# Administration Système
# Stockage

Lucas Nussbaum
lucas.nussbaum@univ-lorraine.fr

Licence professionnelle ASRALL

*Administration de systèmes, réseaux et applications à base de logiciels libres*

# Introduction

Différents besoins :

- Stocker le système
  - Éventuellement, le système de machines virtuelles
- Stocker les données métiers (des utilisateurs et des services)
- Sauvegardes

Des compromis à trouver entre :

- Performances
- Volumétrie
- Coût
- Fiabilité du support et sécurité des données

# Différents types d'accès

- **Bloc** (*block device*)
  - En local : partition, disque, volume RAID ou LVM
  - En réseau, pour accéder à un périphérique distant
    Protocoles : NBD, iSCSI, AoE (ATA over Ethernet), Fibre Channel
  - Pour stocker un système de fichier, une image disque de VM, etc.
- **Fichier**
  - Implémenté par un système de fichier (répertoires, permissions, . . . )
  - Compatible POSIX $\rightsquigarrow$ utilisation transparente pour les applications
  - Local (stocké dans un *block device*) : ext4, BTRFS, ZFS
  - Réseau, clients/serveur : NFS, CIFS
  - Réseau, réparti sