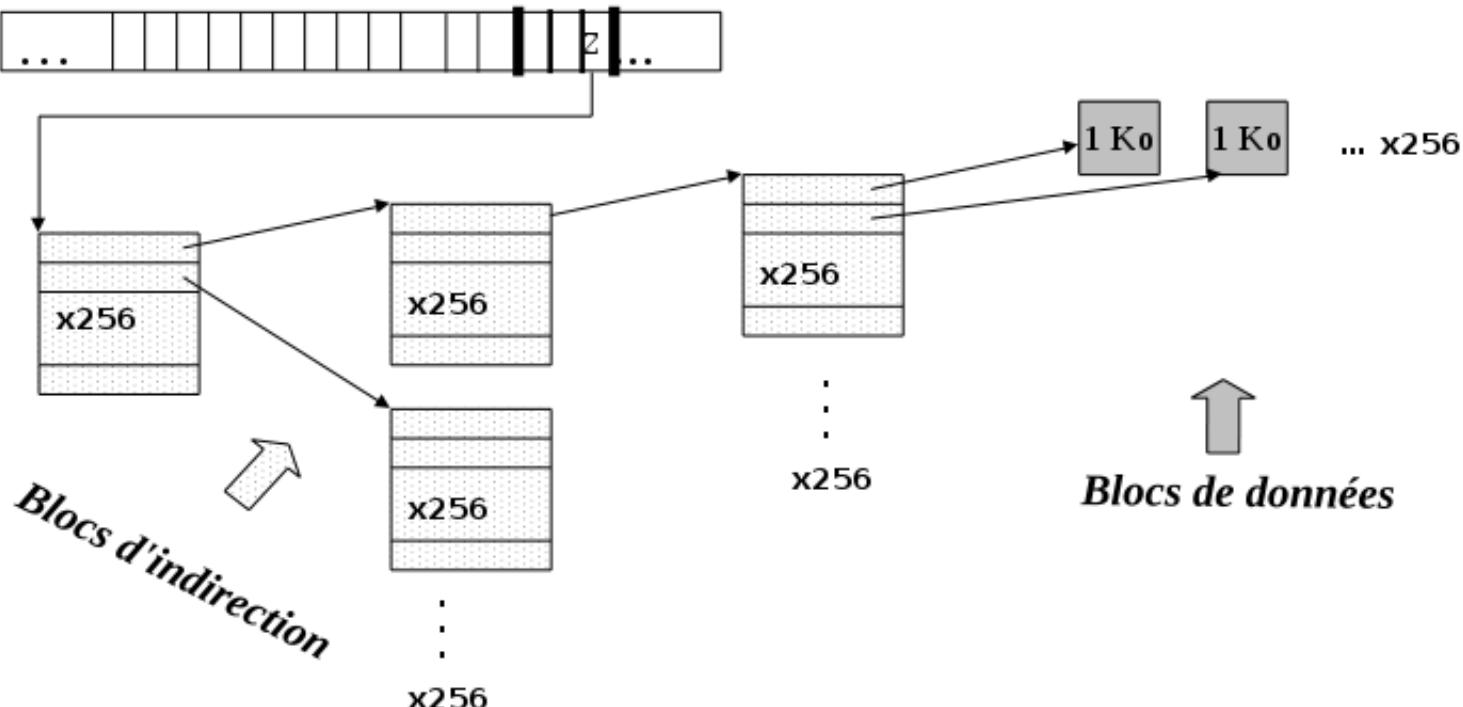
Capacité précédente = 268 Ko + DIBlocks = $(256)^2 * 1 \text{ Ko} = 65536 \text{ Ko} \Rightarrow \text{Total} = 65804 \text{ Ko}$

Gouwy Jean-Louis

Organisation des données

- Le FS Ext2 (suite)

Capacité d'adressage



D
A
T
A

B
L
O
C
K
S

Capacité maximale d'adressage

Capacité précédente = 65804 Ko

Triple Indirect Blocks = $(256)^3 * 1 \text{ Ko} = 16777216 \text{ Ko} \Rightarrow \text{Total} = 16843020 \text{ Ko (16 Go)}$



Organisation des données

- **Le FS Ext2 (suite)**

Capacité d'adressage

EXERCICE Soient des blocs de 1Ko

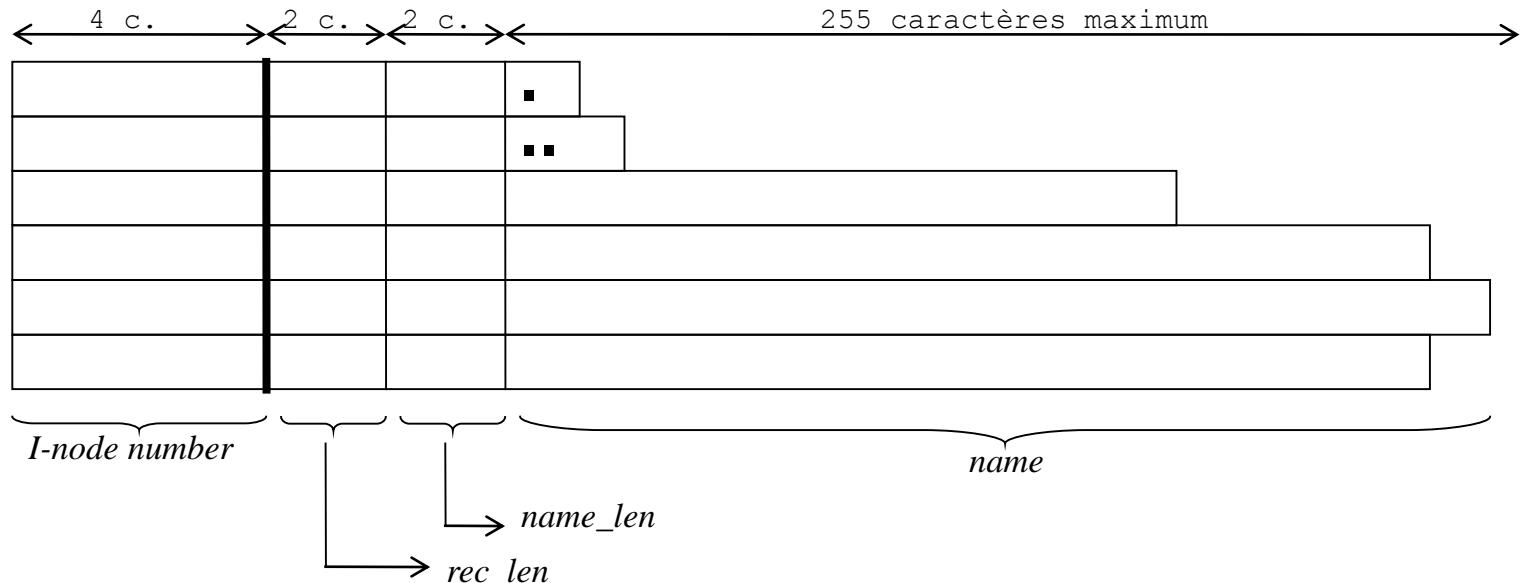
On demande de calculer le nombre de blocs total (données + indirection) monopolisés par un fichier dont la taille réelle s'élève à 10.456.373 caractères.



Organisation des données

- Le FS Ext2 (suite)

Structure d'un répertoire



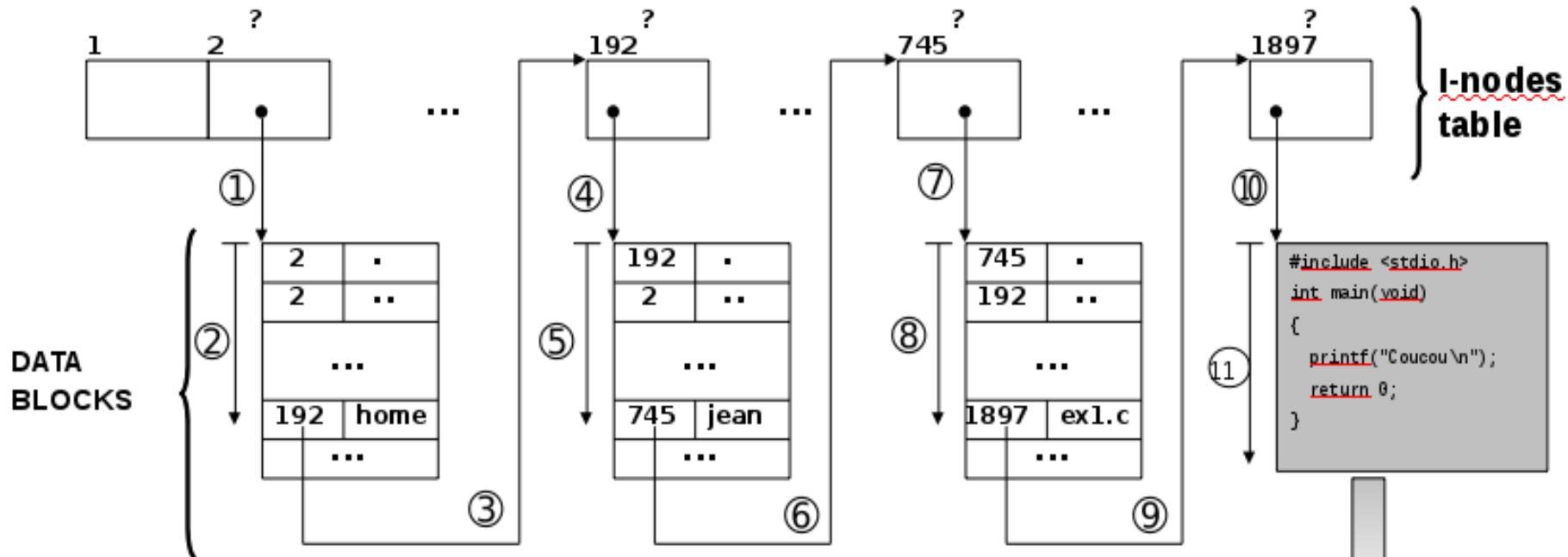
Remarques:

- Le i-node n°2 référence toujours le répertoire racine .
- Le i-node n°1 référence les blocs défectueux.



Organisation des données

- Le FS Ext2 (suite)



Soit la commande:

\$ cat /home/jean/ex1.c

Avec ? : Contrôle sur le type et l'accès



Organisation des données

• Le FS Ext2 (suite)

La table des i-nodes et le super-bloc sont recopiés en Ram lors du montage du fs et sont recopiés périodiquement (~1x/10 sec.) sur le support via la commande **sync** (ou lors de la fermeture du système).

Taille d'un bloc par défaut sur disque dur: 4 Ko

Il est non-journalisé

- ⌚ Risque de perte d'intégrité si coupure de courant ou extinction à chaud.
fsck au rebootage (très lent si taille importante)

En savoir plus:

<http://www-igm.univ-mlv.fr/~dr/NCS/node14.html>

<http://fr.wikipedia.org/wiki/Ext2>

<http://www.csie.ntu.edu.tw/~r95031/LinuxFileSystem.ppt>

Autres file systems non-journalisés:

http://fr.wikipedia.org/wiki/Système_de_fichiers



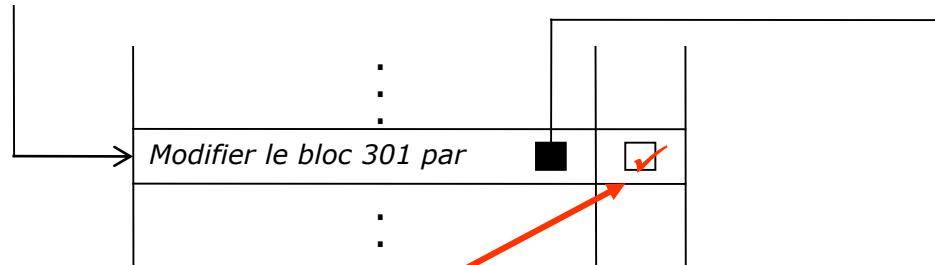
Organisation des données

- **Le FS Ext3**

- Pour remédier aux faiblesses du fs Ext2.
- Principe: fs non-journalisé Ext2 + gestion d'un fichier journal

Exemple: Soit la modification du bloc n°301 d'un fichier:

- ➊ Création d'une entrée dans le journal



- ➋ Modification effective du bloc 301 sur disque dur.
- ➌ Masquage de l'entrée (la modification a été réalisée)



Organisation des données

• Le FS Ext3 (Suite)

⌚ Ecriture en double des données de modification

- une fois dans le journal
- une fois dans le bloc correspondant

... mais le temps de mise à jour est minimisé en utilisant le temps libre du système pour parcourir le journal et ainsi procéder aux modifications.

☺ Si rupture d 'alimentation:

- Pendant l'écriture dans le journal:

Seule l'entrée dans le journal est incorrecte, le fichier reste intact.

Il n'est simplement pas mis à jour et risque donc pas d'être perdu.

- Pendant la mise à jour du fichier:

Au rebootage, le système repère les entrées journal non réalisées et réalise alors les mises à jour qui s'imposent. Il n 'y a donc plus de fsck global à réaliser ...



Organisation des données

- **Le FS Ext3 (suite)**

En savoir plus:

<http://www.linux-france.org/article/sys/ext3fs/ext3.html>
<http://fr.wikipedia.org/wiki/Ext3>
<http://www.csie.ntu.edu.tw/~r95031/LinuxFileSystem.ppt>

Autres file systems journalisés:

http://fr.wikipedia.org/wiki/Système_de_fichiers



Organisation des données

- **Le FS Ext4**

En remplacement des 'indirects blocks', il prend en charge la notion de zone étendue (extent).

Un 'extent' est une zone du disque contigüe éventuellement de très grande capacité (Go).

L'utilisation d'extent réduit la fragmentation et améliore les performances lors de l'utilisation de gros fichiers.

En savoir plus:

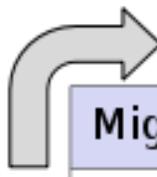
<http://linuxfr.org/news/ext3-est-mort-vive-ext4>

http://www.usenix.org/event/lsf07/tech/cao_m.pdf



Organisation des données

- Compatibilités



Migration	Vers Ext2	Vers Ext3	Vers Ext4
de Ext2	-	<input checked="" type="checkbox"/> (1)	<input checked="" type="checkbox"/> (1) + (2)
de Ext3	<input checked="" type="checkbox"/> (3)	-	<input checked="" type="checkbox"/> (2)
de Ext4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	-

(1) # tune2fs -j /dev/sda5 → on crée le journal

(2) # tune2fs -O extents,uninit_bg,dir_index /dev/sda5
e2fsck /dev/sda5

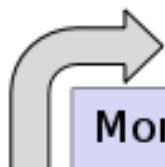
→ sda5 est devenue une partition ext4

(3) # tune2fs -O ^has_journal /dev/sda6 → on enlève l'indic. 'journal'
e2fsck -y /dev/sda6



Organisation des données

- Compatibilités (suite)



Montage	Vers Ext2	Vers Ext3	Vers Ext4
de Ext2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
de Ext3	<input checked="" type="checkbox"/> (1)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> (2)
de Ext4	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

(1) # mount -t ext2 /dev/sda5 /mnt → sda5 (au départ en ext3)
Le journal n'est pas activé.

(2) # mount -t ext4 /dev/sda5 /mnt → sda5 (au départ en ext3)



Organisation des données

- **Manipulations**

Commandes	Ext2	Ext3	Ext4	Description
mke2fs	x	x	x	Crée un FS ext2/3/4
e2fsck	x	x	x	Vérifie ou répare un FS ext2/3/4
dumpe2fs	x	x	x	Affiche des infos sur un FS ext2/3/4
resize2fs	x	x	x	Retaille un FS ext2/3/4 (à froid ou à chaud)
e2label	x	x	x	Affiche ou modifie le label d'un FS ext2/3/4
tune2fs	x	x	x	Modifie les paramètres d'un FS ext2/3/4
dump, restore	x	x	x	Sauvegarde, restaure un FS ext2/3/4
debugfs	x	x	x	Dépanne un FS ext2/3/4
e2image			x	Sauvegarde les métadonnées dans un fichier. Ce type de fichier peut être utilisé pour réparer un FS.
e2freefrag			x	Affiche la fragmentation de la place libre
e2undo			x	Rejoue le journal qui n'a pas été accompli



Organisation des données

- **Le FS Xfs**

Xfs est un système de fichiers développé Silicon graphic, Inc.

Comme Ext4, il est également basé sur les "extents" mais apporte des fonctionnalités supplémentaires.

Fonctionnalités supplémentaires

- Journalisation de métadonnées → récupération après incident plus rapide
- Défragmentation et élargissement possibles alors qu'il est monté et actif.
- Meilleures performances lors des E/S.
- ...



Organisation des données

• Le FS Btrfs

Btrfs (souvent prononcé "ButterFS") est un système de fichiers développé par Oracle.

Au contraire d'ext4 qui est une évolution des systèmes de fichiers ext2/3, btrfs se veut conçu différemment, et apporte certaines fonctionnalités inédites.

Comme Ext4, il est également basé sur les "extents".

Fonctionnalités principales

- Meilleure gestion de l'espace occupé par les petits fichiers.
- Possibilité de créer des snapshots et des sous-volumes.
- Sommes de contrôle des données et des méta-données.
- Compression possible et à la volée des données lors de leur écriture.
- Sauvegarde incrémentale intégrée au système de fichiers.
<https://www.aplu.fr/v2/post/2013/08/02/Les-différents-types-de-sauvegarde>
- Défragmentation à chaud.



Organisation des données

- **Le FS Btrfs (suite)**

En savoir plus:

<http://linuxfr.org/news/btrfs-le-syst%C3%A8me-de-fichiers-du-futur>
<http://doc.ubuntu-fr.org/btrfs>
http://btrfs.wiki.kernel.org/index.php/Btrfs_design
https://btrfs.wiki.kernel.org/index.php/Main_Page



Organisation des données

- Comparatif

Type	Fichier (max)	Fs (max)	Journal	Droits d'accès	Notes
ext2 (Linux)	2 To	16 To	Non	Oui	Utile pour les petits fichiers sur de petits fs. Utile pour des fs montés en read only.
ext3 (Linux)	2 To	16 To	Oui	Oui	Ext2 + journalisation.
ext4 (Linux)	16 To	1 Eo	Oui	Oui	Ext3 + "extents" qui remplacent les blocs d'indirection. Meilleures performances pour les fichiers volumineux. Solution intermédiaire.
xfs (Silicon graphics, inc)	8 Eo	16 Eo	Oui	Oui	Ext4 plus robuste et plus performant que l'ext4.
Btrfs (Oracle Linux)	16 Eo	16 Eo	Oui	Oui	Actuellement en version beta. Très haute capacité de stockage. Se veut concurrent du 'zfs'. Sous licence GPL afin de pouvoir être intégré au noyau Linux. Probablement le futur standard Linux.



Organisation des données

- Comparatif (suite)

Type	Fichier (max)	Fs (max)	Journal	Droits d'accès	Notes
fat (Microsoft) File Alloc. Table	2 Go	2 Go	Non	Non	De moins en moins utilisé sauf pour de petits fs (ex. disquette).
fat32 (Microsoft)	4 Go	8 To	Non	Non	<p>Evolution de la fat depuis W2k et Xp.</p> <p>Microsoft a bloqué volontairement la taille d'un fs à 32Go max lors de son formatage.</p> <p>Une Fat32 peut être formatée avec une taille de 8To max sous Linux.</p> <p>Une FAT32 supérieure à 32Go peut être lue sous Windows.</p>
ntfs (Microsoft) New Techn. FS	16 To	256 To	Oui	Oui	<p>Très peu documenté.</p> <p>L'écriture depuis Linux sur ce système de fichiers est stable à l'aide du pilote ntfs-3g.</p>



Organisation des données

- Comparatif (suite)

Type	Fichier (max)	Fs (max)	Journal	Droits d'accès	Notes
hfs plus (Apple/MacOS)	8 Eo	8 Eo	Oui	Oui	
zfs (Oracle) Z File System	16 Eo	16 Eo	Oui	Oui	Très haute capacité de stockage. Très léger. Il gère les instantanés. Il intègre la gestion des volumes (LVM..) . Sous licence CDDL afin de ne pas pouvoir être intégré au noyau Linux.

Plus d'info: http://en.wikipedia.org/wiki/Comparison_of_file_systems



Organisation des données

Atelier

Gérer les FS Ext2/3/4



GNU/Linux Operating System

LE NOYAU

HELHa

Haute École
Louvain en Hainaut



Plan

- **LE NOYAU**

Présentation

Rôle du noyau / Noyau standard d'installation / Noyau monolithique / Rappel sur les modules

Atelier: Noyau et modules

La compilation

Pourquoi compiler le noyau ? / Pourquoi compiler un module ? / Les outils nécessaires / Introduction à make / Les grandes étapes de la compilation

Atelier: Compilation du noyau



Le noyau

- **Présentation**

Rôles du noyau

- **Gestion des processus**

Allocations équitables du (ou des) processeur(s) aux différents processus,
gestion des priorités, gestion des états des processus ...

- **Gestion de la mémoire**

Allocation mémoire, paging, swapping.

- **Gestion des disques, des périphériques et des FS**

Droits d'accès, Virtual File System (VFS) ...



Le noyau

• Présentation (suite)

Noyau standard d'installation

- Il s'appelle `vmlinuz-`uname -r`` et est installé dans `/boot` par l'outil d'installation de la distribution.
- Il est bien souvent modulaire. La plupart des pilotes (dont ceux nécessaires au démarrage) sont souvent compilés dans des modules séparés dont les binaires d'extension `.ko` sont enregistrés dans le dossier `/lib/modules/`uname -r``.
- Il est souvent lancé par le chargeur de démarrage (Grub).

```
# cat /boot/grub/grub.conf
title CentOS (2.6.32-431.el6.x86_64)
root (hd0,0)
```

*(hd0,0) → 1^{ère} partition du 1^{er} disque du Bios rencontré par Grub
(nomenclature Grub rien n'est encore chargé ni monté !)*

→ ici cette partition correspond à `/dev/sda1` et contient toute l'arborescence `/boot`.



Le noyau

- Présentation (suite)

Noyau standard d'installation

```
kernel /vmlinuz-2.6.32-431.el6.x86_64 ro root=...
```

Grub chargera le noyau 'vmlinuz-2.6.32-431.el6.x86_64' se trouvant à la racine (/) de cette partition en lui passant les arguments ro root=nom_du_fs_principal ...

```
initrd /initramfs-2.6.32-431.el6.x86_64.img
```

Grub montera ce FS virtuel initial. Celui-ci contient les pilotes nécessaires afin que le FS principal puisse être monté par le noyau au bootage du système (contrôleur IDE ou SATA..., type du FS ext2 ou 3 ou 4 ...).

Les pilotes stockés dans cette image dépendent de l'architecture du système. Elle sera donc générée lors de l'installation du système.

```
# ls -l /boot
```

<pre>...</pre>	<p style="text-align: right;"><i>Date d'installation du linux</i></p> <pre>22 nov. 04:40 config-2.6.32-431.el6.x86_64 ... 14 mai 14:56 initramfs-2.6.32-431.el6.x86_64.img ... 22 nov. 04:40 vmlinuz-2.6.32-431.el6.x86_64</pre>
----------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



Date de la compilation du noyau

Gouwy Jean-Louis

Le noyau

• Présentation (suite)

Noyau monolithique

- Il est d'un seul bloc et ne contient donc aucun module externe (tous les pilotes sont dans le noyau).
- Pour des noyaux légers.
- Il est bien souvent utilisé dans les systèmes embarqués (faible besoin en pilotes).

Rem: Ce sont les pilotes du bootloader qui permettront le chargement du noyau en RAM du système embarqué.

- Les pilotes (en nombre limité) sont compilés au sein du noyau.
- Pas de fichier `initramfs`.
(Plus d'info sur l'`initramfs`: <https://wiki.gentoo.org/wiki/Initramfs/Guide/fr>)



Le noyau

• Présentation (suite)

Les modules

- La plupart des pilotes (et quelques sous-systèmes) peuvent être compilés sous forme de module (LKM: Loadable Kernel Module).
- L'ajout d'un périphérique se traduit donc par l'installation de son module binaire natif ou recompilé au départ de son code source.
 - pas besoin de recompiler le noyau
 - mise à jour aisée du noyau par le remplacement, la suppression ou l'ajout de modules
 - chargement en mémoire uniquement des modules utilisés
- Un module ne pourra être chargé en mémoire que si toutes ses dépendances le sont également.
- Un module pourra être déchargé de la mémoire si celui-ci n'est pas une dépendance d'un ou de plusieurs autres modules.
- Le noyau ne peut charger un module que si ce dernier a été compilé pour la version exacte de ce votre noyau.



Le noyau

- **Présentation (suite)**

Les modules

Les commandes

`lsmod` – `insmod` – `modprobe` – `depmod` – `rmmmod` – `modinfo`

Les fichiers

`/etc/modules.conf` → fichier de configuration des modules (noyau 2.4)

`/etc/modprobe.conf` → fichier de configuration des modules (noyau 2.6)

`/etc/modprobe.d/*.conf` → fichiers de configuration des modules
(noyau 2.6 récent)

`/lib/modules/$ (uname -r)` → arborescence de stockage des modules (.ko)

`/lib/modules/$ (uname -r) /modules.dep` → fichier des dépendances entre
modules.

`/etc/sysconfig/modules` → dossier contenant des fichiers permettant de
forcer le chargement de modules au démarrage.
Gouwy Jean-Louis



Le noyau

Atelier

Noyau et modules



Le noyau

• La compilation

Pourquoi compiler le noyau ?

- Ajuster au mieux ses fonctionnalités au matériel et aux besoins (suppression / ajout de diverses fonctionnalités)
- Réglage de ses performances:
Modules dynamiques ou statiques (intégrés au sein du noyau)
- Essais des fonctionnalités d'un nouveau noyau.

Remarques:

- Pour une version commerciale de Linux, la recompilation pourrait être interdite sous peine de perdre le support.
- Certains fabricants ne livrent leur pilote que sous forme binaire et uniquement pour telle ou telle version d'un noyau
 - ces pilotes deviendraient inutilisables sous un nouveau noyau recompilé.



Le noyau

- **La compilation (suite)**

Pourquoi compiler un module ?

Installer un pilote à partir de son code source et ce, quelle que soit la version du noyau.

Les outils nécessaires

- Le compilateur C (souvent `gcc`).
- Les bibliothèques de développement C standard (`glibc`).
- Les bibliothèques spécifiques à la manipulation des menus de configuration du noyau: `ncurses` (pour `menuconfig`), `qt3` (pour `xconfig`)
- L'utilitaire `make`



Le noyau

- **La compilation (suite)**

Les outils nécessaires (suite)

- Le package `module-init-tools` qui contient des programmes permettant la gestion des modules et leur (dé)chargement automatique.
- `mkinitrd` (du package `dracut`) qui crée un fichier `initramfs` associé à une version du noyau.



Le noyau

• La compilation (suite)

Introduction à make

- C'est un puissant langage spécialisé dans la gestion de projets.
(<http://www.gnu.org/software/make/manual/make.html>
<http://ql.developpez.com/tutoriel/outil/makefile/>)
- Il obéit aux règles définies dans un fichier appelé `Makefile` devant se trouver à la racine du dossier dans lequel `make` doit opérer.
- Structure d'un `Makefile`: ensemble de règles
 - . Les dépendances sont analysées.
Si une dépendance est la cible d'une autre règle, cette règle est à son tour évaluée.
 - . Lorsque l'ensemble des dépendances est analysé et si la cible ne correspond pas à un fichier existant ou si un fichier dépendance est plus récent que la règle, les différentes commandes sont exécutées.

cible1: dépendance (s)
 commande (s)
cible2: dépendance (s)
 commande (s)

...



Le noyau

- **La compilation (suite)**

Introduction à make (suite)

- Exemple: Soit un projet C se composant de 2 fichiers sources:

s1.c

```
#include <stdio.h>
int main(void)
{
    f1();
    puts("Bye");
    return 0;
}
```

f1.c

```
#include <stdio.h>
void f1(void)
{
    puts("Hello");
}
```



Le noyau

- **La compilation (suite)**

Introduction à make (suite)

- Création d'un Makefile qui permettra de générer un binaire monprog à partir de la compilation de ces 2 fichiers.

Makefile

```
projet: s1.o f1.o  
        gcc -o monprog s1.o f1.o
```

```
s1.o: s1.c  
        gcc -c s1.c
```

```
f1.o: f1.c  
        gcc -c f1.c
```

Les objets s1.o et f1.o doivent être générés pour générer la cible projet via la commande gcc -o monprog s1.o f1.o.

Le fichier s1.c doit exister pour générer la cible s1.o via la commandes gcc -c s1.c

Le fichier f1.c doit exister pour générer la cible f1.o via la commandes gcc -c f1.c

Pas de link.



Le noyau

- **La compilation (suite)**

Introduction à make (suite)

- Appel à make et exécution

```
$ make projet      → on renseigne la cible 'projet' du makefile
gcc -c s1.c
gcc -c f1.c
gcc -o monprog s1.o f1.o
```

Exécution des commandes du makefile pour générer cette cible et donc l'exécutable du projet.

```
$ ls -l
-rw-rw-r--. 1 joe joe   50  8 juil. 18:55 f1.c
-rw-rw-r--. 1 joe joe 1480  8 juil. 19:06 f1.o
-rw-rw-r--. 1 joe joe    94  8 juil. 18:41 Makefile
-rwxrwxr-x. 1 joe joe 6542  8 juil. 19:06 monprog
-rw-rw-r--. 1 joe joe    69  8 juil. 18:55 s1.c
-rw-rw-r--. 1 joe joe 1552  8 juil. 19:06 s1.o
```

```
$ ./monprog
Hello
Bye
```



Le noyau

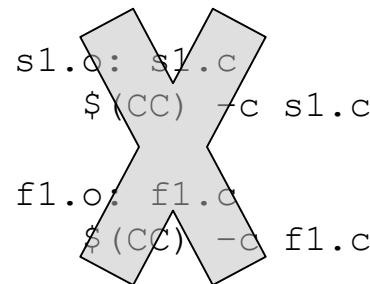
• La compilation (suite)

Introduction à make (suite)

- make peut manipuler un grand nombre de variables

Makefile (nouvelle écriture)

```
CC = gcc      → il suffit de modifier une seule ligne pour changer de compilateur
OBJS = s1.o f1.o
projet: $(OBJS)
        $(CC) -o monprog $(OBJS)
```



Plus besoin d'indiquer ces règles. Elles sont implicites.
Lorsqu'on ne les indique pas clairement, make ajoutera lui-même les règles suivantes:

```
s1.o: s1.c
```

```
$(CC) $(CFLAGS) -c s1.c
```

```
f1.o: f1.c
```

```
$(CC) $(CFLAGS) -c f1.c
```

Ici, la variable CFLAGS (options passées au compilateur) est vide.



Le noyau

• La compilation (suite)

Introduction à make (suite)

- Ecrire proprement un Makefile

```
CC = gcc
OBJS = s1.o f1.o
...
DESTDIR = ./projet
```

all: projet → permettre le classique 'make all' pour obtenir l'exécutable.
'make' pointe toujours vers la 1^{ère} règle du makefile et, ici, est donc équivalente à 'make all'.

```
projet: $(OBJS)
        $(CC) -o monprog $(OBJS)
```

install: all → permettre l'installation du programme
mkdir -p \$(DESTDIR)
cp monprog \$(DESTDIR)

clean: → effacer les fichiers inutiles
rm -f \$(OBJS)

distclean: clean → effacer tout sauf les sources de l'auteur
rm -f monprog

Gouwy Jean-Louis



Le noyau

- **La compilation (suite)**

Introduction à make (suite)

- **Avantages**

- . Lors de la recompilation complète du projet, `make` ne recompile que les sources dont la date est plus récente que leur fichier objet.
- . Exécutions (compilations) parallèles possibles sous des architectures multi-processeurs.

Exemple: `make -j2 projet` → 2 compilations indépendantes seront lancées en même temps...

- . C'est un outil universel qui peut être utilisé dans le cadre d'une gestion de projet autre que la compilation multi-sources.

Son usage s'avère facile partout où il s'agit d'obtenir quelque chose à partir de fichiers existants

Exemple: encoder des CD en MP3, créer des images avec POV...



Le noyau

- **La compilation (suite)**

Les grandes étapes de la compilation

1. Récupérer et installer les sources du noyau à partir du site officiel (kernel.org) ou à partir du site de votre distribution.
2. Patcher (éventuellement) les sources.
Les patches correspondent à des modifications des sources.
Une nouvelle fonctionnalité du noyau se présente sous forme de source ou sous forme d'un patch.
3. Configurer le noyau.
C'est l'étape la plus importante et la plus longue.
Concrètement, on va faire le choix des modules qui composent le noyau.
On choisit aussi si l'inclusion est statique ou dynamique...
4. Compiler le noyau.
5. Créer (éventuellement) un fichier `initramfs`.
6. Installer les modules.
7. Configurer le bootloader pour permettre le démarrage du noyau.
8. Redémarrer le système en utilisant le nouveau noyau.
9. Tests



Le noyau

Atelier

Compilation du noyau et des modules





GNU/Linux Operating System

DEMARRAGE DU SYSTEME

HELHa

Haute École
Louvain en Hainaut



Plan

- **DEMARRAGE DU SYSTEME**

Séquence de démarrage

Vue générale / Processeur vers BIOS / BIOS vers MBR / MBR vers Noyau / Noyau vers Init

Les bootloaders LILO et GRUB

LILO / GRUB Legacy / GRUB2 + **Atelier:** GRUB Legacy

Les systèmes d'initialisation

Le Pid n°1 / sysVinit / Upstart / Systemd + **Atelier:** L'init sous CentOS 6.x

Arrêt du système

Introduction / sysVinit / Upstart / Systemd

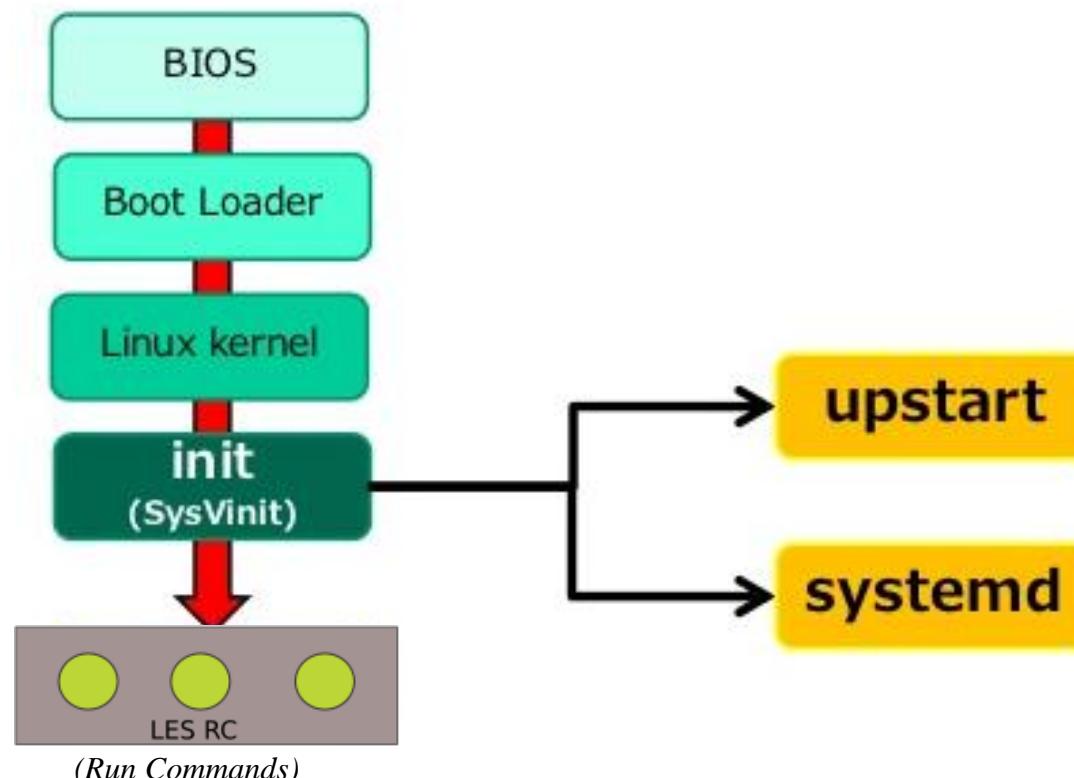
Autres ressources



Démarrage du système

- Séquence de démarrage

Vue générale

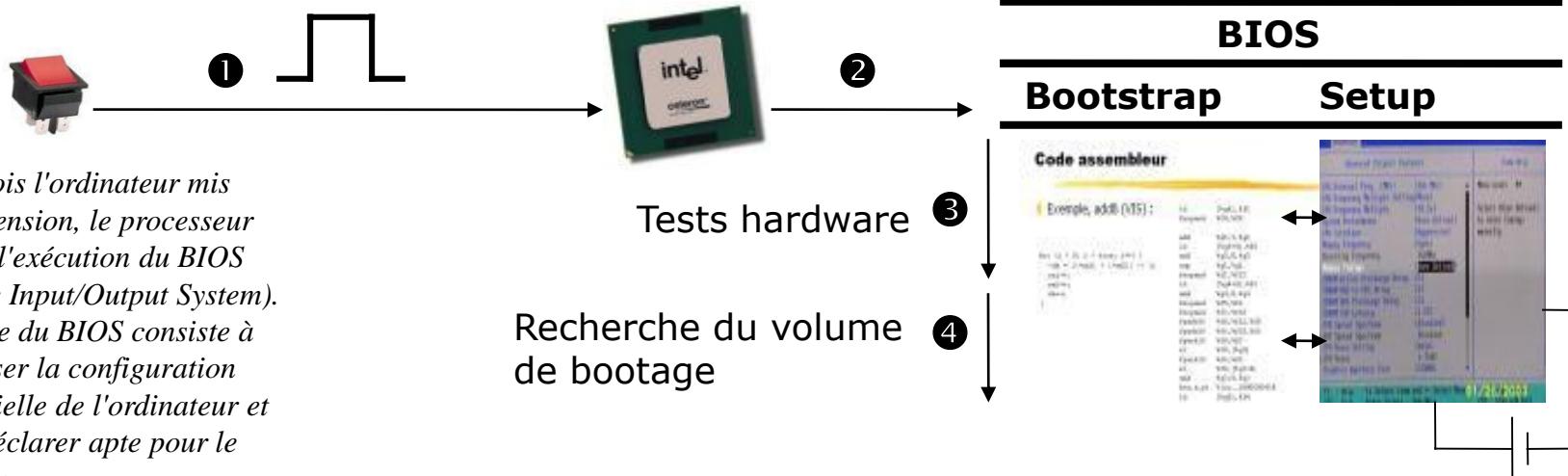


Démarrage du système

- Séquence de démarrage

Processeur vers Bios (Legacy)

Linux, à l'instar de tout système d'exploitation, effectue une série d'opérations lors de son chargement. L'étude de celles-ci vous autorisera à les configurer et de mieux appréhender cette étape importante.



Une fois l'ordinateur mis sous tension, le processeur lance l'exécution du BIOS (Basic Input/Output System). Le rôle du BIOS consiste à analyser la configuration matérielle de l'ordinateur et à la déclarer apte pour le service.

Une fois le système initialisé correctement, le BIOS déclenche une interruption précédant la recherche d'un système d'exploitation . Celle-ci s'établit selon un ordre prédéfini au sein du "Setup" (configuration du BIOS). En règle générale, il commence sa recherche sur le floppy ou le CD-ROM puis sur le disque dur.



Démarrage du système

• Séquence de démarrage

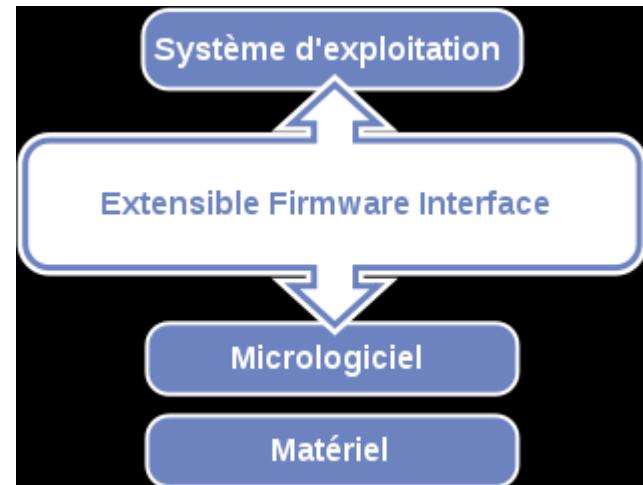
Processeur vers Bios

Remarque

Actuellement, les Bios dits 'UEFI' tentent de s'imposer sur les nouvelles cartes mères pour remplacer les Bios traditionnels dits 'Legacy'.

Fonctionnalités:

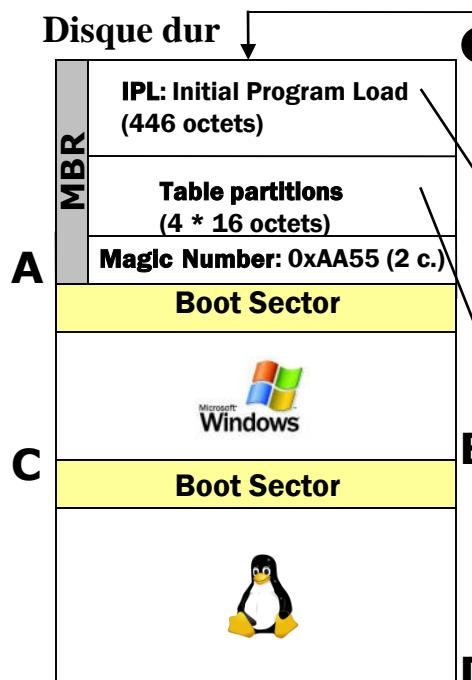
- Remplacement du MBR par la GPT →
 - . Bootage du système sur disque > à 2,2 To
 - . Jusqu'à 128 partitions primaires.
- Il fournit un shell proche de celui du Linux.
- Il peut être utilisé pour lancer d'autres applications UEFI, ce qui inclut des UEFI bootloaders.
- Secure Boot: lancement de systèmes d'exploitation reconnus (fonctionnalité dénoncée comme anormale par la communauté du libre)
- Configuration en mode 'Legacy' possible.



Démarrage du système

- Séquence de démarrage

Bios Legacy vers MBR



① *Le BIOS lit le 1^{er} secteur (MBR:Master Boot Record) du périph. amorçable selon l'ordre défini.*
Lorsqu'il trouve le magic number^() (0xAA55), il charge alors en 0X7C00 l'IPL et l'exécute.*

() C'est une signature et pas un checksum !*

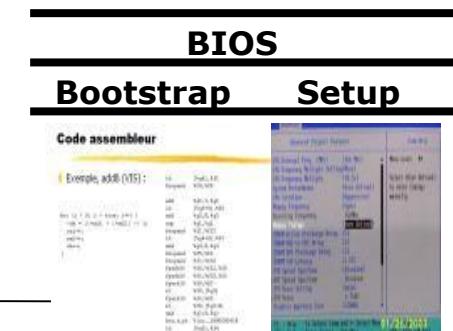
② *Programme d'amorce destiné soit à:*
-Rechercher l'amorce du bootloader (ex. le 'stage1' de Grub)
-S'il n'y a pas de bootloader, alors le Bios recherche la partition active dans la table des partitions et charge son boot sector (programme d'amorce de l'OS – bien souvent son bootloader !).



Win A B

Linux C D ✓

→ *Partition active*
*(Max. 4 partitions (4*16 octets))*



Démarrage du système

• Séquence de démarrage

Bios vers MBR

Rôle du chargeur

Permettre la cohabitation de plusieurs OS, ou au moins charger un OS. C'est lui qui doit charger et exécuter le noyau Linux, ou le programme d'amorçage de Windows.

Principaux chargeurs

Dans le monde GNU/Linux, on n'utilise presque plus LILO, au profit de GRUB

LILO . Le chargeur historique de Linux.

- . Nécessite de 'compiler' le MBR à partir de son fichier de conf. (lilo.conf) lors de l'installation de nouveaux paramètres.

GRUB . Shell paramétrable en dynamique lors du boot.

- . Evite la 'recompilation' du MBR lors des modifications d'options.
- . Le plus en vogue.

SYSLINUX . Chargeur pour démarrage en réseau (PXEboot)ou à partir d'une clé ou d'un CD.



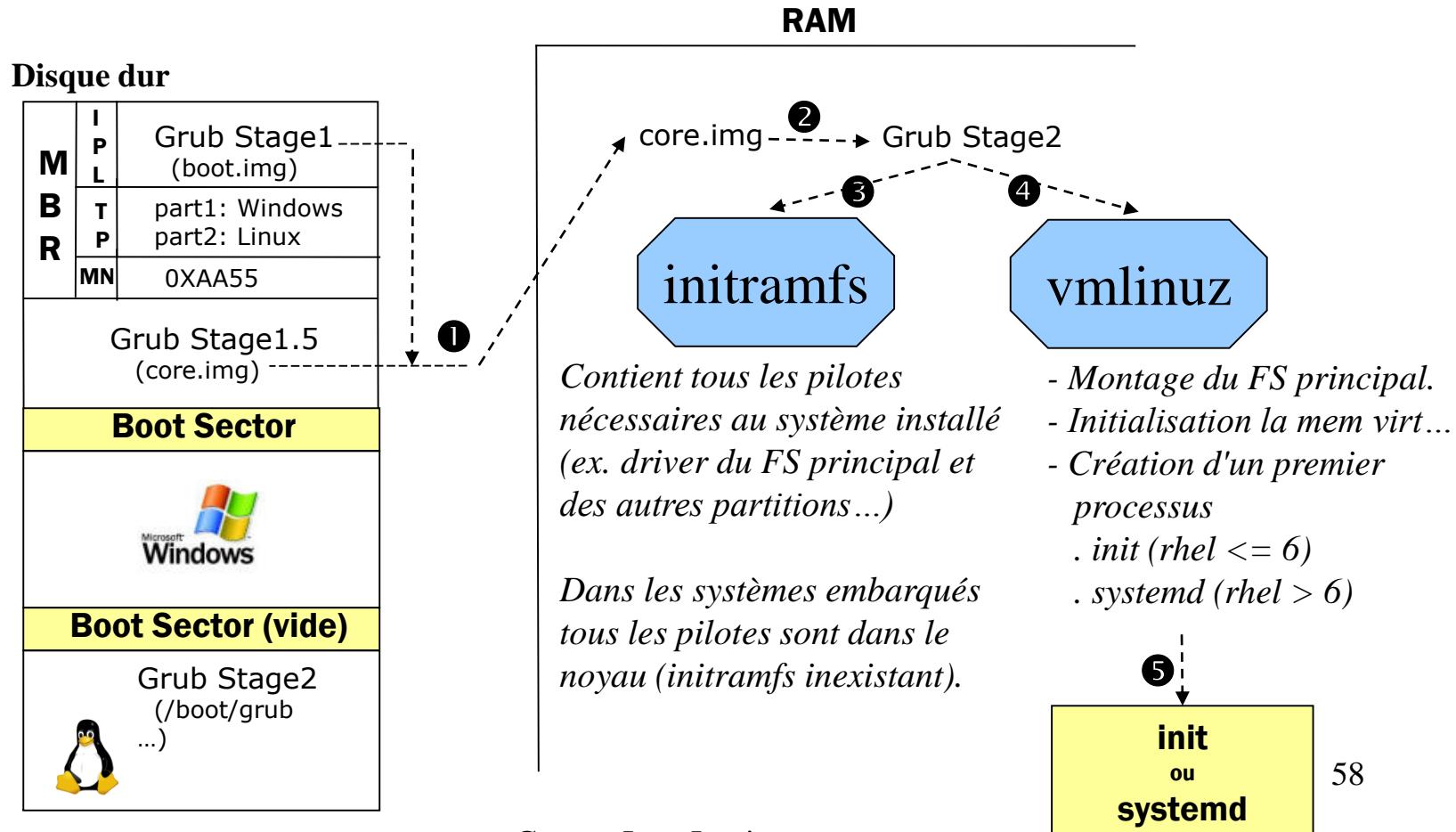
Autres: NTLDLR (Windows), Das U-Boot (universel), RedBoot (systèmes embarqués)...
Gouwy Jean-Louis

Démarrage du système

- Séquence de démarrage

MBR vers Noyau

Cas n°1: L'amorce de Grub est installée dans le MBR

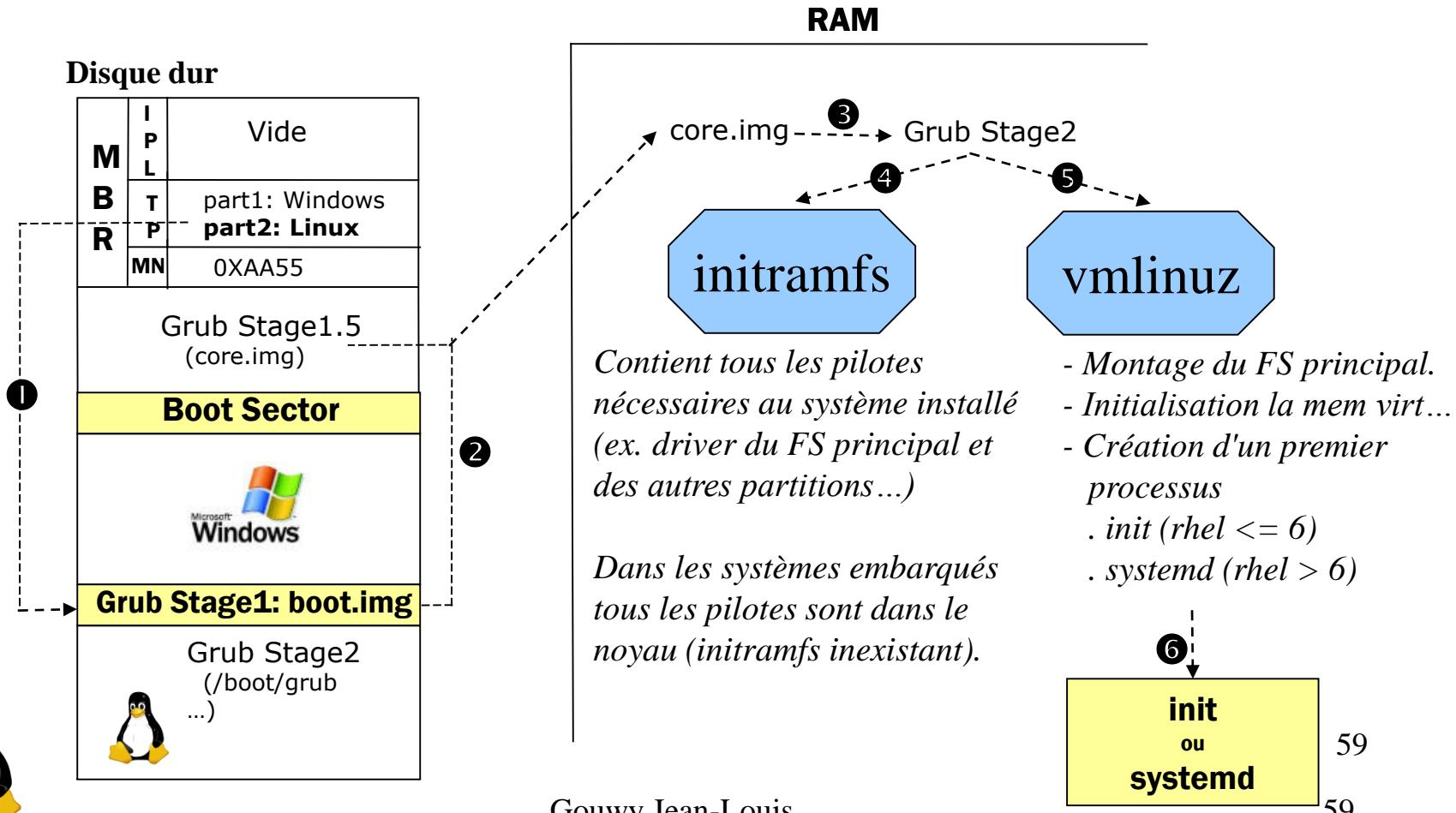


Démarrage du système

- Séquence de démarrage

MBR vers Noyau

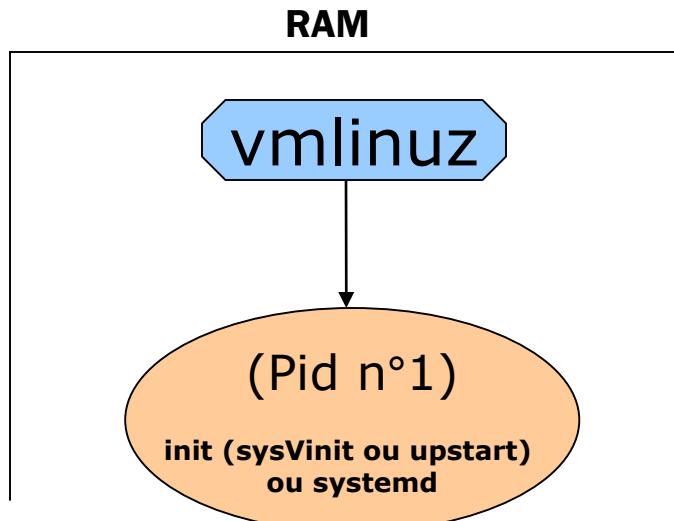
Cas n°2: L'amorce de Grub est installée dans le secteur de démarrage



Démarrage du système

- Séquence de démarrage

Noyau vers Init



Le PID 1 s'avère être le parent de tous les processus du système.

Il va placer la machine dans un runlevel c'est à dire un état dans lequel certains processus existent, et d'autres non. Cela passe par le lancement de différents services .



Démarrage du système

• Les bootloaders LILO et GRUB

LILO

- Documentation : man pages lilo(8), lilo.conf(5)
- Commande `lilo` après chaque modification de configuration
- Fichiers créés (par défaut) : /boot/map, /boot/boot.MMmm
- Fichier de configuration /etc/lilo.conf

```
boot=/dev/hda
install=menu
prompt
default=Linux
image=/boot/vmlinuz-2.6.26
label="Linux"
root=/dev/hda1
append=""
other=/dev/hda3
label="Windows"
```



LILO – Plus d'info: : <http://lilo.alioth.debian.org>

Démarrage du système

• Les bootloaders LILO et GRUB

GRUB Legacy

- RHEL5/6
- Numérotation "universelle" des disques:

Grub identifie les disques selon leur ordre de présentation par le Bios (et pas selon leur type IDE- SCSI – SATA ...).

1 ^{er} disque → hd0	1 ^{ère} part. physique du 1 ^{er} disque	→ (hd0,0)
	2 ^{ème} part. physique du 1 ^{er} disque	→ (hd0,1)
	3 ^{ème} part. physique du 1 ^{er} disque	→ (hd0,2)
	4 ^{ème} part. physique du 1 ^{er} disque	→ (hd0,3)
	part. étendue du 1 ^{er} disque	→ (hd0,n ^o dern. part. phys. +1)

1 ^{ère} part. logique de la part. étendue du 1 ^{er} disque	→ (hd0,5)
2 ^{ème} part. logique de la part. étendue du 1 ^{er} disque	→ (hd0,6)

...

2^{ème} disque → hd1 ...

...

Démarrage du système

- **Les bootloaders LILO et GRUB**

GRUB Legacy

➤ Manipulation simplifiée:

Pas de commande à lancer

Fichier de configuration unique: /boot/grub/menu.lst

```

default=0           → N° d'ordre de l'OS à charger (0=le 1er)  Image de fond de grub → grub est capable de décompresser ...
timeout=5          → Compte à rebours de 5 secondes
splashimage=(hd0,1)/grub/splash.xpm.gz
hiddenmenu          → Le menu sera caché => Esc pour qu'il apparaisse
title CentOS (2.6.29)
root (hd0,1)         → Partition sur laquelle le
kernel /vmlinuz-2.6.29 ro root=/dev/sda3 rhgb quiet   → noyau est stocké
initrd /initramfs-2.6.29.img
Titre affiché       → Si 'kernel panic', ne pas indiqué cet argument
dans le menu        pour voir le debug...
...                 → Chargement du noyau avec ses
                        paramètres de lancement ...
Le FS est monté en  → Initialisation d'un système minimal en Ram avant le
lecture seule (il    chargement du noyau et pouvant charger le système
sera monté en       de fichier principal et les modules obligatoires
lecture/écriture    nécessaires ... Cette image est adressable via le
                     pilote installé dans core.img (stage1.5).
après la vérification
effectuée par fsck).  → Partition sur laquelle
                        le FS principal est
                        stocké. Adressable via
                        le pilote installé dans
                        l'initramfs.
  
```



Démarrage du système

• Les bootloaders LILO et GRUB

GRUB Legacy

- Le chainloading:

Grub lance directement les OS ouverts sans passer par leur secteur de boot, mais pour les OS propriétaires, il active leur secteur de boot.

```
...
title Win7
rootnoverify (hd0,0)
chainloader +1
```

→ *Identification de la partition racine*

└→ *Chargement d'un fichier comme chargeur secondaire.
Il s'agit du 'boot sector' de Windows
(charger +1 bloc à partir de hd0,0).*



Démarrage du système

- **Les bootloaders LILO et GRUB**

GRUB Legacy - Plus d'info: : <http://doc.fedora-fr.org/wiki/Catégorie:GRUB>
https://fr.wikipedia.org/wiki/GNU_GRUB

- Chargement du noyau Grub complexe : 3 stages



*Code d'appel à stage 1_5.
boot.img (446 bytes)*

*Logé dans le MBR (sector 0)
ou dans le bootsector.*



- **core.img (32256 bytes) (Sectors 1-62)**
- *Généré à l'installation de Grub. Il contient le driver du FS de la partition /boot.*
- *Liste des drivers possibles dans /boot/grub: e2fs_stage_1_5, fat_stage_1_5, xfs_stage1_5...*

Aucun driver pour LVM, ext4 et Btrfs → impossible d'installer le système de bootage sur une telle partition.

- *Appel à Stage 2*



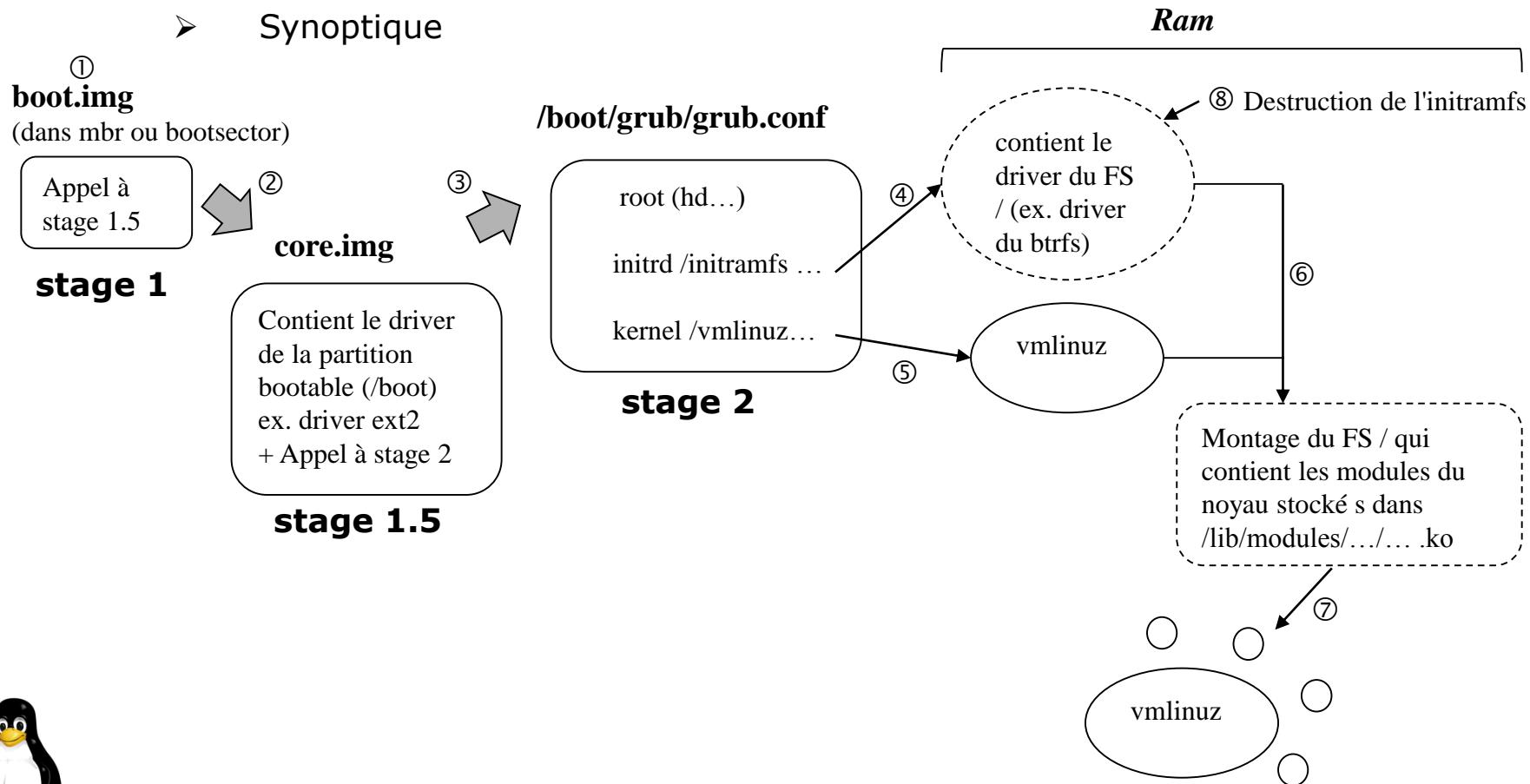
- **/boot/grub**
C'est le noyau de Grub.
- *Lecture du fichier de configuration pour:*
 - afficher le menu
 - offrir un shell
 - ...
- *Puis, montage du FS virtuel initial (initramfs), chargement du noyau et montage du FS principal (driver de celui-ci dans l'initramfs).*



Démarrage du système

- **Les bootloaders LILO et GRUB**

GRUB Legacy



Démarrage du système

• Les bootloaders LILO et GRUB

GRUB 2

- La numérotation a changé:
(hd0,1) = /dev/hda1 (ou /dev/sda1)
Repérage par UUID ou LABEL conseillé.
- Fichiers de configuration:
Effectif : /boot/grub2/grub.cfg (écrit en langage de script)
Reconstruit par grub2-mkconfig
Sources multiples :
/etc/default/grub
/etc/grub.d/*
- Quelques différences vis-à-vis du Grub Legacy

D'autres systèmes de fichiers sont supportés comme Ext4, HFS+,
NTFS, BTRFS ...
Lecture directe sur les périphériques LVM et RAID possible.
Les étapes (stage) 1, 1.5 et 2 sont remplacées par l'utilisation de
fichiers 'images'.



GRUB 2 – Plus d'info: : <http://doc.fedoraproject.org/wiki/Catégorie:GRUB>

Démarrage du système

Atelier

Grub Legacy

(GRand Unified Bootloader - Legacy)



Démarrage du système

• Les systèmes d'initialisation

Le Pid n°1

(Pid n°1)

**init (sysVinit ou upstart)
ou systemd**

- . initialiser les actions de "bas niveau" (montage de systèmes de fichiers, activation du swap, ...),
- . lancer des processus (les services),
- . proposer un moyen de se connecter au système.

SysVinit

Red Hat Enterprise Linux < 6.0

```
# ps 1
1 ? Ss 0:00 init [3]
```

Upstart

Red Hat Enterprise Linux 6

```
# ps 1
1 ? Ss 0:00 /sbin/init
```

Systemd

Red Hat Enterprise Linux 7

```
# ps 1
1 ? Ss 0:00 /usr/lib/systemd/systemd ...
```



Démarrage du système

• Les systèmes d'initialisation

Le Pid n°1

- Le Pid n°1 gère:
 - l'activation de processus lors du démarrage du système
 - différents niveaux de fonctionnement (runlevels) de la machine, dans lesquels on doit ou ne doit pas lancer certains processus ou services.
- Les runlevels standards

Niveau	Rôle
0	Arrêt du système
1	Mode maintenance (single user)
2	Mode multi-utilisateurs de base (incluant TCP/IP)
3	Multi-utilisateurs + RPC (pour NIS, NFS, par ex.)
4	Libre (en principe maintenu comme le 3)
5	Multi-utilisateurs complet (3) + XDM (X-Window)
6	Reboot



runlevel ou # who -r → connaître le runlevel courant

Démarrage du système

- **Les systèmes d'initialisation**

sysVinit

- Point d'entrée: **/etc/inittab**
- Démarrage

```
si:::sysinit:/etc/rc.d/rc.sysinit
```

```
id:3:initdefault:
```

runlevel choisi par défaut s'il n'est pas passé en paramètre au noyau

```
13:3:wait:/etc/rc.d/rc 3
```

+ Appel à **rc.local**

Script de démarrage exécuté (lien symbolique S99local)



Appel à rc.sysinit, qui exécute les tâches de bas niveau: conf. du hostname, du clavier, de l'horloge, de la swap, des quotas, (dés)activation de selinux, montage de /proc ...

Lancé en mode "sysinit" (=wait, mais uniquement lors d'un boot)

Appel à rc, qui lance tous les services concernés par le runlevel choisi - lancé en mode "wait"

*(on attend qu'il rende la main pour continuer)
- utilise des scripts de démarrage installés par les packages fournissant les services*

- . stockés dans /etc/init.d (ou /etc/rc.d/init.d)*
- . des liens symboliques sont créés dans /etc/rcX.d, où X est le runlevel concerné*

Démarrage du système

• Les systèmes d'initialisation

sysVinit

➤ Terminaux

```
1:2345:respawn:/sbin/mingetty tty1
x:5:respawn:/etc/X11/prefdm -nodaemon
```

Lancement de processus monitorés par "init", comme les terminaux de connexion (lancés en mode "respawn": si le processus rend la main, on le relance)

➤ Actions sur événements

```
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
pf::powerfail:/sbin/shutdown -f -h +2 "Arrêt dans 2 minutes"
pr:12345:powerokwait:/sbin/shutdown -c "Arrêt annulé"
```

Quelques déclarations d'actions à exécuter sur évènement (ctrlaltdel, powerfail, powerokwait, ...)

➤ Changement de runlevel

```
telinit <runlevel> ou init <runlevel> (ex. # init 1)
```



Démarrage du système

• Les systèmes d'initialisation

sysVinit

- Exemple de fichier /etc/inittab

```
id:5:initdefault:  
si::sysinit:/etc/rc.d/rc.sysinit  
10:0:wait:/etc/rc.d/rc 0  
11:1:wait:/etc/rc.d/rc 1  
12:2:wait:/etc/rc.d/rc 2  
13:3:wait:/etc/rc.d/rc 3  
14:4:wait:/etc/rc.d/rc 4  
15:5:wait:/etc/rc.d/rc 5  
16:6:wait:/etc/rc.d/rc 6  
ca::ctrlaltdel:/sbin/shutdown -t3 -r now  
pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting Down"  
pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Cancelled"  
1:2345:respawn:/sbin/mingetty --noclear tty1  
2:2345:respawn:/sbin/mingetty tty2  
3:2345:respawn:/sbin/mingetty tty3  
4:2345:respawn:/sbin/mingetty tty4  
5:2345:respawn:/sbin/mingetty tty5  
6:2345:respawn:/sbin/mingetty tty6  
x:45:respawn:/etc/X11/prefdm -nodaemon
```

Si ce champ est vide, alors la commande sera exécutée quelle que soit le niveau de runlevel dans lequel on entre.

Structure d'un record:

identifier:runlevels_list:keyword:command



Démarrage du système

- **Les systèmes d'initialisation**

sysVinit

- Remarque: La commande `mingetty`

```
...
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
...
```

`mingetty` (get TeleTYpe) est un processus permettant de gérer les consoles virtuelles.

Il relit et interprète le contenu de son fichier de configuration `/etc/issue` chaque fois qu'il est (re)lancé...

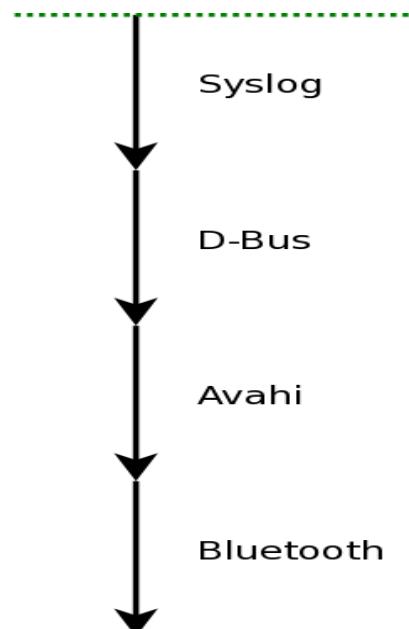


Démarrage du système

• Les systèmes d'initialisation

sysVinit

- Lancement des services



Les services Avahi et Bluetooth nécessitent tous deux le fonctionnement de D-Bus, qui, à son tour, réclame Syslog.

- Les services sont lancés (arrêtés) séquentiellement selon leur ordre d'apparition dans /etc/rc.d/rcx.d
- Système simple hérité des systèmes Unix.
- Temps de chargement relativement long.



Démarrage du système

- **Les systèmes d'initialisation**

sysVinit

- Contrôle des services:

- Arrêt/Démarrage/Redémarrage d'un service

```
# service <service> start|stop|restart
```

- Installation d'un service

```
vi myservice
#!/bin/bash
# chkconfig: 2345 90 60
# description: Mon service perso
# processname: myservice
...

```

```
# cp service /etc/rc.d/init.d
# chmod +x /etc/rc.d/init.d/myservice
# chkconfig --add myservice
```

Runlevels de démarrage + n° de séquence des liens S + n° de séquence des liens K



Démarrage du système

- **Les systèmes d'initialisation**

sysVinit

- Contrôle des services:

- Désinstallation d'un service

```
# chkconfig --del <service>
```

- Activation/Désactivation d'un service lors du passage dans un runlevel donné

```
# chkconfig [--level <levels>] <service> on|off
```

- Liste de tous les services et daemons par runlevel

```
# chkconfig --list
```

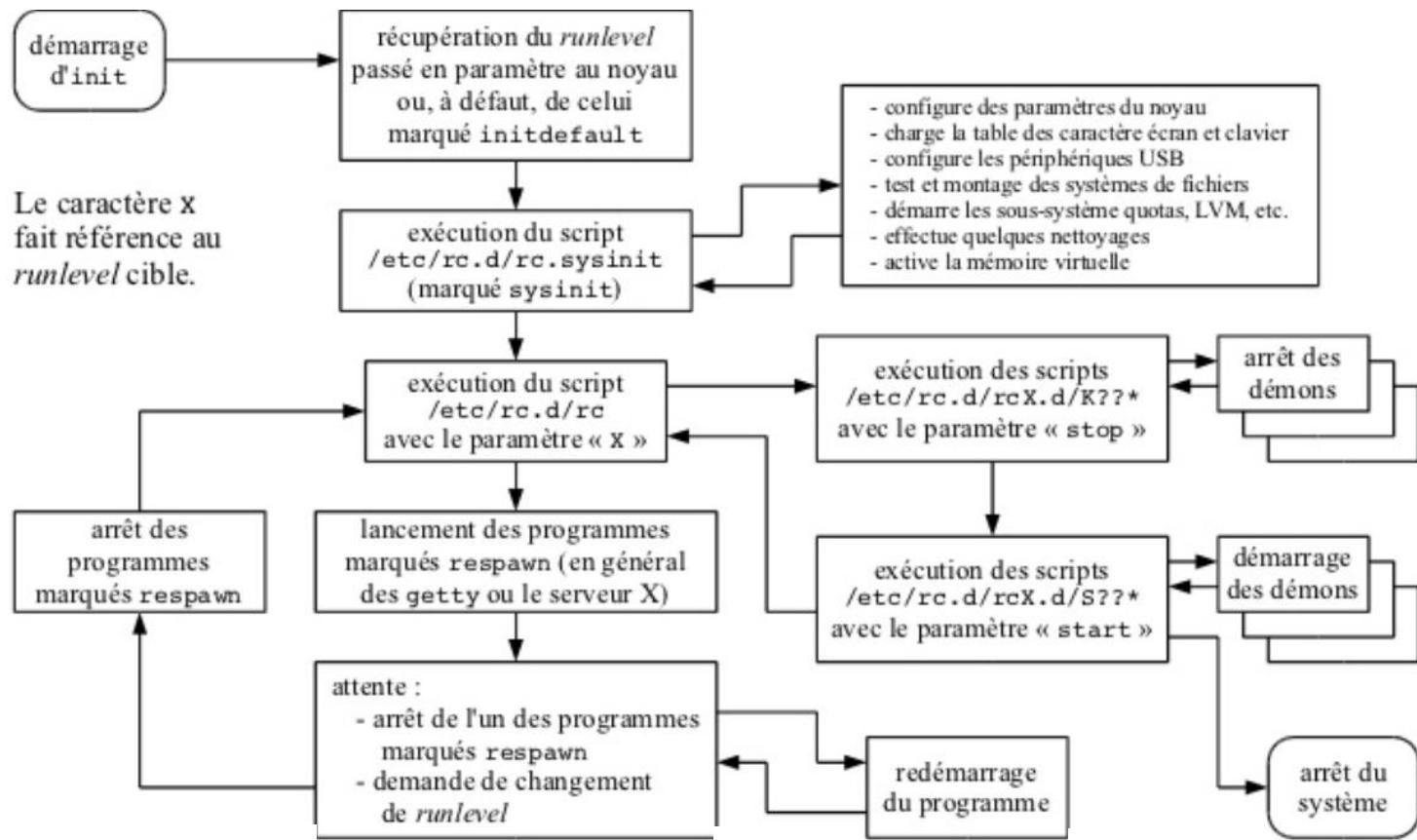


Démarrage du système

• Les systèmes d'initialisation

sysVinit

➤ Vue d'ensemble



Démarrage du système

• **Les systèmes d'initialisation**

Upstart

- Nouvelle approche du programme Init créée par Canonical (Ubuntu).
- Supervise le démarrage et l'arrêt des services (jobs) en utilisant une approche événementielle.
- Caractéristiques:
 - . Un service est démarré ou arrêté suite à la réception d'un événement.
 - . Le démarrage ou l'arrêt d'un service déclenche des événements.
 - . Un événement peut provenir de n'importe quel processus.
 - . Les services peuvent être relancés s'ils meurent.
 - . Le transfert des événements utilise le système D-BUS.
- Les packages concernant les services installent toujours les scripts de démarrage dans `/etc/init.d` et les liens symboliques dans `/etc/rcx.d`.
- **Le système Upstart de RHEL6 maintient donc la compatibilité avec les "runlevels".**



Démarrage du système

• Les systèmes d'initialisation

Upstart

- Point d'entrée: **/etc/init/**

Au démarrage, "init" lit le contenu de /etc/init.

Il y trouve des fichiers définissant des jobs, c'est à dire les lignes de commandes à exécuter et les évènements qui déclenchent leur exécution.

Il existe deux types de job :

- *Une tâche est exécutée une fois et retourne à l'état attente (« waiting ») après son exécution (équivalent aux commandes définies en wait dans le fichier /etc/inittab).*
- *Un service est supervisé et doit être relancé (respawned) s'il s'arrête de manière inattendue (équivalent des commandes définies en respawn dans le fichier /etc/inittab).*



Démarrage du système

• Les systèmes d'initialisation

Upstart

➤ Démarrage

/etc/inittab

→ id:3:initdefault

Uniquement utilisé par Upstart pour le runlevel par défaut.

/etc/init/rcS.conf

→ /etc/rc.d/rc.sysinit
→ exec telinit \$runlevel

Initialisation du système

/etc/init/rc.conf

→ exec /etc/rc.d/rc \$RUNLEVEL

Gestion des runlevels individuels



Démarrage du système

• Les systèmes d'initialisation

Upstart

➤ Terminaux

/etc/init/start-ttys.conf
→ initctl start tty TTY=\$tty

Envoi de l'événement 'start' au job décrit dans tty.conf (avec le n° du terminal en argument). Chaque terminal est lancé en mode 'respawn' (voir /etc/init/tty.conf)

➤ Actions sur événements

/etc/init/control-alt-delete.conf
→ start on control-alt-delete
exec /sbin/shutdown -r now "Control-Alt-Delete pressed"

Lorsque Upstart reçoit l'événement <ctrl><alt>, le lancement de l'arrêt du système est programmé;

➤ Changement de runlevel

telinit <runlevel> ou init <runlevel> (ex. # init 1)



Démarrage du système

• Les systèmes d'initialisation

Upstart

- Exemple (script de démarrage des scripts sysVinit de niv. x sur RHEL6)

/etc/init/rc.conf

```
start on runlevel [0123456]  
  
stop on runlevel [!$RUNLEVEL]  
  
task  
  
export RUNLEVEL  
console output  
exec /etc/rc.d/rc $RUNLEVEL
```

La valeur de x se trouvant dans la variable RUNLEVEL.

Au démarrage, Upstart garnit cette variable du runlevel par défaut indiqué dans /etc/inittab.

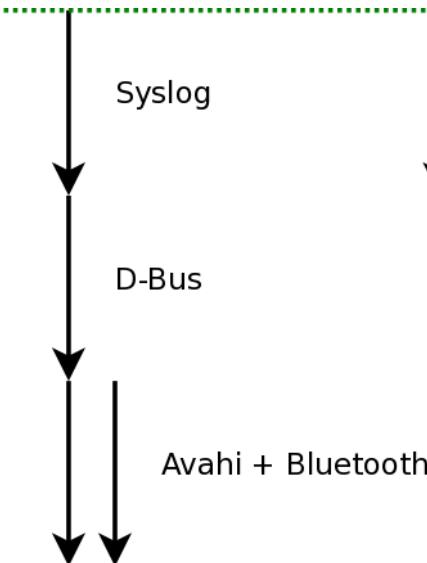


Démarrage du système

• Les systèmes d'initialisation

Upstart

➤ Lancement des services:



Les services Avahi et Bluetooth nécessitent tous deux le fonctionnement de D-Bus, qui, à son tour, réclame Syslog.

- Upstart est orienté "événement" et travaille avec des "jobs".
- Système plus complexe.
- Chaque job, lors de son arrêt ou son démarrage, est susceptible d'arrêter ou de démarrer un autre job.
- Quand un "événement" survient, Upstart démarre en parallèle les jobs en attente sur cet événement.
- Cela peut réduire le temps de chargement (parallélisation possible).



Démarrage du système

• Les systèmes d'initialisation

Upstart

➤ Contrôle des services:

- Arrêt/Démarrage/Redémarrage d'un service

```
# initctl start|stop|restart <job>
```

- Installation d'un service

```
vi <job>.conf  
# cp <job>.conf /etc/init
```

- Désinstallation d'un service

```
# rm -f /etc/init/<job>.conf
```

- Activation/Désactivation d'un service lors du passage dans un runlevel donné:

Ajout de la ligne suivante au job.conf concerné:

```
start on runlevel RUNLEVEL=[<levels>]
```

✓ L'Upstart de RHEL6 reste compatible avec les commandes service et chkconfig du sysVinit ... ☺



Démarrage du système

- **Les systèmes d'initialisation**

Upstart

➤ Plus d'info:

- <http://upstart.ubuntu.com>
- http://upstart.ubuntu.com/cookbook/upstart_cookbook.pdf



Démarrage du système

- **Les systèmes d'initialisation**

Systemd

- Concurrent de Upstart (Ubuntu)
- Embarqué sur Fedora, Mandriva, OpenSuse, Debian et ... RHEL7 (Lennart Poettering, son auteur, travaille chez Red Hat...)
- Les scripts sysVinit sont remplacés par des "unités" de service.
- Le PID n°1 s'appelle "systemd"
- Le composant de base géré par systemd s'appelle une unité (unit).



Démarrage du système

- **Les systèmes d'initialisation**

Systemd

- Point d'entrée:

/lib/systemd/system

- . Contient des fichiers de configuration.
- . Chaque fichier décrit une unité et a un nom de la forme sshd.service, graphical.target, cups.socket, ... suivant son type.

/etc/systemd/system

- . Contient les configurations modifiées par l'administrateur.
- . Les fichiers de ce répertoire sont prééminents sur ceux de /usr/lib/systemd.



Démarrage du système

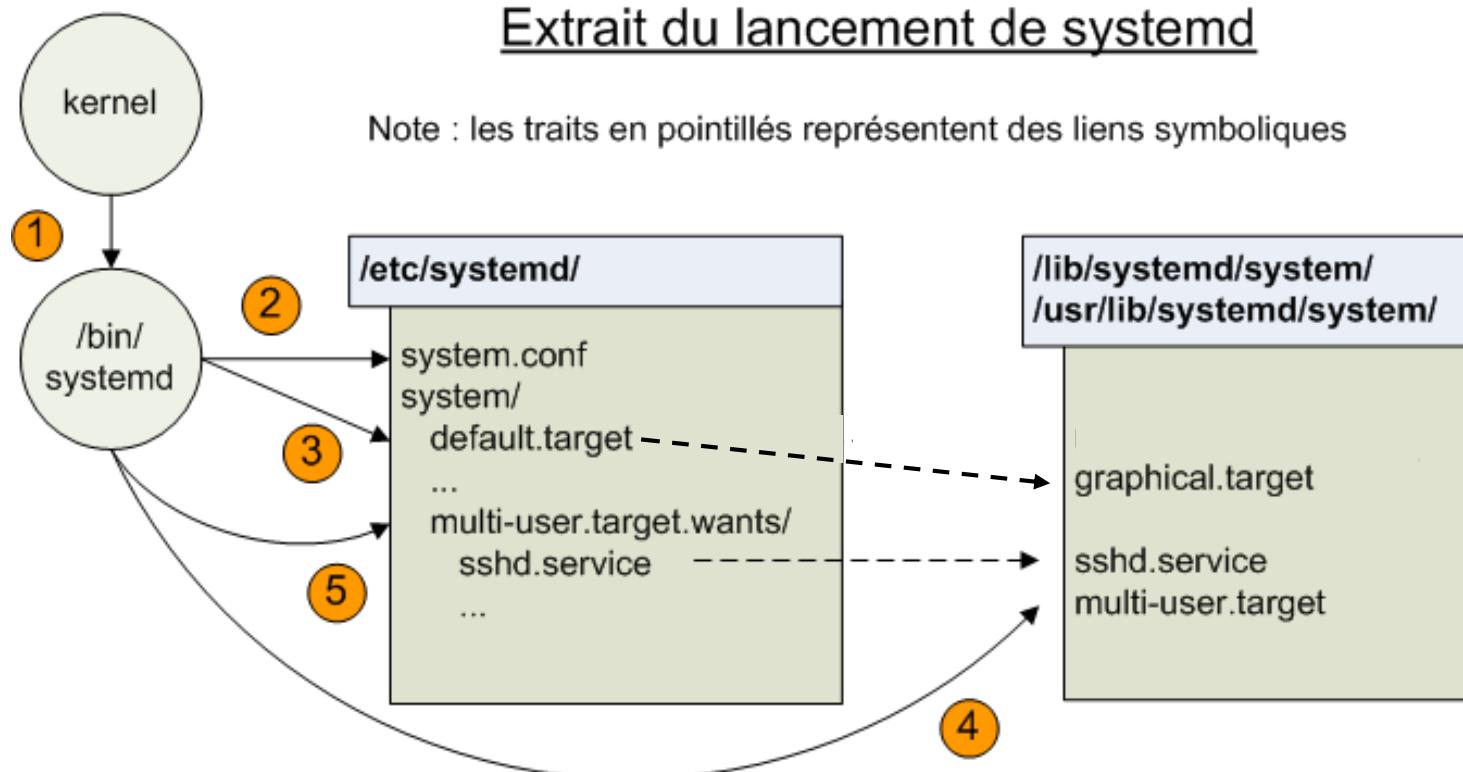
- **Les systèmes d'initialisation**

Systemd

- Démarrage

Extrait du lancement de systemd

Note : les traits en pointillés représentent des liens symboliques



Démarrage du système

- **Les systèmes d'initialisation**

Systemd

- Démarrage

1. Lorsque `systemd` est lancé par le noyau...
2. ... il lit son fichier `system.conf`, ...
3. puis examine la 1^{re} unité à lancer: `default.target` qui est en réalité un lien symbolique vers `graphical.target`.

Ce fichier contient une directive `After=multi-user.target`

4. ... donc `systemd` cherche la configuration de cette unité et l'interprète ...
5. ... mais il existe un dossier `multi-user.target.wants`, donc `systemd` parcourt ce répertoire pour activer les unités qui y sont décrivées.
On y trouve par exemple `sshd.service`, etc.



Démarrage du système

• Les systèmes d'initialisation

Systemd

- Terminaux:

```
getty.target -> getty.target.wants
getty@tty1.service ->/lib/systemd/system/getty@.service
```

- Actions sur les événements

```
ctrl-alt-del.target
```

- Changement de runlevel

```
systemctl isolate multi-user.target ou
systemctl isolate runlevel1.target ou
telinit 1 ou
init 1
```



Démarrage du système

• Les systèmes d'initialisation

Systemd

- Exemple 1 (`/etc/systemd/system/default.target`)



/usr/lib/systemd/system/graphical.target

La cible 'graphical' lance une interface graphique (runlevel 5).

```
[Unit]
Description=Graphical Interface
Documentation=man:systemd.special(7)
Requires=multi-user.target
After=multi-user.target
Conflicts=rescue.target
Wants=display-manager.service
AllowIsolate=yes

[Install]
Alias=default.target
```

graphical.target requiert la cible multi-user.target et démarrera après que multi-user ait achevé son démarrage.

graphical.target ne peut pas être activée en même temps que la cible rescue.target qui correspond au runlevel 1 !

graphical.target sera activée même si display-manager.service dont elle dépend échoue lors de son activation.

Rem: La directive Requires exige quant à elle que toutes les unités dont elle dépend soient activées avec succès !

La commande 'systemctl isolate graphical.target' passera la machine en mode graphique (c'est l'ancien init 5).

La commande 'systemctl enable graphical.target' exécutera un lien symbolique vers /etc/systemd/default.target.



Démarrage du système

• Les systèmes d'initialisation

Systemd

- Exemple 2

/usr/lib/systemd/system/sshd.service

Lancement et administration du service sshd.

```
[Unit]
Description=OpenSSH server daemon
After=network.target sshd-keygen.service
Wants:sshd-keygen.service
```

Description du service, et la gestion des interactions avec les autres services.

```
[Service]
EnvironmentFile=/etc/sysconfig/sshd
ExecStart=/usr/sbin/sshd -D $OPTIONS
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
Restart=on-failure
RestartSec=42s
```

Commande pour lancer le service, et éventuellement celles pour le relancer, le tuer...

```
[Install]
WantedBy=multi-user.target
```

Décrit comment ce service sera installé par une commande 'systemctl enable sshd.service'



Démarrage du système

• Les systèmes d'initialisation

Systemd

Lancement des services



Les services Avahi et Bluetooth nécessitent tous deux le fonctionnement de D-Bus, qui, à son tour, réclame Syslog.

- Système plus complexe.
- Parallélisation des processus par ouverture de sockets: `systemd` peut lancer en parallèle deux processus, même s'ils sont interdépendants.
- Cela peut réduire le temps de chargement.



Démarrage du système

• Les systèmes d'initialisation

Systemd

- Contrôle des services:

Remplacement de la commande 'chkconfig'	
chkconfig sshd on	systemctl enable sshd.service
chkconfig sshd off	systemctl disable sshd.service
chkconfig sshd --list	systemctl is-enable sshd.service
Remplacement de la commande 'service'	
service sshd status	systemctl status sshd.service service sshd status
service sshd start	systemctl start sshd.service service sshd start
service sshd stop	systemctl stop sshd.service service sshd stop
service sshd reload	systemctl reload sshd.service service sshd reload
Remplacement des commandes 'init' et 'telinit'	
init 0	systemctl poweroff init 0
init 6	systemctl reboot init 6
init 3	systemctl isolate runlevel3.target init 3
Choix du niveau d'exécution au démarrage	
valeur de initdefault dans /etc/inittab	valeur du lien symbolique default.target



Démarrage du système

- **Les systèmes d'initialisation**

Systemd

- Contrôle des services:

Fonctionnalité	sysVinit	systemd
Activation d'un service au démarrage	commande chkconfig et lien symbolique vers /etc/init.d/	chkconfig et liens symboliques pour les scripts non encore enregistrés. Commande systemctl pour les processus pris en charge par systemd
Démarrage ou arrêt d'un service	commande service	commande service ou commande systemctl
Changement de niveau d'exécution	commande init	commande init ou commande systemctl

- La commande **systemctl** centralise toutes les opérations...



Démarrage du système

• Les systèmes d'initialisation

Systemd

➤ Contrôle des services:

Autres opérations avec systemctl:

```
# systemctl list-unit-files (la liste de toutes les unités situées sous  
/lib/systemd/system/)
```

```
# systemctl list-units -t service (la liste et la description des unités de  
type service)
```

```
# systemctl daemon-reload (relecture de la configuration de systemd ou  
des unités par systemd – en cas de modification)
```

Activation/Désactivation d'un service au démarrage:

```
# systemctl enable <unit>  
# systemctl disable <unit>
```



Démarrage du système

- **Les systèmes d'initialisation**

Systemd

- Comparaison SysVinit et Systemd

Service Related Commands		
Comments	SysVinit	Systemd
Start a service	service dummy start	systemctl start dummy.service
Stop a service	service dummy stop	systemctl stop dummy.service
Restart a service	service dummy restart	systemctl restart dummy.service
Reload a service	service dummy reload	systemctl reload dummy.service
Service status	service dummy status	systemctl status dummy.service
Restart a service if already running	service dummy condrestart	systemctl condrestart dummy.service
Enable service at startup	chkconfig dummy on	systemctl enable dummy.service
Disable service at startup	chkconfig dummy off	systemctl disable dummy.service
Check if a service is enabled at startup	chkconfig dummy	systemctl is-enabled dummy.service
Create a new service file or modify configuration	chkconfig dummy --add	systemctl daemon-reload

Note : New version of systemd support “systemctl start dummy” format.



Démarrage du système

- **Les systèmes d'initialisation**

Systemd

- Comparaison SysVinit et Systemd

Runlevels		
Comments	SysVinit	Systemd
System halt	0	runlevel0.target, poweroff.target
Single user mode	1, s, single	runlevel1.target, rescue.target
Multi user	2	runlevel2.target, multi-user.target
Multi user with Network	3	runlevel3.target, multi-user.target
Experimental	4	runlevel4.target, multi-user.target
Multi user, with network, graphical mode	5	runlevel5.target, graphical.target
Reboot	6	runlevel6.target, reboot.target
Emergency Shell	emergency	emergency.target
Change to multi user runlevel/target	telinit 3	systemctl isolate multi-user.target (OR systemctl isolate runlevel3. target)
Set multi-user target on next boot	sed s/^id::*:initdefault:/ id:3:initdefault:/	In -sf /lib/systemd/system/multi- user.target /etc/systemd/system/ default.target
Check current runlevel	runlevel	systemctl get-default
Change default runlevel	sed s/^id::*:initdefault:/ id:3:initdefault:/	systemctl set-default multi-user.target



Démarrage du système

- **Les systèmes d'initialisation**

Systemd

- Comparaison SysVinit et Systemd

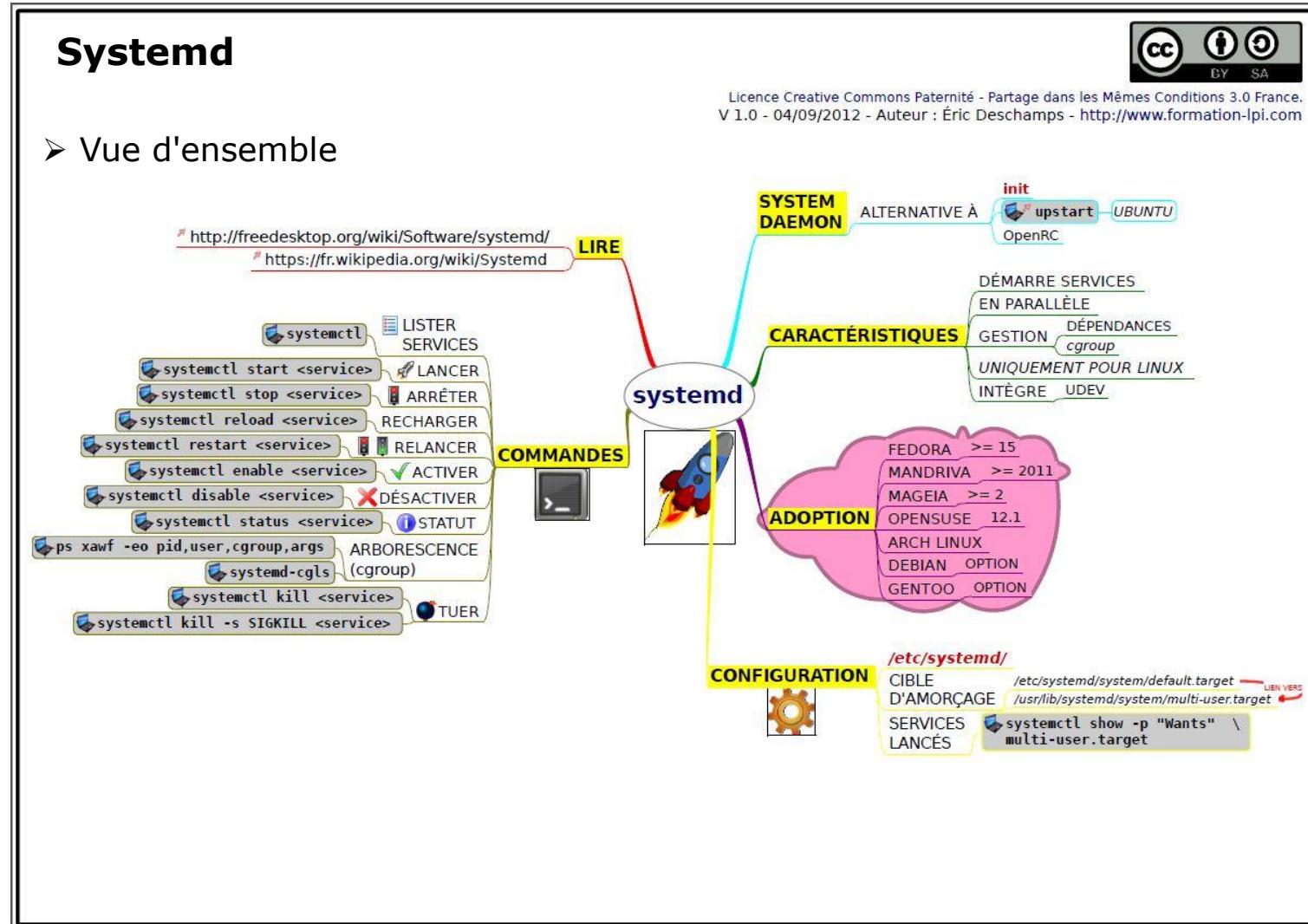
Miscellaneous Commands		
Comments	SysVinit	Systemd
System halt	halt	systemctl halt
Power off the system	poweroff	systemctl poweroff
Restart the system	reboot	systemctl reboot
Suspend the system	pm-suspend	systemctl suspend
Hibernate	pm-hibernate	systemctl hibernate
Follow the system log file	tail -f /var/log/messages or tail -f /var/log/syslog	journalctl -f

Systemd New Commands	
Comments	Systemd
Execute a systemd command on remote host	systemctl dummy.service start -H user@host
Check boot time	systemd-analyze or systemd-analyze time
Kill all processes related to a service	systemctl kill dummy
Get logs for events for today	journalctl --since=today
Hostname and other host related information	hostnamectl
Date and time of system with timezone and other information	timedatectl



Démarrage du système

- Les systèmes d'initialisation



Démarrage du système

• Les systèmes d'initialisation

Systemd

➤ Plus d'info:

- SYSTEMD VAINQUEUR DE UPSTART ET DES SCRIPTS SYSTEM V ?
Linux Magazine France N°153
- https://fedoraproject.org/wiki/SysVinit_to_Systemd_Cheatsheet
- <https://fedoraproject.org/wiki/Systemd>
- <http://linoxide.com/linux-command/systemd-vs-sysvinit-cheatsheet/>
- <https://www.mathrice.fr/IMG/pdf/systemd.pdf>
- <https://www.redhat.com/archives/rh-community-de-nrw/2014-July/pdfC1TTlawaxB.pdf>



Démarrage du système

• Arrêt du système

Introduction

- Les systèmes Linux ne peuvent pas être éteints à chaud.
- En effet, Linux stocke dans une mémoire tampon toutes les opérations de lecture/écriture sur le disque dur ou la disquette pour minimiser les accès disque.
- Il est donc possible qu'une nouvelle version d'un fichier, d'un i-node ou d'un répertoire soit dans la mémoire tampon, c'est-à-dire en Ram, le disque dur ne contenant que l'ancienne version.
- Si vous éteignez alors votre ordinateur, le système de fichiers deviendra incohérent, avec tous les risques de pertes de données que cela suppose.

Précautions à prendre avant d'arrêter un système Linux:

- Interrompre tous les processus utilisateurs en cours sur le système.
- Transférer le tampon vers le disque dur: commande sync.
- Extinction de la machine.

Pour ce faire, chaque système d'initialisation possède ses propres commandes.



Démarrage du système

• Arrêt du système

sysVinit

```
# shutdown -h now ou halt ou init 0 (arrêt immédiat)  
# shutdown -r now ou reboot ou init 6 (redémarrage immédiat)
```

☞ Derrière toutes ces commandes, se cache l'accès aux scripts de niveau 0 ou de niveau 6.

➤ Quelques autres utilisations de la commande shutdown

```
# shutdown -h +2 "Extinction de la machine dans 2 min" &  
Arrêt du système dans 2 minutes avec diffusion d'un message .
```

```
# shutdown -c "Annulation du shutdown"  
Annulation du shutdown précédent avec diffusion d'un message .
```

```
# shutdown -h 11:59 &  
Arrêt du système à 11h 59.
```



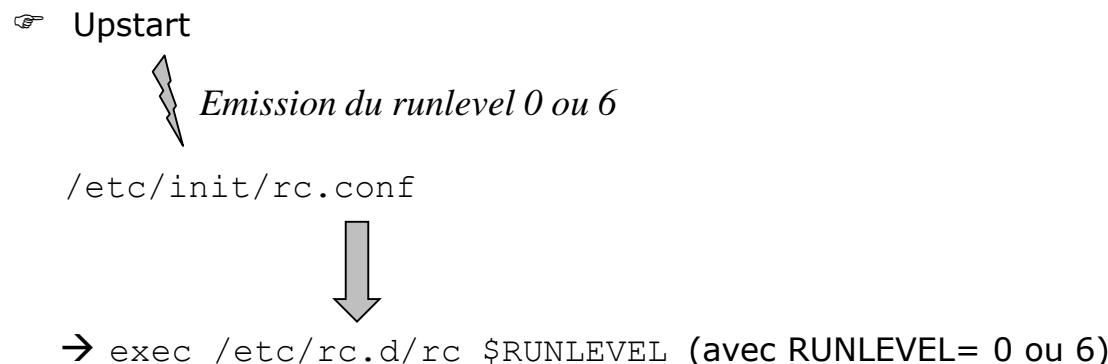
Démarrage du système

- **Arrêt du système**

Upstart

Compatibilité avec sysVinit

```
# shutdown -h now ou halt ou init 0 (arrêt immédiat)
# shutdown -r now ou reboot ou init 6 (redémarrage immédiat)
```



Démarrage du système

• Arrêt du système

Systemd

```
# systemctl halt (arrêt immédiat)
```

→ runlevel0.target
poweroff.target

```
# systemctl reboot (redémarrage immédiat)
```

→ runlevel6.target
reboot.target



Démarrage du système

Atelier

L'init sous CentOS 6.x



Démarrage du système

- **Autres ressources**

- Linux Administration (Tome 2) – Chapitre 10: Le démarrage
Jean-François Bouchaudy – 2^{ème} Edition – Eyrolles 2011



GNU/Linux Operating System

STOCKAGE LE RAID

HELHa

Haute École
Louvain en Hainaut



Plan

- **LE RAID**

Présentation

Description / Buts

RAID matériel ou logiciel

Le RAID matériel / Le RAID logiciel / Choix

Les niveaux de RAID

Introduction / Les niveaux simples / Les niveaux combinés

Atelier: Le RAID logiciel

Références



Le RAID

• Présentation

Description

- RAID: *Redundant Array of Independant Disks*
- Un disque (*array* ou *matrice* ou *grappe*) RAID est donc constitué d'un ensemble (*agrégat*) de disques qui ne sera vu que comme une seule entité par le système d'exploitation.
- La technologie RAID a été créée en 1987 à l'université de Berkeley.

Le but initial: remplacer un disque "gros système" très coûteux par un ensemble de disque "classiques" beaucoup moins chers.

→ terminologie initiale:

Redundant Array of Inexpensive Disks



Le RAID

• Présentation

Buts

- Etendre la capacité de stockage (but initial)

Une grappe de 2 disques de 250 Go et de 500 Go seront vu comme un seul disque de 1,5 To.

Mais aussi actuellement:

- Obtenir une meilleure tolérance aux pannes

. Objectifs:

Haute disponibilité: En cas de panne d'un des disques, les autres continuent de tourner normalement .

Le système continue de fonctionner.

Transparence: Si le contrôleur le permet, le disque défectueux peut être changé à chaud (*hotplug*).

Son contenu sera automatiquement reconstruit à partir du contenu des autres disques.



Le RAID

• Présentation

Buts (suite)

- Obtenir une meilleure tolérance aux pannes (suite)

- . Techniques:

Soit par duplication des données sur plusieurs disques (*mirroring*).

Le disque défectueux sera reconstruit par recopie des données d'un des autres disques.

Soit par l'utilisation des données de parité (*error correction*).

Les données sont écrites (ou lues) par bandes (*strips ou chunks*).

Ainsi, une matrice RAID de $n+1$ disques pourrait être gérée par l'éclatement des données sur les n premiers disques et la génération d'une bande de parité sur le $n+1^{\text{ème}}$ disque.

Les données (bandes) d'un disque hors service pourront être reconstituées à l'aide des bandes des autres disques

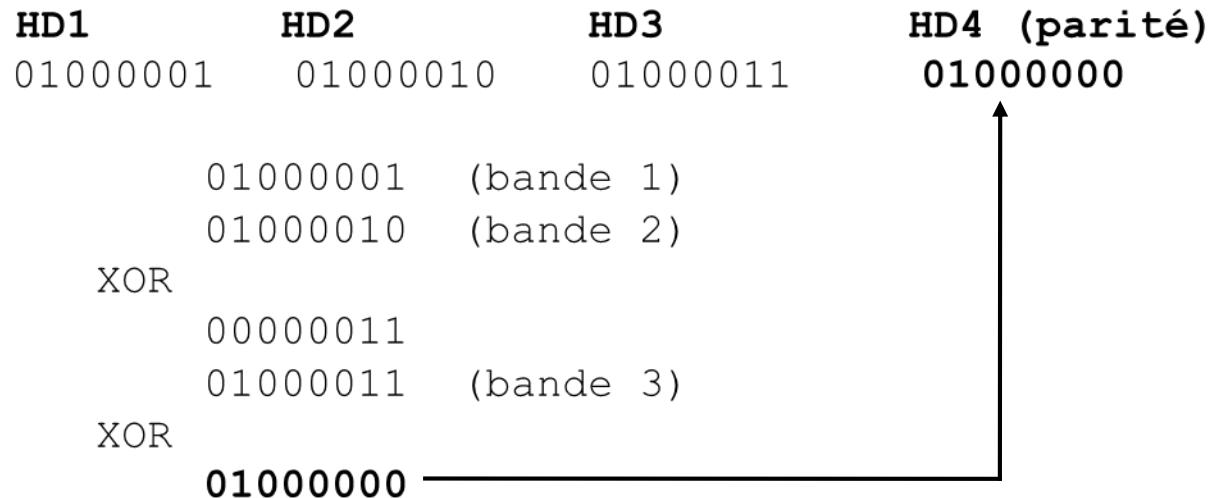


Le RAID

- Présentation

Buts (suite)

- Obtenir une meilleure tolérance aux pannes (suite)
 - . Exemple: Soit une grappe de 4 disques et une bande d'un byte (*)
 - a) Le contrôleur RAID doit se charger de l'écriture de "ABC"...



(*) En réalité, la taille d'une bande (*chunk-size*) est beaucoup plus élevée
(ex. entre 16 et 2048 ko).

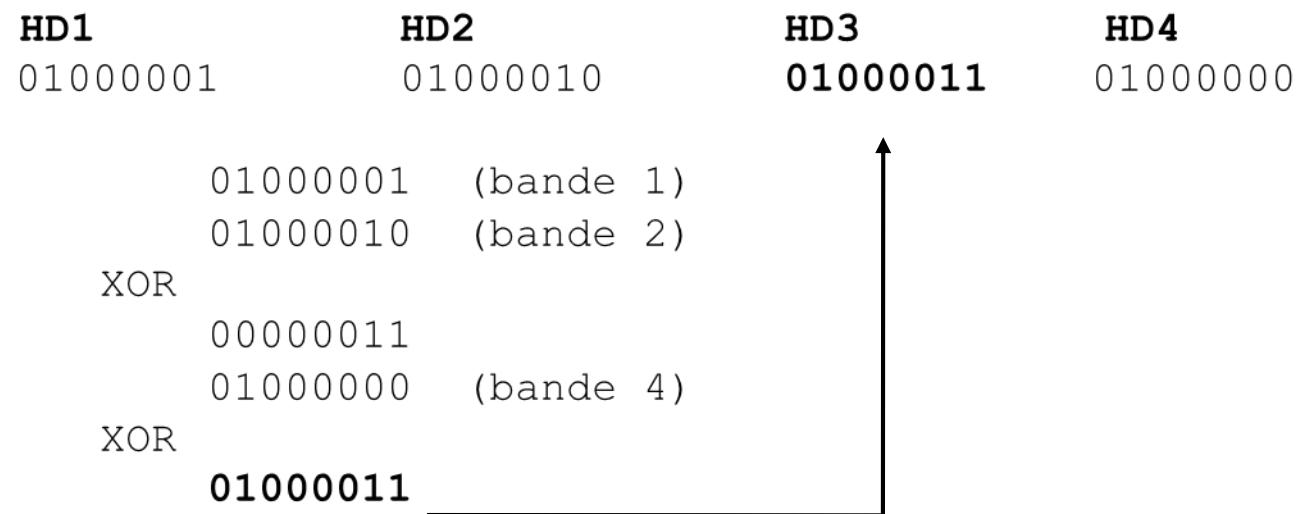


Le RAID

- Présentation

Buts (suite)

- Obtenir une meilleure tolérance aux pannes (suite)
 - . Exemple (suite)
 - b) Panne du disque 3: Reconstitution de ses bandes par le contrôleur RAID



Le RAID

• Présentation

Buts (suite)

- Obtenir une meilleure tolérance aux pannes (suite)

- . Sauvegardes

Une installation de type RAID, bien que consistant à dispatcher des données sur plusieurs disques durs, n'est en aucun cas une technique de sauvegarde.

Elle ne dispense en aucun cas de faire des sauvegardes régulières...

- Obtenir des meilleures performances

Par l'éclatement des I/O sur plusieurs disques gérés chacun par un contrôleur distinct. Dans ce cas les I/O sont parallélisées.



Le RAID

• Les types de RAID

Le RAID matériel

- Géré par un matériel dédié
 - . Avec des contrôleurs de disques RAID internes

Il s'agit d'un adaptateur (IDE, SATA ou SCSI) située sur la carte mère ou rajoutée via un slot d'extension PCI gérant la technologie RAID. Ces adaptateurs peuvent être dotés d'interfaces IDE (obsolète), SATA ou SCSI.

Le type d'adaptateur dépend du type de disque qu'il est sensé gérer. Le coût de la technologie choisie ne dépend pas uniquement du coût de la carte (nombre de slots) mais surtout du coût des disques qui en dépendent.

Il est accompagné d'un logiciel de gestion.



Adaptec RAID 3085



Le RAID

• Les types de RAID

Le RAID logiciel

- Complètement géré par un driver au niveau de l'OS.
- Dans le cas Linux, le driver et les niveaux de RAID qu'il devra gérer doivent être compilé correctement dans le noyau.

```
Multiple devices driver support (RAID and LVM)
<*>  RAID support
[*]    Autodetect RAID arrays during kernel boot
<*>  Linear (append) mode
<*>  RAID-0 (striping) mode
< >  RAID-1 (mirroring) mode
< >  RAID-10 (mirrored striping) mode
< >  RAID-4/RAID-5/RAID-6 mode
< >  Multipath I/O support
< >  Faulty test module for MD
```



Le RAID

- **RAID matériel ou logiciel**

Choix

RAID logiciel (***)	RAID matériel
Implémenté dans le noyau de l'OS.	Compilé au sein de la carte d'extension ou de la carte mère.
Plus flexible. Moins cher.	Plus rapide (*). Plus cher.
Dépendance vis-à-vis de l'OS. Requiert un noyau compilé correctement. Utilise les ressources du système (*).	Dépendance vis-à-vis du matériel. La gestion du RAID n'est plus à la charge du noyau.
(*) De moins en moins contraignant vu la puissance des serveurs actuels.	(*) De moins en moins vrai vu la puissance des serveurs actuels.



Le RAID

• Les niveaux de RAID

Introduction

- Les disques assemblés en technologie RAID peuvent être utilisés de différentes façons, appelées "*niveaux RAID*".
- Les niveaux les plus courants: 0, 1 et 5 pour les niveaux simples
01 et 10 pour les niveaux combinés
- Le choix d'implémentation d'un niveau de RAID dépend, après analyse des besoins, du compromis que l'on est prêt à accorder entre:
 - Performance / Coût / Sécurité
- Tous les types de RAID ne sont pas spécialement gérés par tous les matériels ou tous les systèmes d'exploitation (voir documentation de l'éditeur).



Le RAID

• Les niveaux de RAID

Introduction (suite)

Quand un disque faisant partie d'un RAID tombe en panne, on est à la merci d'une deuxième panne tant qu'on n'a pas remplacé le disque hors service → risque de perte des données de toute la matrice RAID

- ➔ Configuration possible d'un disque de rechange (*Spare disk*) excepté en RAID linéaire et en RAID 0.
 - . Utilisé pour reconstruire immédiatement et automatiquement le RAID dans le cas d'une panne d'un disque.
 - . Le RAID sera quand même défaillant si une 2^{ème} panne survient pendant la reconstruction du disque.



Le RAID

- **Les niveaux simples**

RAID linear

- Appelé aussi RAID 0 concat ou JDOB (Just a Bunch Of Disks)
- Implémenté sur toutes les cartes Raid et OS.
- Former une unité logique de grande capacité en concaténant des disques.
- Ecriture des données: Quand le 1^{er} disque est rempli, on continue les ajouts sur le 2^{ème}, et ainsi de suite.
- Ce niveau est considéré comme un mode RAID bien qu'il n'implémente pas la redondance de données.



Le RAID

- **Les niveaux simples**

RAID linear (suite)

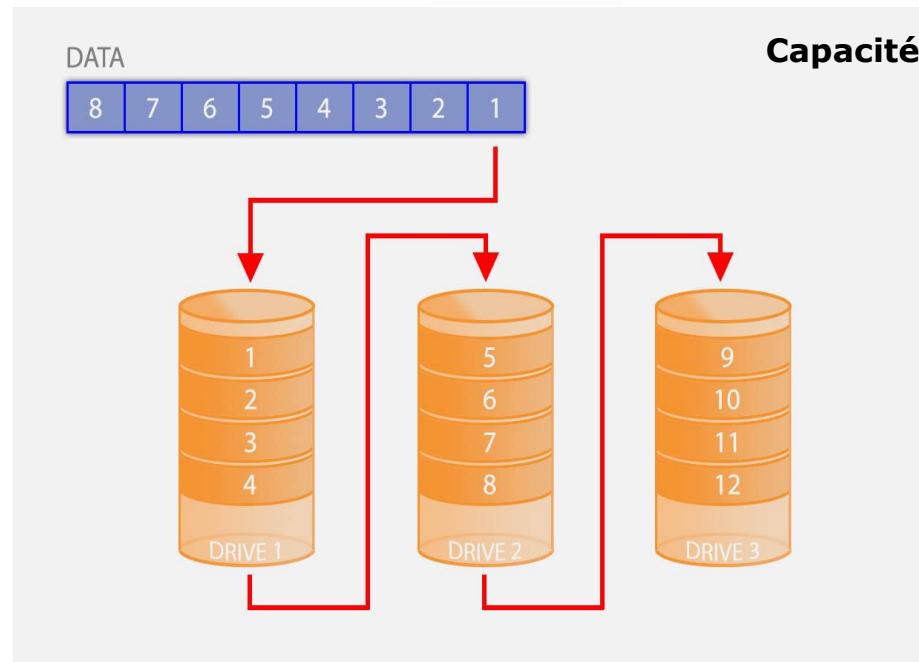
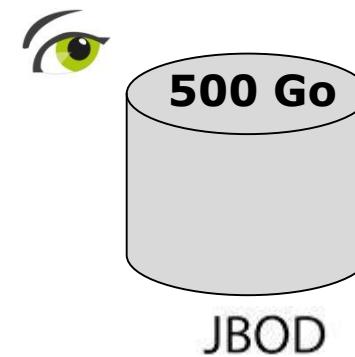
- ☺ Disposer d'un plus grand volume disque.
- ☹ Pas de tolérance aux pannes

perte d'un disque
=
perte de toutes les données

Pas d'amélioration des performances car les I/O ne sont pas fractionnées entre les disques.

Mise en pratique:

2 disques minimum (sans spare disk)



Le RAID

- **Les niveaux simples**

RAID 0 Stripe

- Implémenté sur toutes les cartes Raid et OS.
- Il utilise le principe de la segmentation des données.
- Ecriture des données:
 - Les données sont divisées en bandes (strips ou chunks) qui sont réparties sur les différents disques (*).
- Ce niveau est considéré comme un mode RAID bien qu'il n'implémente pas la redondance de données.

(*) La taille des bandes influence fortement le débit de transfert moyen
(plus petite est la bande, meilleur est le débit)



Le RAID

- **Les niveaux simples**

RAID 0 Stripe (suite)

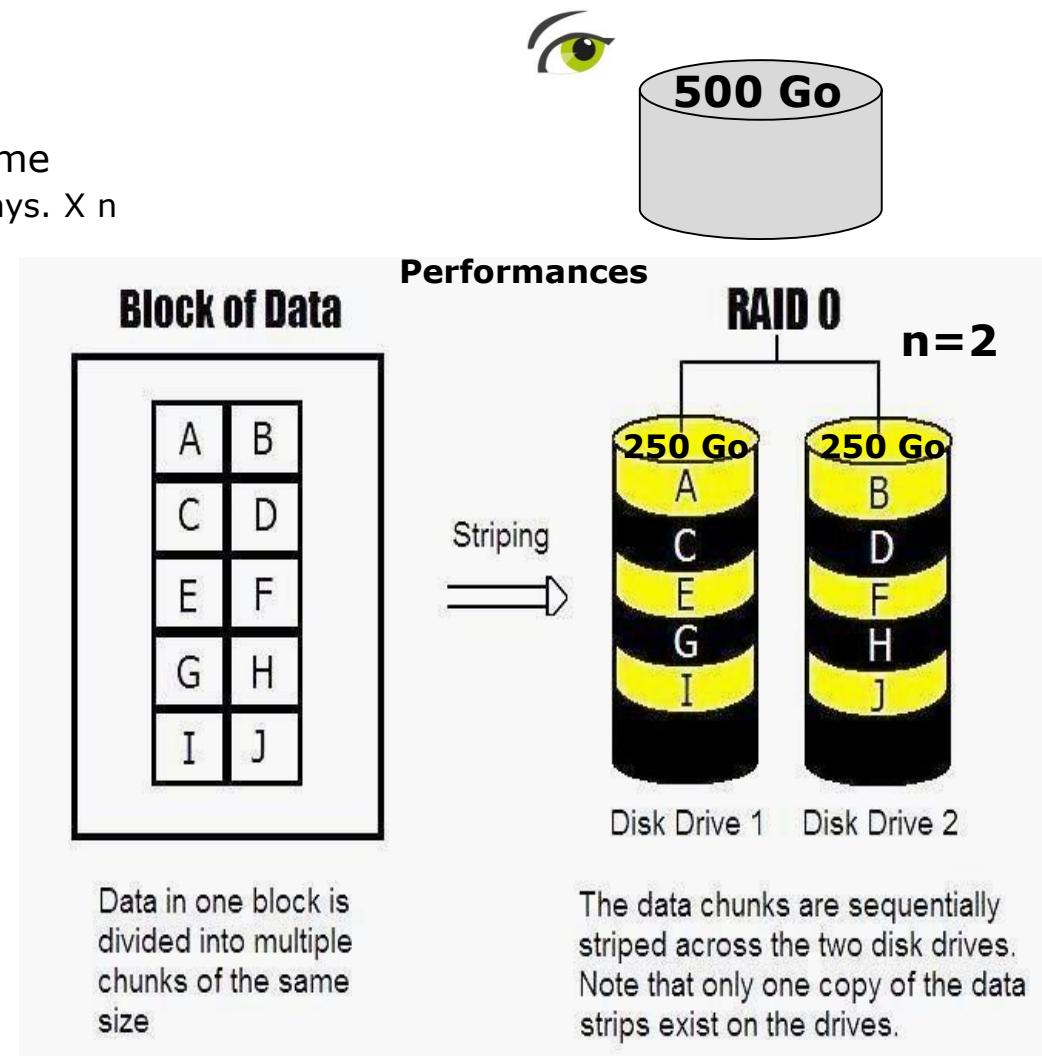
- ☺ Disposer d'un plus grand volume disque: Volume logique = Vol. phys. X n

Performances accrues (n bandes peuvent être lues ou écrites simultanément) mais limitées à environ n fois le débit du disque le plus lent de la matrice.

- ☺ Pas de tolérance aux pannes
Pas de spare disk.

Mise en pratique:

2 disques minimum (sans spare disk) de taille et débits similaires de préférence.



Le RAID

• Les niveaux simples

RAID 0 Stripe (suite)

- En RAID, une bande devient la plus petite unité d'allocation.

On fixe généralement le "chunk size" au "block size" du file system qui sera formaté sur la matrice RAID.

Conventions

Chunk size

Gros fichiers	Haut (256 Ko ou plus)
Petits fichiers	Bas (0.5, 1, 2 Ko ...)

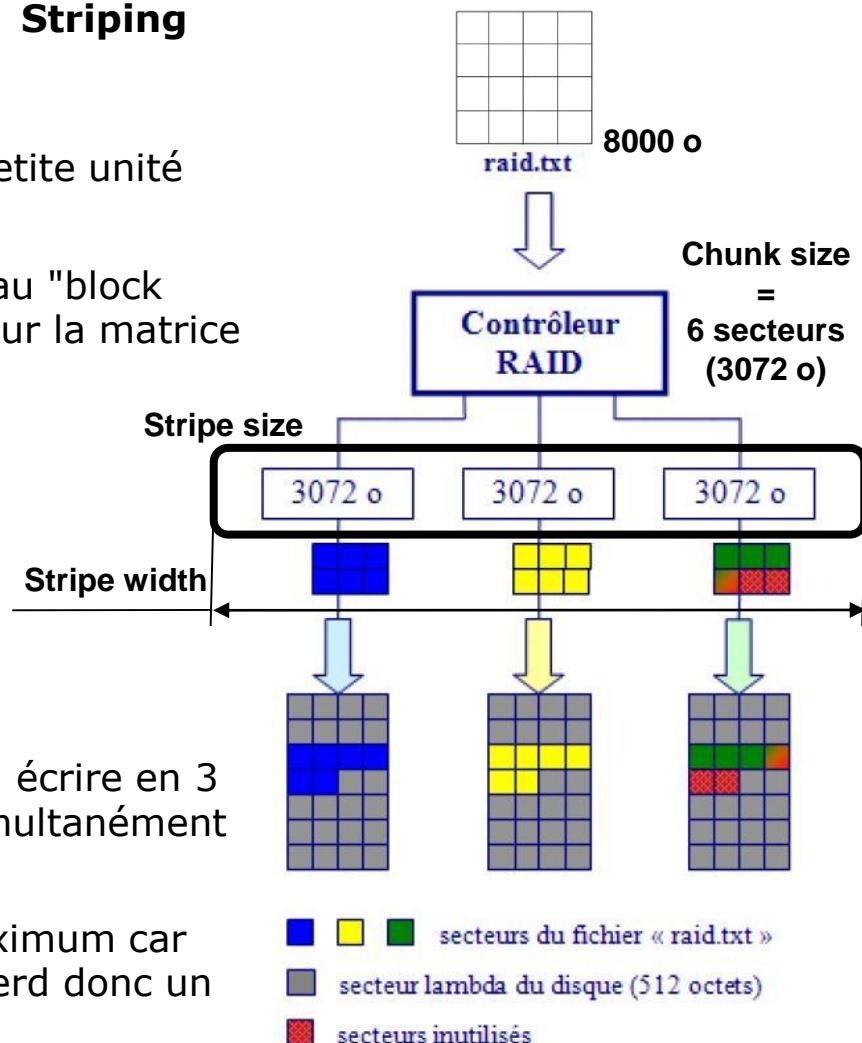
Exemple (ci-contre):

Le contrôleur RAID découpe le fichier à écrire en 3 bandes de 3072 bytes puis les écrit simultanément sur les 3 disques.

La 3^{ème} bande n'est pas utilisée au maximum car elle ne contient que 1856 octets. On perd donc un espace de stockage de 1216 octets.



Principe d'une segmentation Striping



Le RAID

- **Les niveaux simples**

RAID 1

- Appelé aussi "mirroring".
- Implémenté sur toutes les cartes Raid et OS.
- Il répond à une des règles fondamentales de la sécurité des systèmes RAID:

Lors d'une panne "disque", il faut:

- . disposer d'une copie des données
- . que le système reste fonctionnel en attendant le remplacement du disque défectueux.
- Ecriture des données:
Chaque segment du fichier est dupliqué sur chaque disque de la matrice RAID.



Le RAID

- **Les niveaux simples**

RAID 1 (suite)

- ☺ Tolérance aux pannes: ce raid survit à la perte de n-1 disques.

Amélioration des performances en lecture si le contrôleur raid le permet:

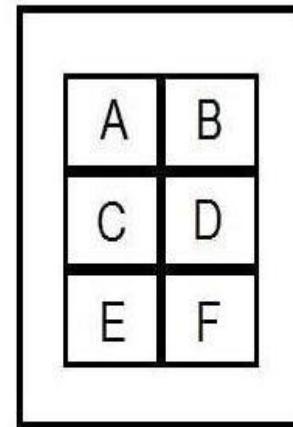
1^{ère} lecture: A sur hd1 à B sur hd2 simult.
2^{ème} lecture: C sur hd1 à D sur hd2 simult.

...

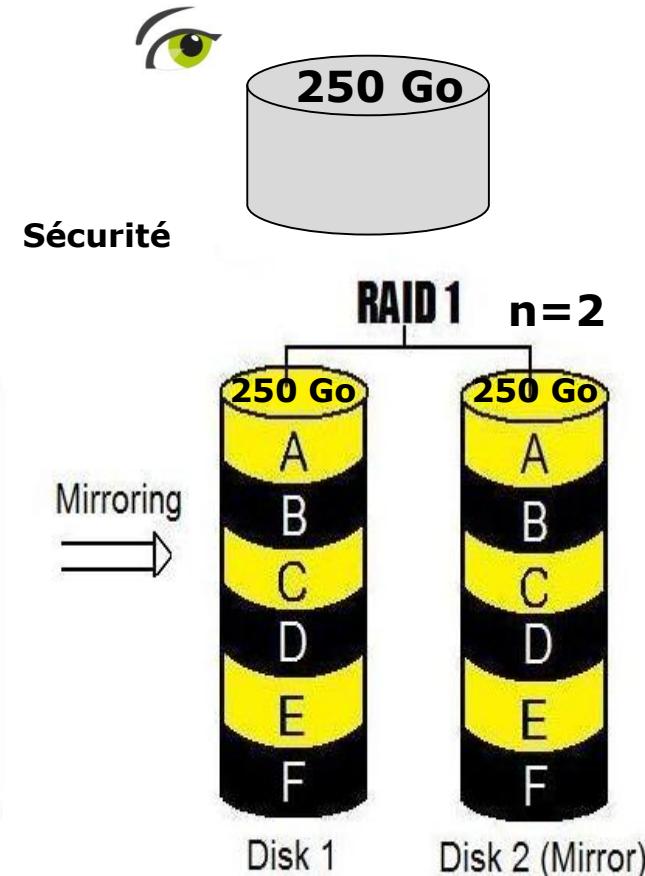
- ☺ Performance en écriture réduite à la performance d'écriture du disque le plus lent de la matrice.

Coût de stockage élevé et en corrélation avec le nombre de disques: Vol. log. = Plus petit vol. phys.

Block of Data



A data set is divided into chunks of the same size



The chunks are stored in Disk 1 and an exact copy of each chunk is stored in Disk 2. Thus a mirror of Disk 1 is created.



Mise en pratique:

2 disques minimum de taille et performances similaires de préférence.

Recommandé sur des serveurs stockant des données sensibles et tournant 24h/24h.

Le RAID

- **Les niveaux simples**

RAID 5

- Utilise le "striping" et la "parité" répartie sur les différents disques.
- Un disque peut tomber en panne sans qu'on sans rendre compte.

Lors d'une panne "disque", la matrice RAID reste opérationnelle car les données du disque défectueux sont simulées à partir des données utiles et des parités réparties sur les disques restant en fonction.

- Ecriture des données:

Les données sont divisées en bandes (strips ou chunks) qui sont réparties sur les différents disques avec une parité pour chaque répartition.

- Souvent associé aux technologies hot swap (extraction à chaud des disques) et hot spare (reconstruction dynamique à chaud).



Le RAID

- **Les niveaux simples**

RAID 5 (suite)

- ☺ Tolérance aux pannes: ce raid survit à la perte d'un seul disque.

Performances en lecture aussi élevées qu'en RAID 0.

Disposer d'un plus grand volume disque:
Volume logique= Vol. phys. min. X (n-1)

Surcoût de stockage minimal et d'autant plus faible que n est élevé.

- ☹ Petite pénalité en écriture du fait du calcul de la parité.

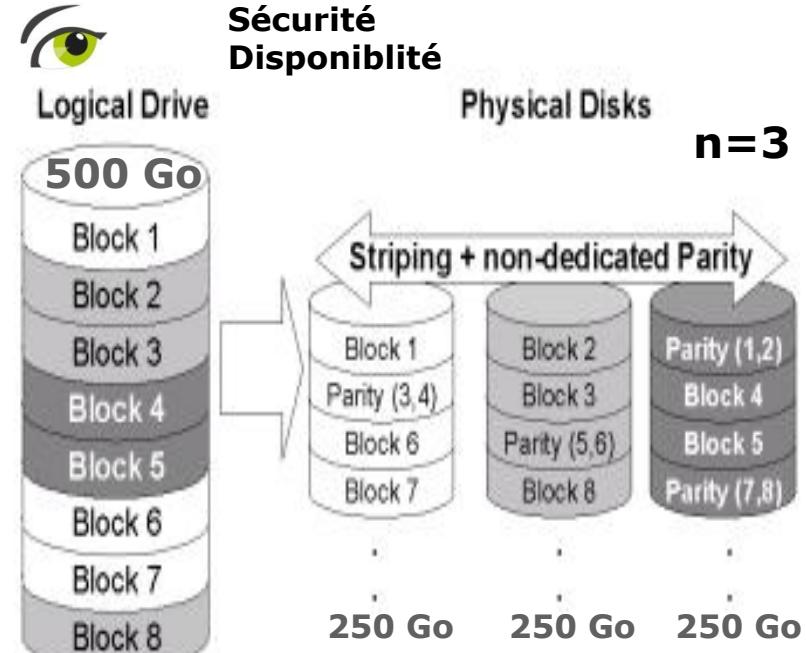
Lorsqu'un disque est défectueux, grosse pénalité due à la parité à recalculer à chaque I/O (uniquement tant que le nouveau disque n'est pas reconstitué).

Mise en pratique:

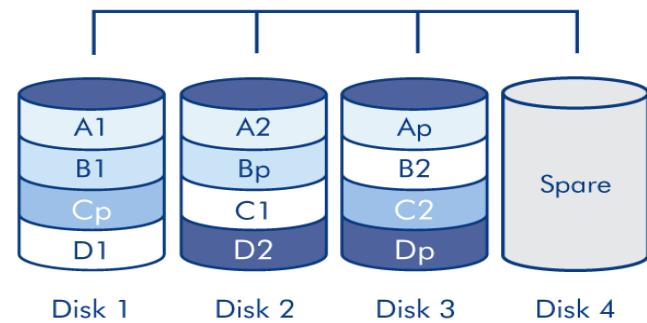
3 disques minimum de taille et performances similaires de préférence.

+ 1 spare disk (souvent).

Gouwy Jean-Louis



RAID 5+Spare



Le RAID

- **Les niveaux simples**

RAID 5 (suite)

- En RAID5, les parités sont réparties sur les différents disques contrairement au RAID4 où elles sont stockées sur un même disque.
- Augmentation des performances en réduisant les files d'attente dues à l'écriture des données de parité.

Exemple:

- . Soient 2 écritures (E1 & E2) à effectuer sur un système RAID4 et un système RAID5.
- . Chacune des écritures est accompagnée du calcul des données de parité (C1 & C2).
- . Les données de parité (P1 & P2) sont écrites.

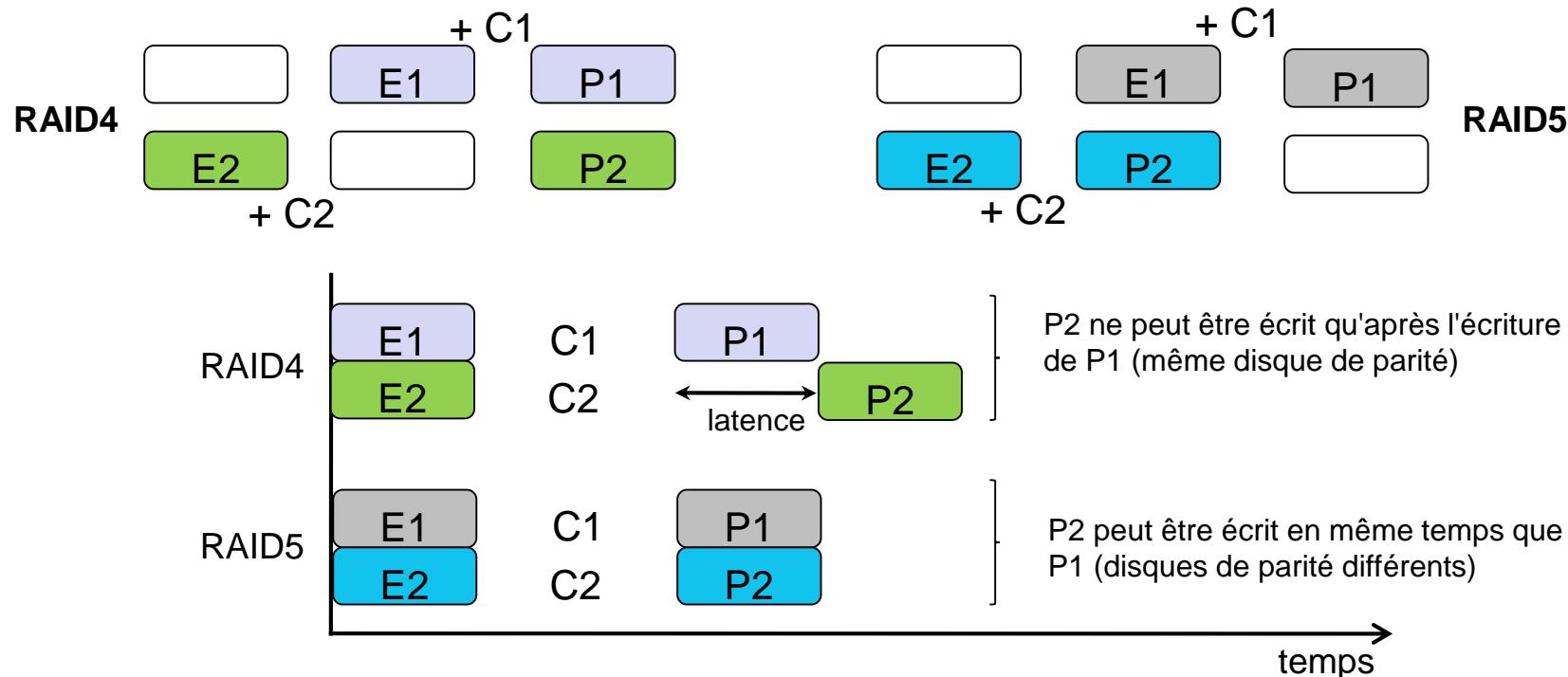


Le RAID

- **Les niveaux simples**

RAID 5 (suite)

Exemple: suite



Le RAID

- **Les niveaux simples**

RAID 5 (suite)

→ Réduction de l'usure du disque de parité.

En RAID4, une écriture entraîne d'office une écriture sur le même disque de parité. Ce qui n'est pas le cas en RAID5.

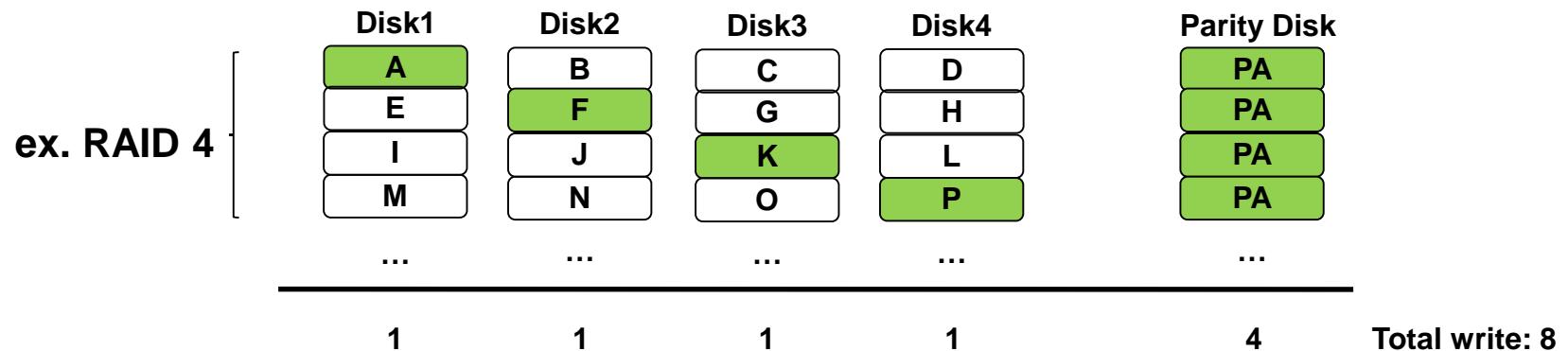


Le RAID

- **Les niveaux simples**

RAID 5 (suite)

- Soit la réécriture de nouveaux chunks A, F, K et P sur le système RAID 4 suivant:



Chaque écriture sur les différents disques entraîne d'office une réécriture d'une bande de parité sur le même disque.

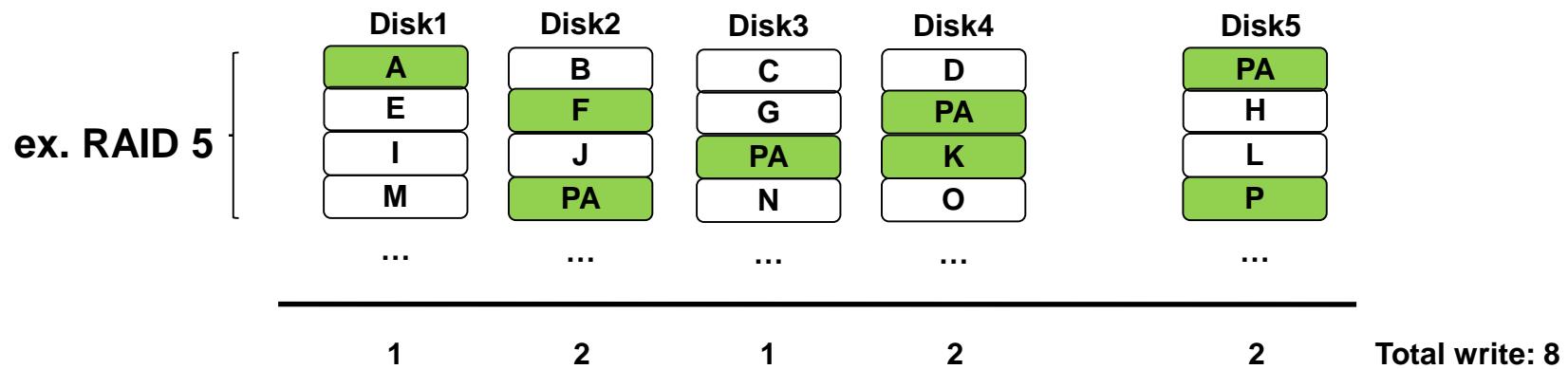


Le RAID

- **Les niveaux simples**

RAID 5 (suite)

→ Même cas de figure mais sur le système RAID 5 suivant:



Chaque écriture sur les différents disques entraîne d'office une réécriture d'une bande de parité sur des disques différents.



Le RAID

- **Les niveaux simples**

Les moins utilisés

- RAID 2: Evolution du RAID 1 (n'est plus utilisé).
- RAID 4: Ressemble au RAID 5, mais les données de parité sont centralisées sur un seul disque, ce qui diminue à la fois les performances et la fiabilité.
- RAID 6: Est une évolution du RAID 5, mais avec deux bandes de parité au lieu d'une, ce qui accroît la fiabilité (on peut perdre deux disques) mais ce qui accroît également son coût.
La technique du XOR est remplacée par une technique de calcul de parité fondée sur les polynômes (hors cadre du cours).



Le RAID

- **Les niveaux combinés**

RAID 01 (0+1)

- C'est un disque logique (RAID 1 mirroring) construit sur y ensembles de disques logiques (RAID 0 striping construits chacun sur n disques).
- Chaque matrice RAID 0 forme un tout, quel que soit le nombre de disques durs qui y participent.
- Le mirroring s'applique à l'ensemble du striping.
- Ecriture des données:
Les données sont divisées en bandes qui sont "mirrorées" et réparties sur les différentes matrices RAID 0.
- Construction:
 1. Construire les matrices RAID 0
 2. Créer un RAID 1 à partir de ces matrices RAID 0



Le RAID

- **Les niveaux combinés**

RAID 01 (suite)

- ☺ Tolérance aux pannes: ce raid survit à la perte d'un côté du miroir:

Si l'un des disques tombe, le mirroring s'arrête.

Performances élevées en lecture/écriture

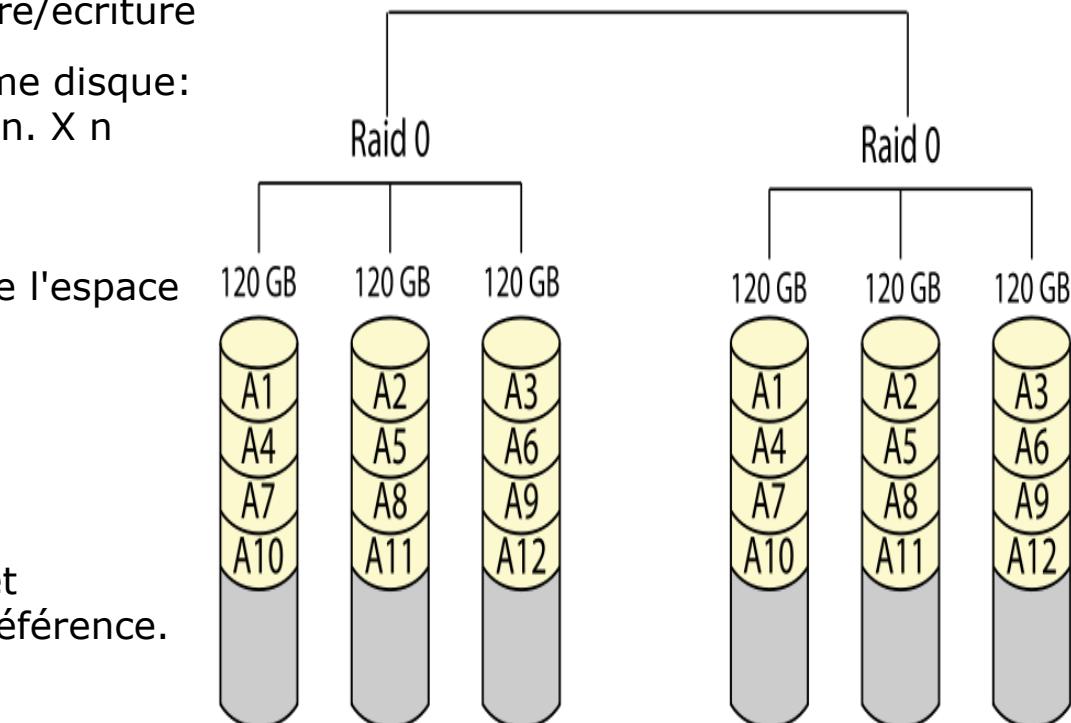
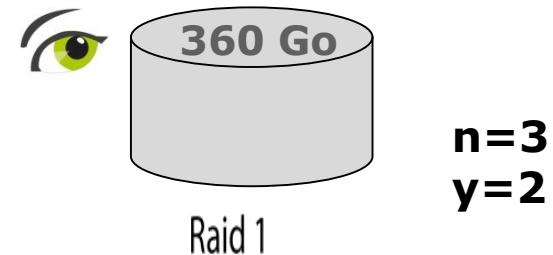
Disposer d'un plus grand volume disque:
Volume logique= Vol. phys. min. X n

- ☹ Perte de 50% (au minimum) de l'espace disque.

Surcoût de stockage.

Mise en pratique:

4 disques minimum de taille et performances similaires de préférence.



Le RAID

- **Les niveaux combinés**

RAID 10 (1+0)

- C'est un disque logique (RAID 0 striping) construit sur y ensembles de disques logiques (RAID 1 mirroring construits chacun sur n disques).
- Chaque matrice RAID 1 forme un tout, quel que soit le nombre de disques durs qui y participent.
- Le striping s'applique à l'ensemble du mirroring.
- Ecriture des données:
 - Les données sont divisées en bandes qui sont réparties sur les différentes matrices RAID 1 puis "mirrorées" sur chaque disque de ces matrices.
- Construction:
 1. Construire les matrices RAID 1
 2. Créer un RAID 0 à partir de ces matrices RAID 1



Le RAID

- **Les niveaux combinés**

RAID 10 (suite)

- ☺ Meilleure tolérance aux pannes que RAID 01:
Ce raid survit à la perte de n-1 disques et ce, sur chaque ensemble.

Performances élevées en lecture/écriture

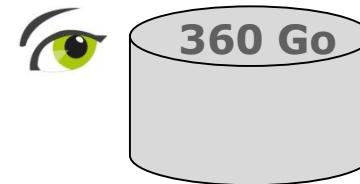
Disposer d'un plus grand volume disque:
Volume logique= Vol. phys. min. X y

- ☹ Perte de 50% (au minimum)de l'espace disque.

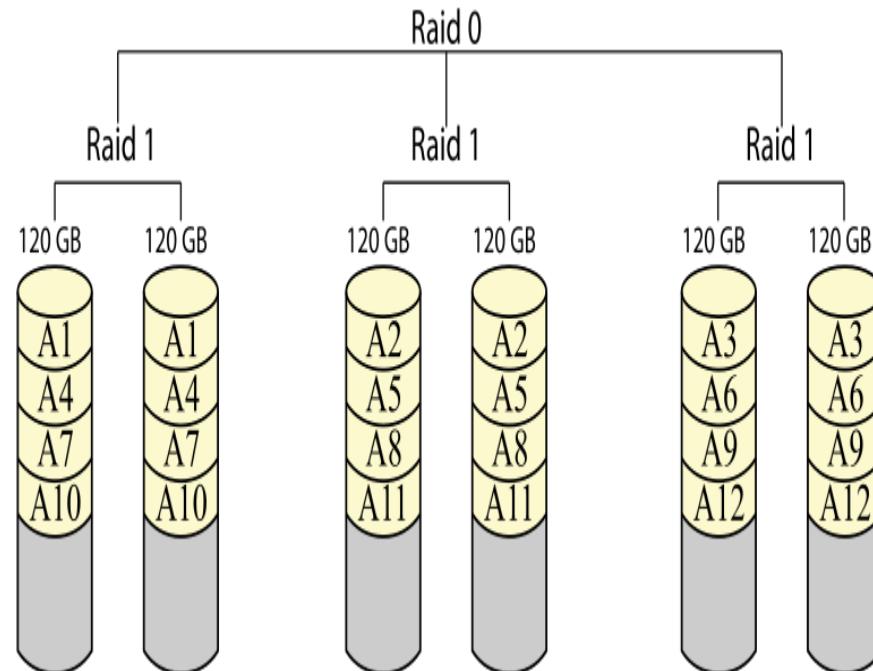
Surcoût de stockage.

Mise en pratique:

4 disques minimum de taille et performances similaires de préférence.



n=2
y=3



Le RAID

• **Les techniques de stockage réseau**

Introduction

- Explosion des données en volume et en importance.
- Décroissance du coût du stockage physique.
- Coût de gestion et d'entretien en augmentation.

- Où mettre les données ?
Comment les organiser ?
Comment les gérer ?
Comment permettre leur accès aux utilisateurs ?
- Avec bien sûr des objectifs:
 - . d'intégrité
 - . de performance
 - . de sécurité
 - . de limitation des coûts.

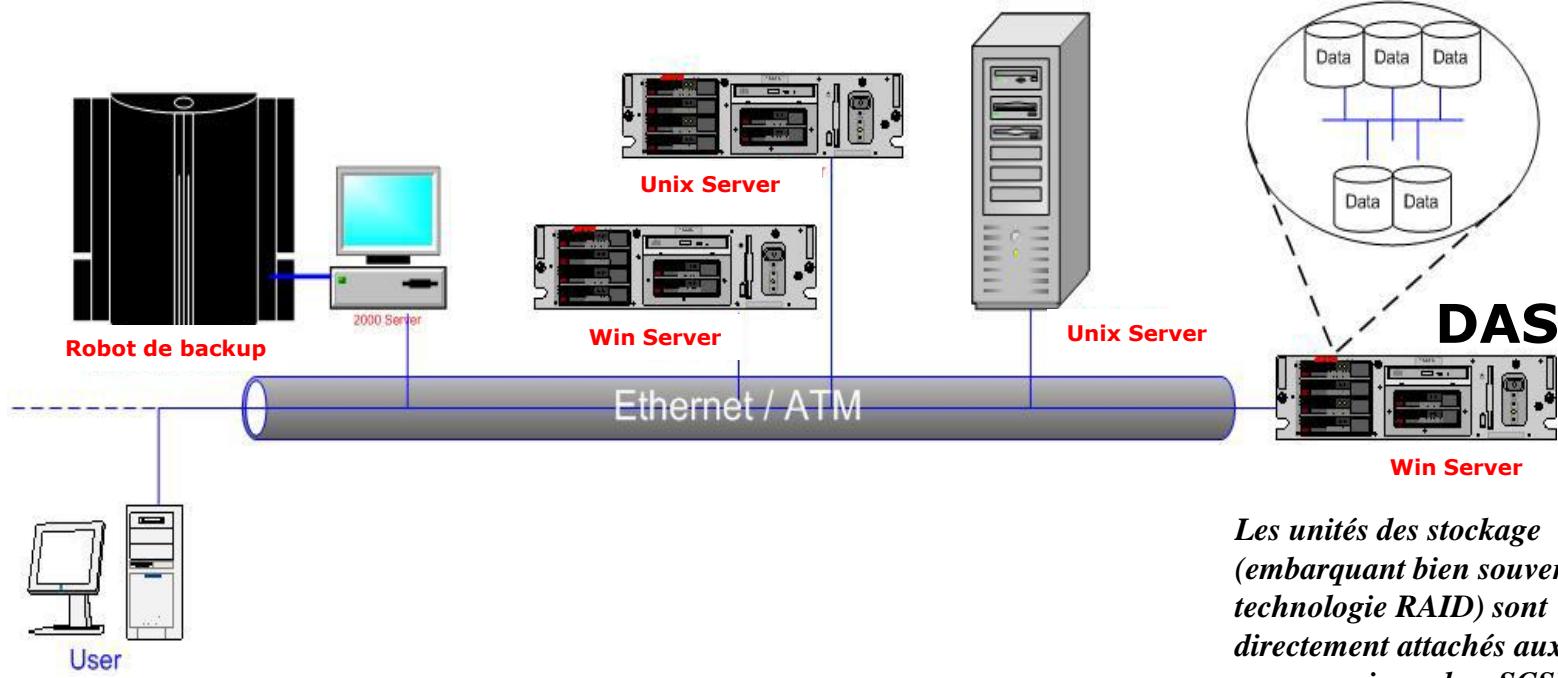
Dell Powervault MD1000



Le RAID

- La technologie DAS (*Direct Attached Storage*)

Architecture



Le RAID

- **La technologie DAS**

Avantages / Inconvénients

-Avantages

- . Le coût reste relativement réduit
- . La simplicité de la mise en place du réseau, de son maintien, et de la configuration du serveur.

-Inconvénients

- . Espace disque des machines figé ou faiblement évolutif (à moins d'un surdimensionnement du LAN)
- . Etranglement du réseau de production s'il est fortement utilisé et si on y ajoute une solution de sauvegarde (robot avec bandes DLT)

→ 2 problématiques:

- Hausse de la demande en espace disque, d'où augmentation des sauvegardes.
- Hausse de la disponibilité des serveurs, d'où réduction du temps de sauvegarde.
Gouwy Jean-Louis

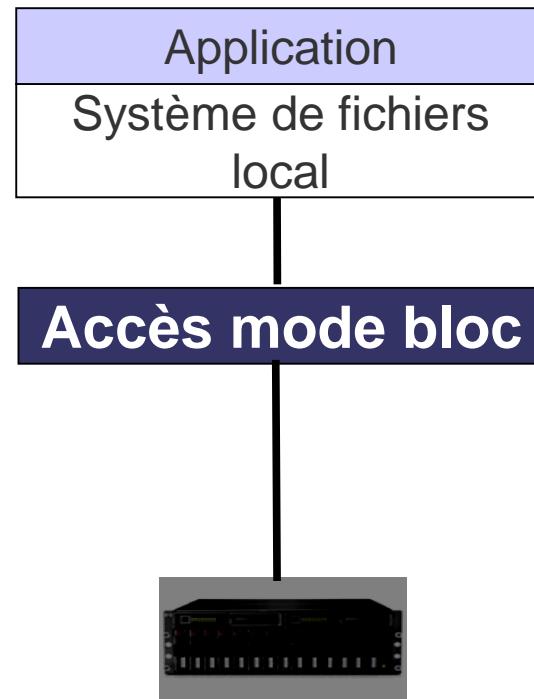


Le RAID

- **La technologie DAS**

Caractéristique

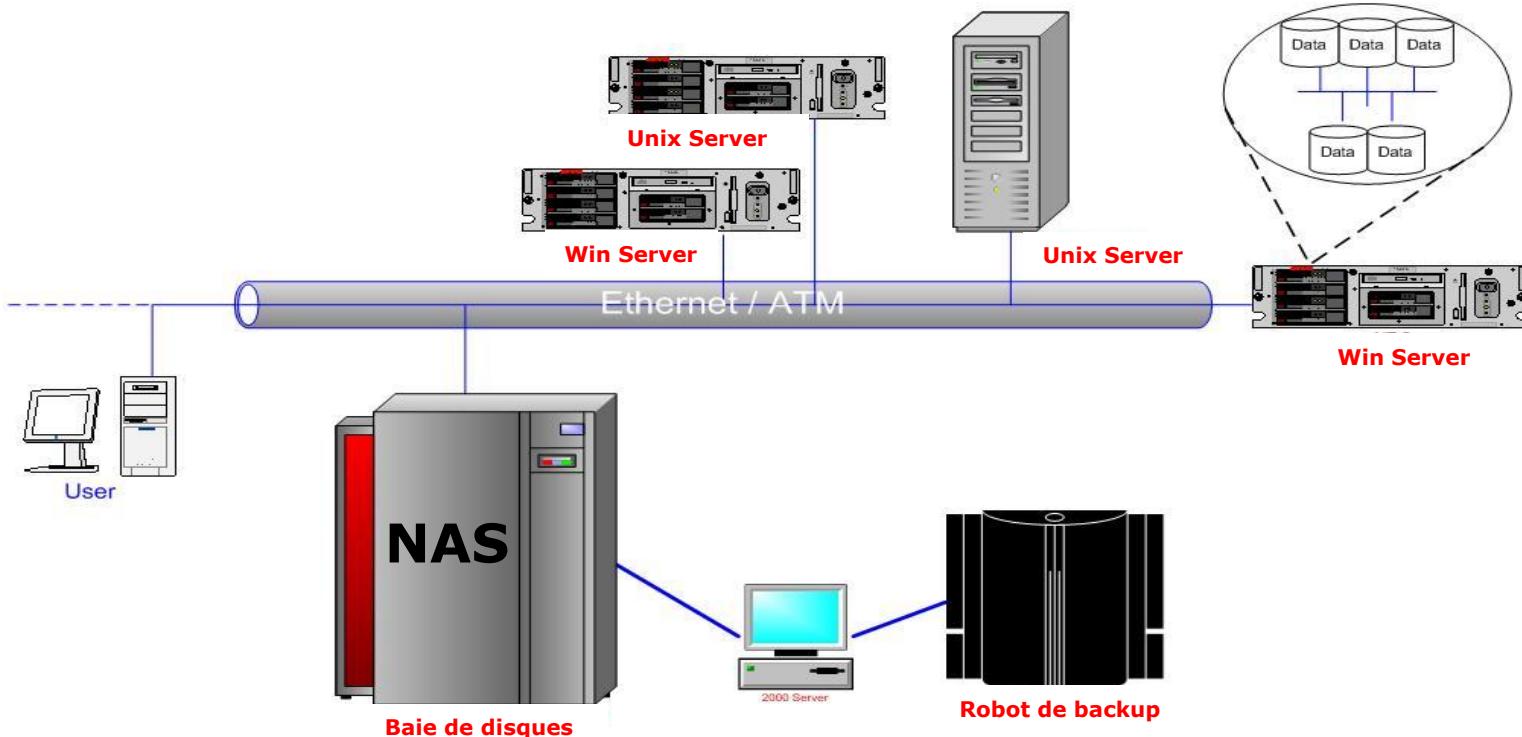
Mode bloc



Le RAID

- La technologie NAS (*Network Area Storage*)

Architecture



Le serveurs NAS s'intègre dans le LAN comme un serveur classique. Il est accessible par n'importe quel type de client (OS).

Placement judicieux du serveur de sauvegarde et de son robot. En effet, si ce serveur est directement rattaché au NAS sans passer par le LAN, la charge induite par les transferts de fichiers lors d'une sauvegarde n'affectera pas le LAN.



Le RAID

• La technologie NAS

Avantages

- . Facilité d'installation sur un LAN sans l'immobiliser.
- . Souplesse dans l'ajout de disque (modification de la capacité de stockage).
- . Simplification du partage de données

Sur des réseaux hétérogènes, chaque client peut accéder au NAS grâce aux divers protocoles supportés par celui-ci:

NFS:	pour les clients Unix/Linux
SMB/CIFS:	pour les clients Microsoft
AFP:	pour les clients Mac

Mais aussi: FTP, HTTP ...

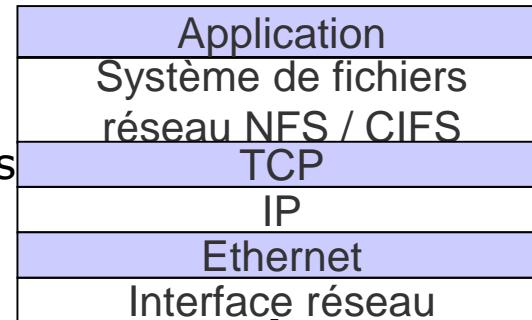


Le RAID

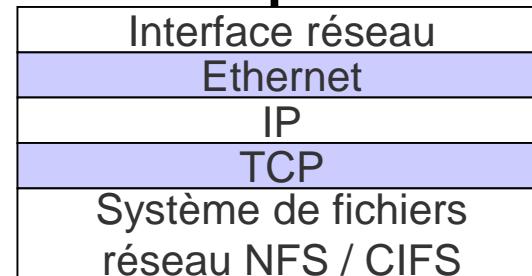
• La technologie NAS

Caractéristique

Mode fichiers



Accès mode fichier



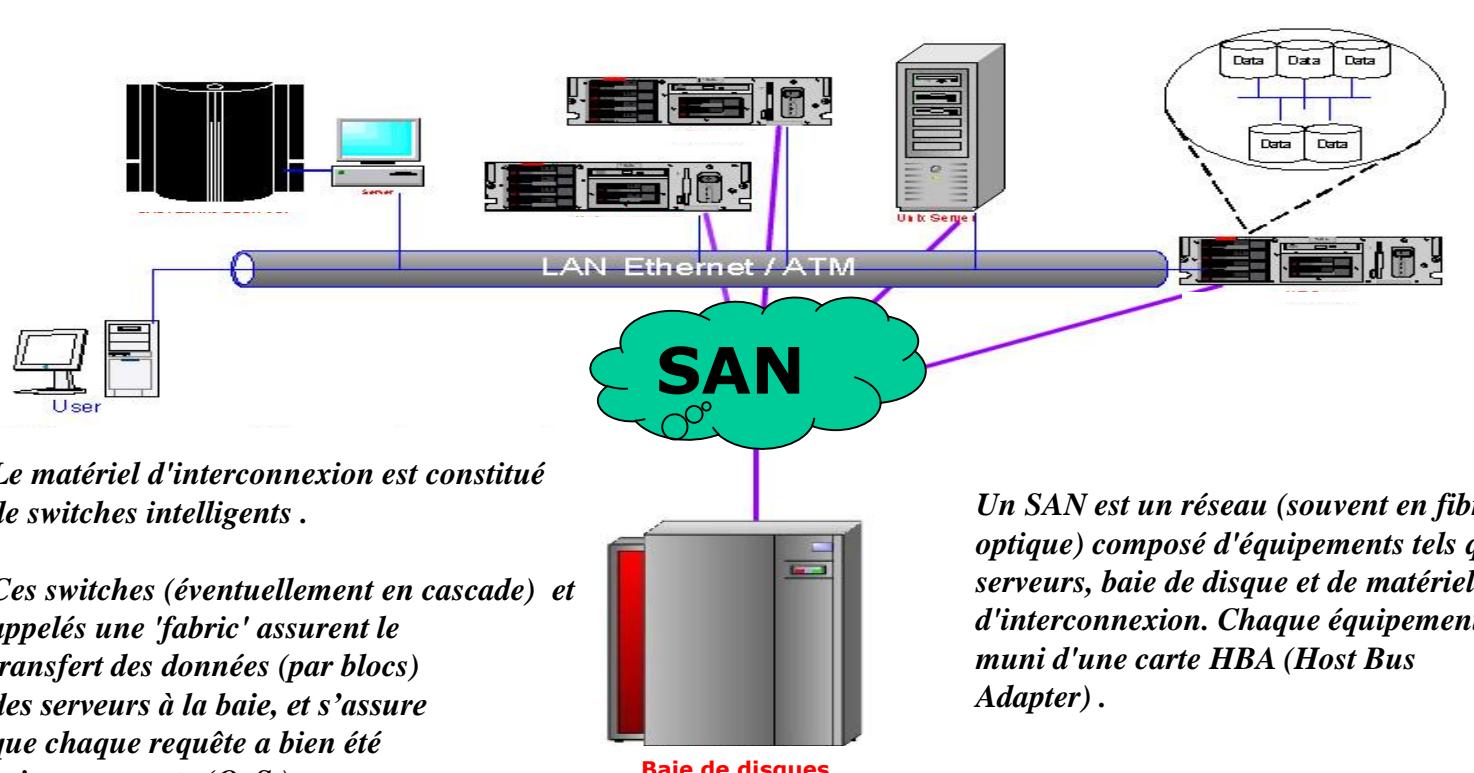
Accès mode bloc



Le RAID

- La technologie SAN (*Storage Area Network*)

Architecture

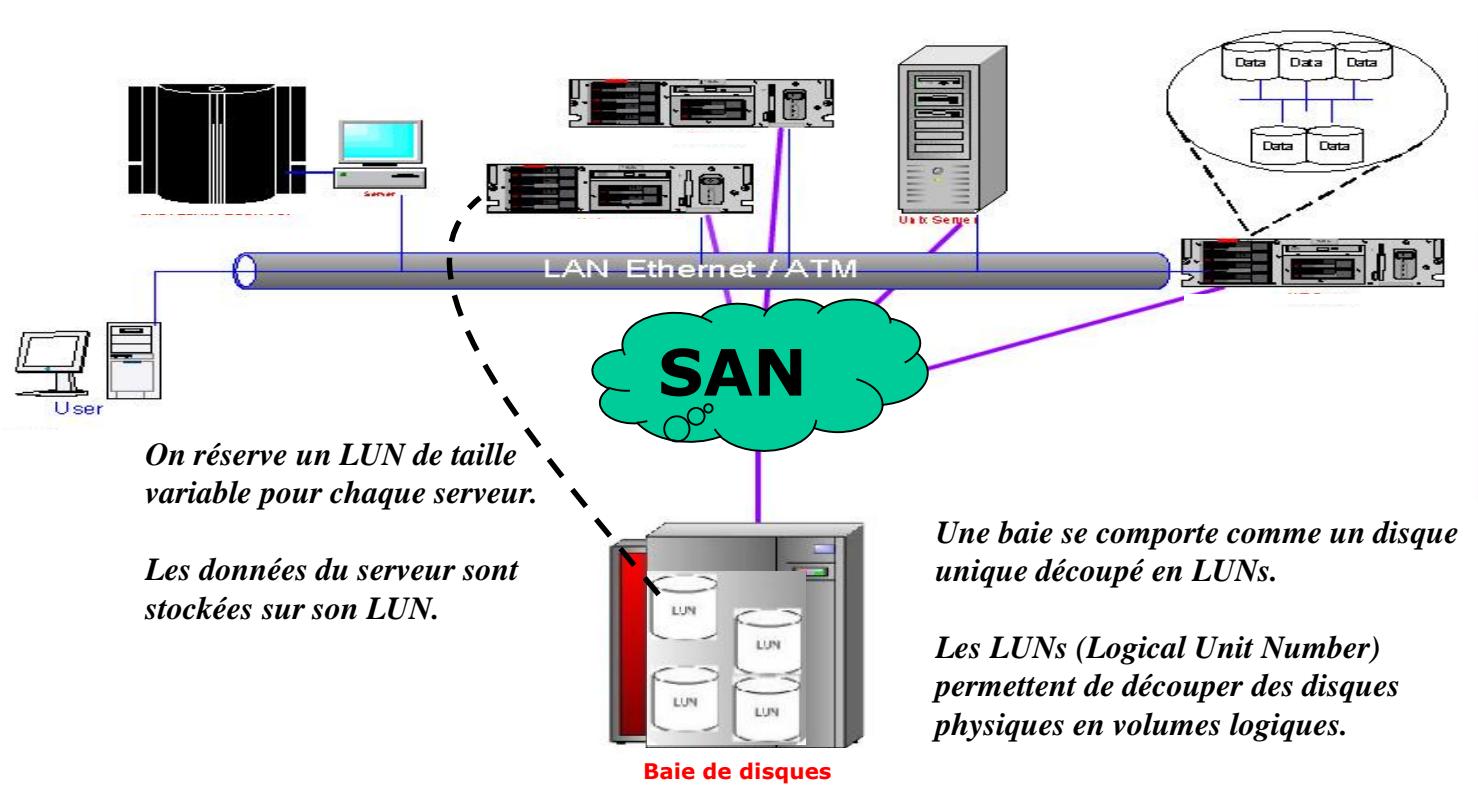


*Les protocoles les plus utilisés:
FC – FCoE - iSCSI*

Le RAID

• La technologie SAN

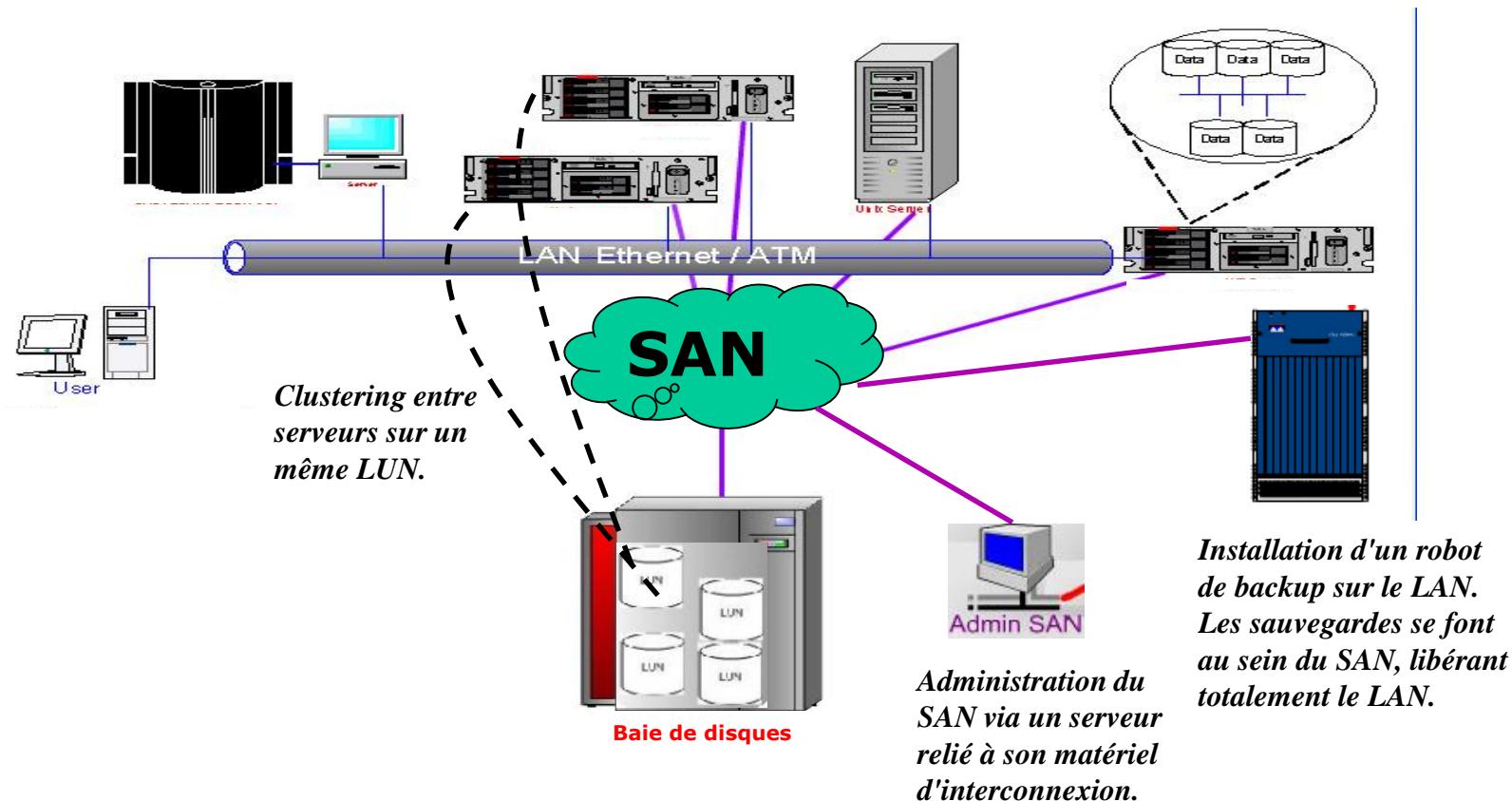
Architecture



Le RAID

- La technologie SAN

Architecture



Le RAID

- **La technologie SAN**

Avantages / Inconvénients

-Avantages

- . Un réel désencombrement du LAN.
- . Des performances importantes.
- . Allocation dynamique de l'espace de stockage entre les serveurs via des outils d'administration.

-Inconvénients

- . Coût élevé de l'installation.
- . Déploiement complexe (souvent en présence d'un expert):

Configuration des switches pour gérer:

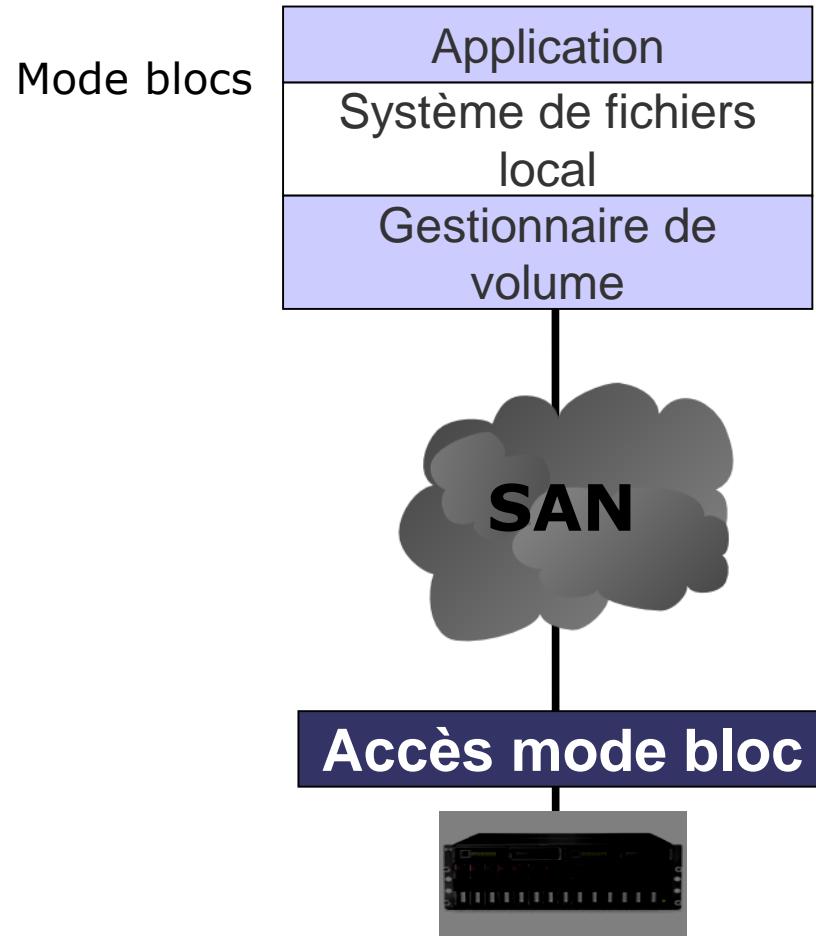
- . le multipath
- . le zoning
-



Le RAID

- **La technologie SAN**

Caractéristique



Le RAID

- **Quelques distributions Linux**

Openfiler

Distribution (basée sur CentOs) dédiée à la mise en œuvre d'un NAS et d'un SAN.

<https://www.openfiler.com>

FreeNAS

Distribution (basée sur FreeBsd) dédiée à la mise en œuvre d'un NAS.

<http://www.freenas.org>



Le RAID

Atelier

Le RAID logiciel



Le RAID

• Références

- <http://www.labo-microsoft.org/articles/web/>
- <http://www.jmax-hardware.com/forum/index.php/topic,3360.0.html>
- http://fr.wikipedia.org/wiki/RAID_%28informatique%29
- <http://phon.pagesperso-orange.fr/phonsite/secu/stockage/raid.htm>
- <http://www.commentcamarche.net/contents/996-protection-les-systemes-raid>
- <https://wiki.archlinux.fr/RAID>
- <http://www.intel.com/support/fr/chipsets/imsm/sb/cs-009337.htm>
- https://raid.wiki.kernel.org/index.php/RAID_setup
- http://lea-linux.org/documentations/Admin-admin_fs-raid
- <http://fr.academic.ru/dic.nsf/frwiki/1399858>
- http://www.intellique.com/fichiers/pres_raid.pdf



GNU/Linux Operating System

STOCKAGE LE LVM

HELHa

Haute École
Louvain en Hainaut



Plan

- **LE LVM**

Intérêt

Concepts de base

Le volume physique (PV) / Le groupe de volumes (VG) / Le volume logique (LV)

Structure des VG, PV, LV

Le physical extend (PE) / Le logical extend (LE)

Avantages

Exemple (Partitionnement en dur - Partitionnement LVM)

Fonctionnalités avancées

Déplacement de volume / Snapshots (Principe – Backup) / Mapping

Atelier: Le LVM



Plan

- **LE LVM**

Le LVM et le RAID (voir Bouchaudy 8.8...)

Le RAID et le LVM (voir Bouchaudy 7.17...)

Atelier: RAID LVM

Références



Le LVM

- **Intérêt**

Un file system (FS) est normalement installé dans une partition d'un disque dur.

- ⌚ Un FS ne peut dépasser la taille d'un disque.
- ⌚ Modification de sa taille difficile car il faut:
 - Sauvegarder les données
 - Repartitionner le disque
 - Restaurer les données



Le LVM

• Intérêt

LVM = Logical Volume Manager

- ☺ Fournit une couche d'abstraction au-dessus des disques physiques.
- ☺ Permet de véritablement "gérer" un ou des espaces de stockage comparé à des partitions fixes.
 - Un FS est créé dans un volume logique.
 - Un volume logique peut s'étendre sur plusieurs disques.
 - La taille d'un volume logique peut être modifiée, ce qui permet de modifier la taille du FS.
- ☺ Meilleure utilisation de l'espace de stockage → indispensable dans des systèmes de stockage centralisé (SAN ...)
- ☹ Plus de couches à gérer: Physical Volume (pv)
 Volume Group (vg)
 Logical Volume (lv)



Le LVM

- **Concepts de base**

Le volume physique (PV: Physical Volume)

- Typiquement une partition (tag 8e) ou un disque entier dont on a effacé le MBR ou un simple fichier grâce au pilote loopback ou un RAID logiciel.

Le groupe de volumes (VG: Volume Group)

- Groupement de un ou plusieurs PV.
- Nommage possible: utile pour les différencier facilement.
- Au final, un VG correspond à un disque physique au sens usuel.

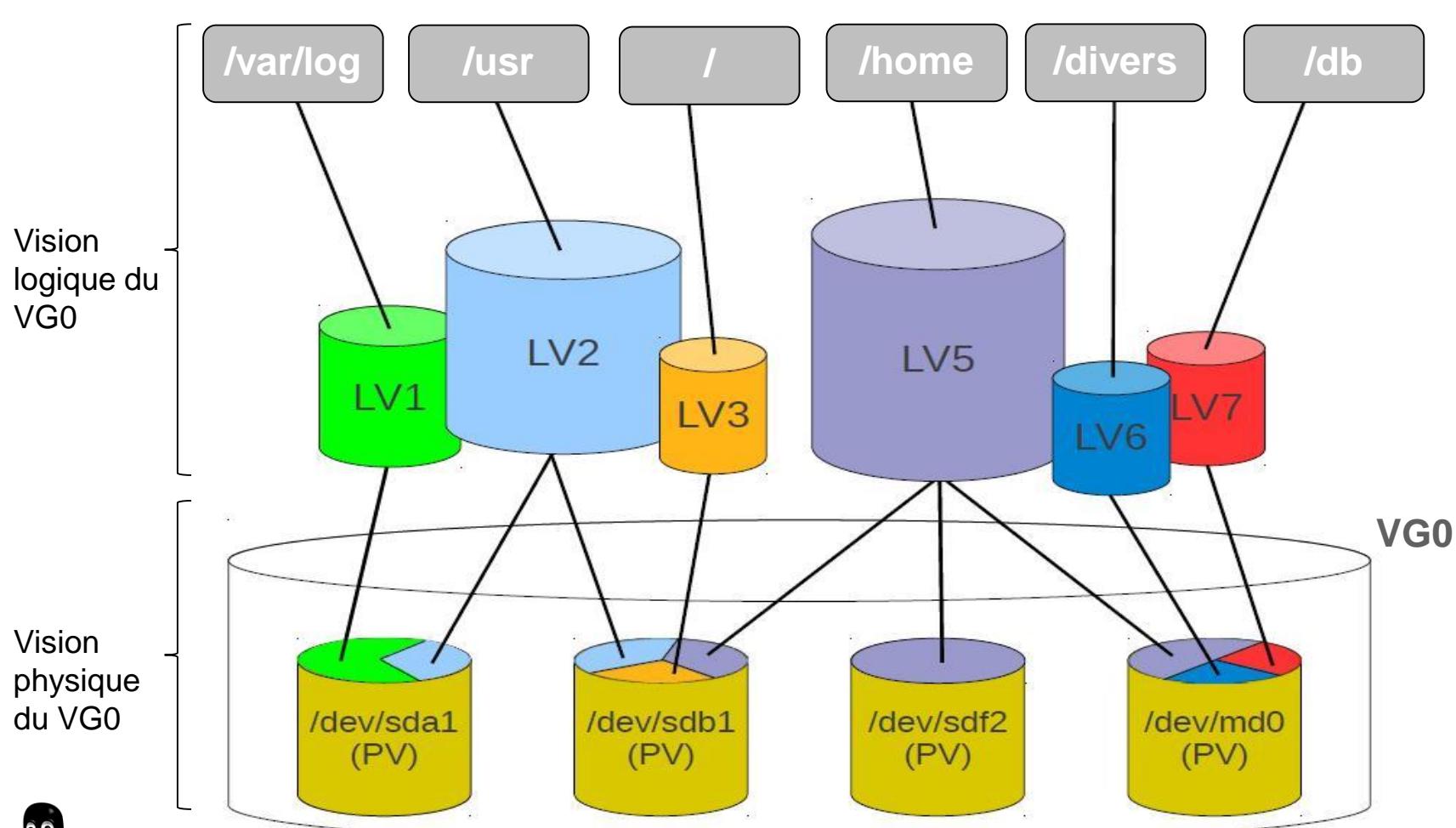
Le volume logique (LV: Logical Volume)

- Un LV dans un VG est comme une partition sur un disque.
- Au final, un LV correspond à une partition (primaire non étendue, ou disque logique) au sens usuel.
- C'est lui que l'on va formater avec le système de fichiers de son choix.
- Possibilité de modifier sa taille au besoin (selon le FS embarqué).



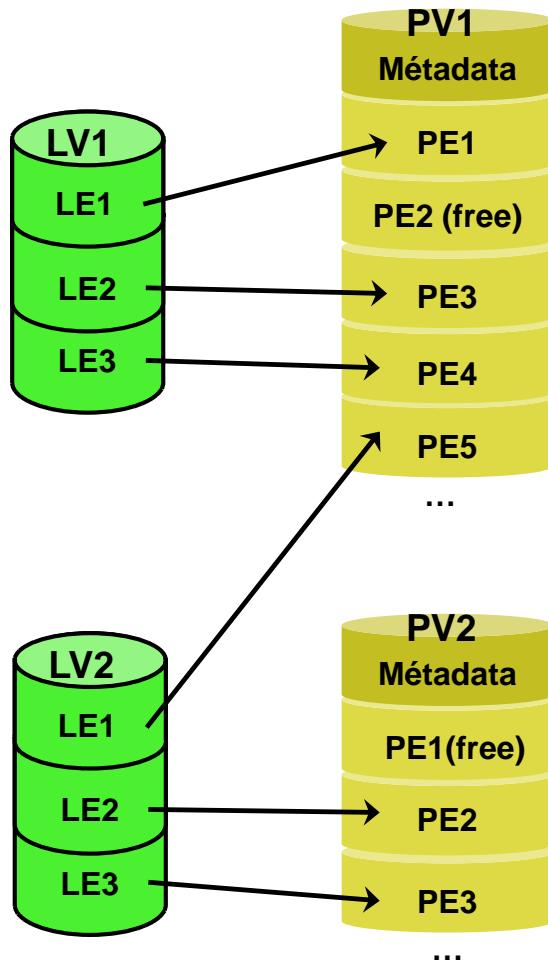
Le LVM

- Concepts de base



Le LVM

• Structure des VG, PV, LV



Le physical extend (PE)

C'est la brique élémentaire contenue dans un PV.
Sa taille est définie à la création du VG (par défaut de 4Mo).

Le logical extend (LE)

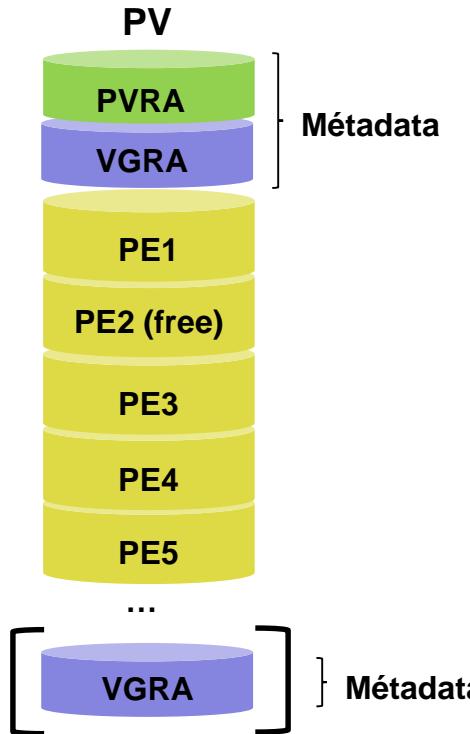
C'est la brique élémentaire contenue dans un LV
Sa taille est la même que le PE qui le supporte.

- ✓ Un PE et un LE ont la même taille et sont l'unité minimale qui peut être gérée dans un système LVM.
- ✓ Chaque LE correspond à un PE sur un PV.
- ✓ Les PE peuvent être alloués à n'importe quel LV mais un PE peut appartenir à seulement un LV à la fois.



Le LVM

- **Structure des VG, PV, LV (suite)**



PV reserved Area (PVRA)

Contient notamment le UUID du PV.

VG reserved Area (VGRA)

Contient une table d'allocation des PE/LE associées à la gestion du VG (un PV contient 0, 1 ou 2 VGRA).

Sauvegarde d'un VG (en 2 temps)

- Sauvegarde des données (FS...).
- Sauvegarde des métadata.

Restauration d'un VG (en 3 temps)

- Création des PV avec les UUID d'origine.
- Restauration des VGRA (le VG est opérationnel mais les LV sont vides).
- Restauration des données (FS...).

Réorganisation des données.

Il est possible de:

- Découper en deux un VG.
- Fusionner deux VG en un seul.
- Déplacer des PE d'un PV à un autre. On peut déplacer tous les PE ou uniquement ceux appartenant à un LV.



Le LVM

• Avantages

- Redimensionnement à chaud (selon le système de fichiers utilisé).

Le LVM offre un grand dynamisme:

- Retaille d'un VG par ajout ou suppression de PV.
- Retaille d'un LV par ajout ou suppression de LE.

Mais en final, il n'y a pas vraiment de dynamisme sur la gestion des données car pour ce faire, il faut:

- Sauvegarder les données.
- Arrêter l'accès aux données (umount ...).
- Retailler le LV.
- Réinstaller les données.
- Redémarrer l'exploitation.

Remarques:

- Sauvegarde inutile dans le cas d'un LV de swap.
- Pour un LV abritant un FS, la plupart des FS peuvent être agrandis. Dans ce cas, il n'est pas nécessaire de les sauvegarder. Par contre, si on veut les rétrécir, il faut le plus souvent opérer selon la recette ci-dessus.



Le LVM

• Avantages

- ❑ Création et suppression de volumes à chaud.
- ❑ Agrégation de disques.
- ❑ Fonctionnalités avancées comme les snapshots, les miroirs, etc..

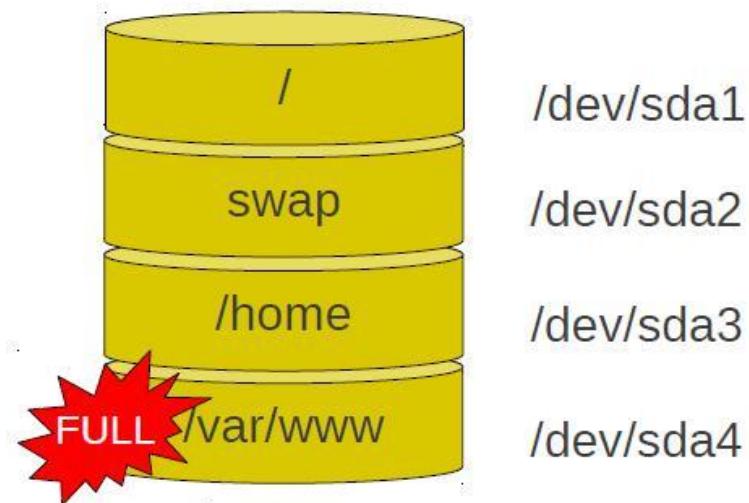


Le LVM

- **Avantages**

Exemple

- ❖ Partitionnement en dur



Pour récupérer de la place, obligation de casser une partition...

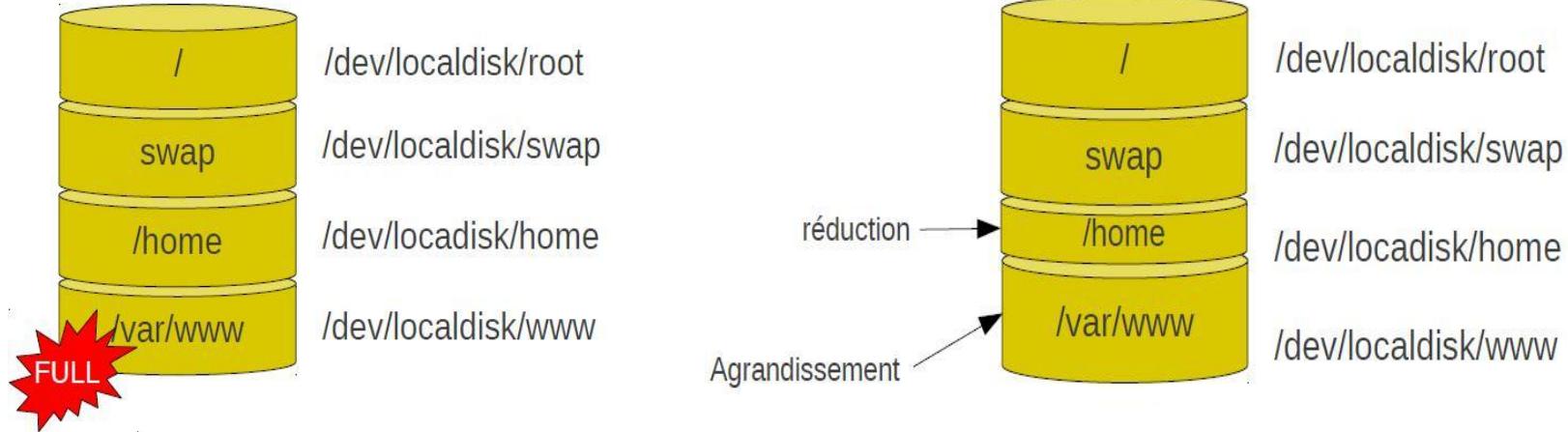


Le LVM

• Avantages

Exemple

❖ Partitionnement LVM



Grâce à LVM, on redimensionne les partitions au besoin...



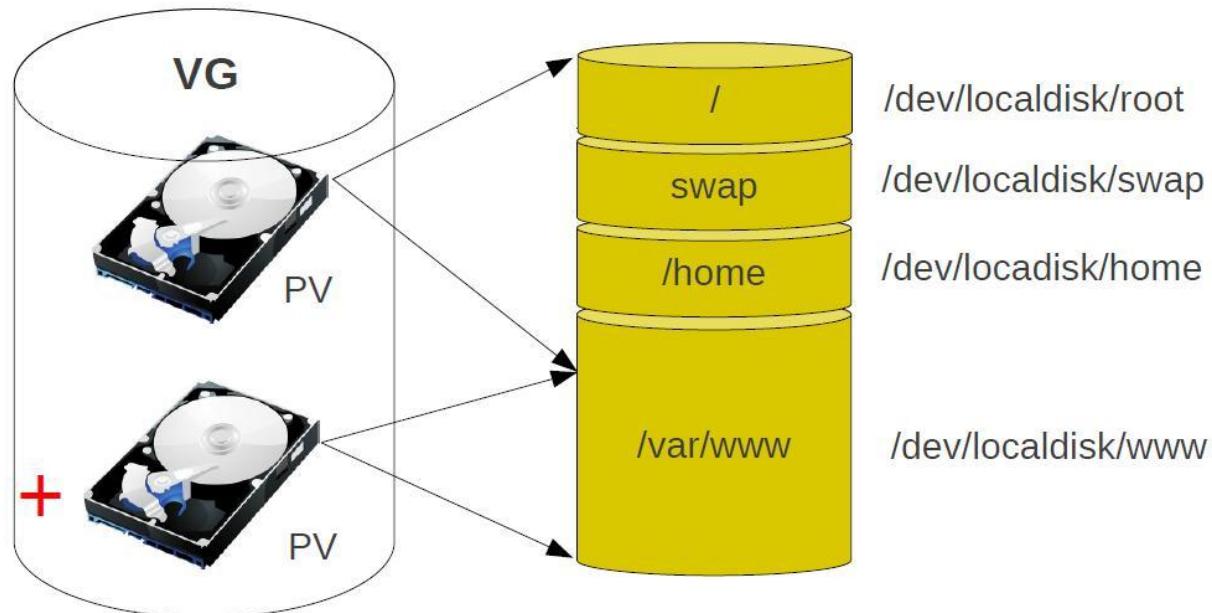
- ☞ Dans cet exemple, le système de fichiers utilisé supporte la réduction !

Le LVM

- Avantages

Exemple

- ❖ Partitionnement LVM



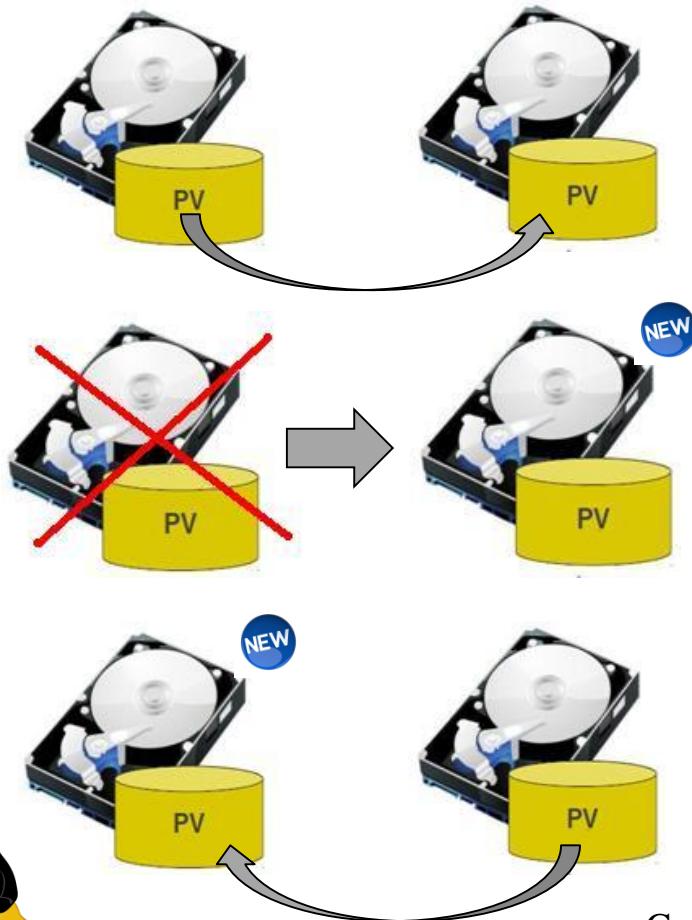
Ici, on ajoute un PV au VG à chaud pour agrandir le LV www...



Le LVM

- Fonctionnalités avancées

Déplacement de volume à chaud



Il devient facile de remplacer un disque défaillant ou aux caractéristiques insuffisantes. Il suffit de déplacer à chaud les données sur un autre disque au sein d'un même volume group.

Le changement du disque, à chaud ou à froid (selon le hardware présent) est alors possible.

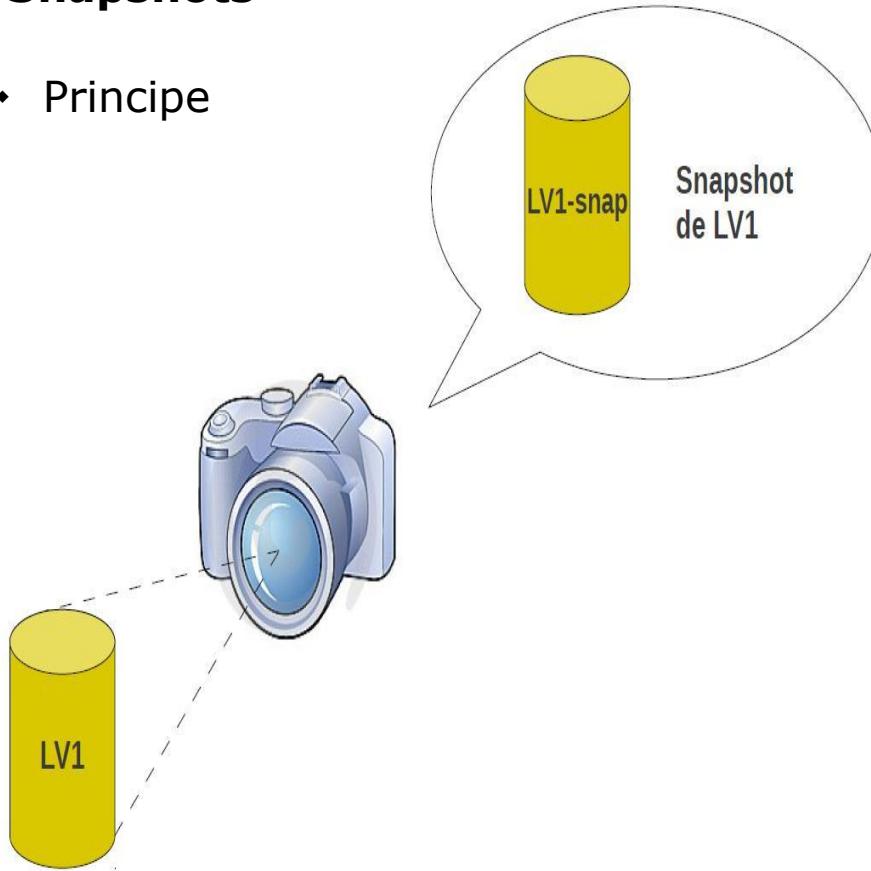
Il ne reste plus qu'à restituer les données par le même biais.

Le LVM

• Fonctionnalités avancées

Snapshots

- ❖ Principe



Les instantanés (*snapshots*) permettent de prendre une photographie d'un LV à un instant donné.

Par la suite, seul le LV1 verra les modifications, et son snapshot restera inchangé (donc cohérent).

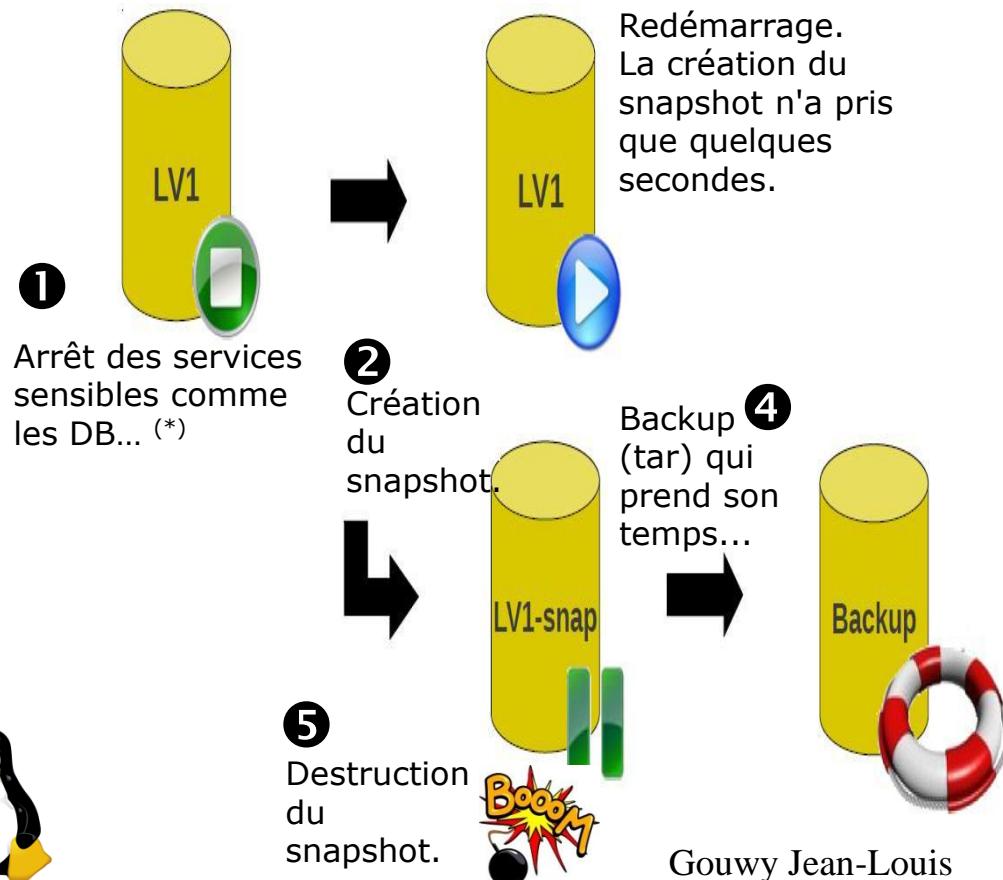
Bien que les snapshots soient considérablement plus petits que les volumes logiques qu'ils sauvegardent, il faut néanmoins disposer d'un minimum d'espace libre dans le groupe de volumes afin d'y créer un nouveau volume contenant le snapshot.

Le LVM

- Fonctionnalités avancées

Snapshots

- ❖ Backup



(*) En production, il n'est jamais conseillé d'arrêter une DB fortement sollicitée en écriture.

Pour réaliser un backup du LV contenant cette DB, il faudrait exécuter le script suivant:

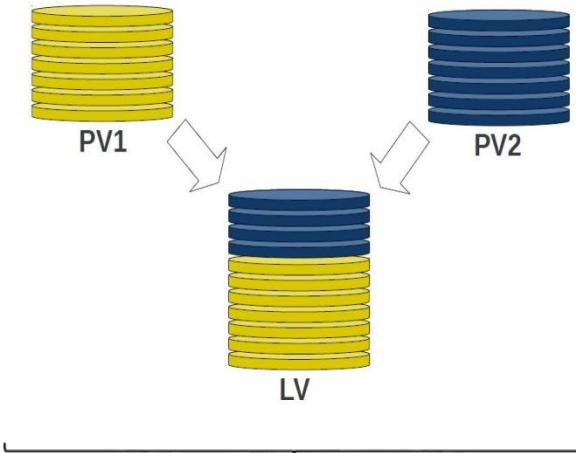
- a) Verrouillage des tables.
- b) Démarrage d'un journal des transactions.
- c) Création du snapshot.
- d) Déverrouillage des tables.
- e) Sauvegarde.
- f) Destruction du snapshot

Le LVM

- Fonctionnalités avancées

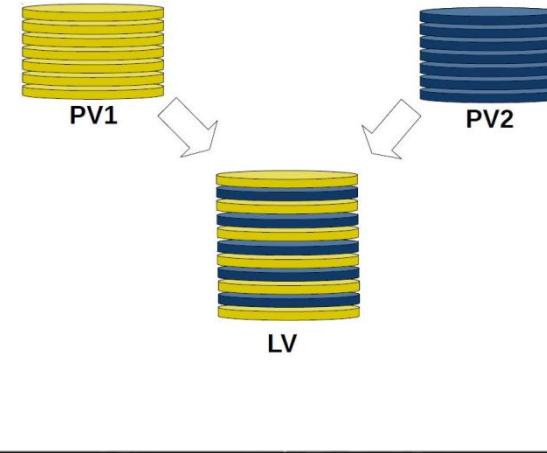
Mapping

- ❖ Linear



Cas le plus courant, les PV sont utilisés à la suite pour la création de nouveaux LV.

- ❖ Stripped



Les PE sont répartis sur les 2 PV (comme un RAID0)



Atelier

Le LVM

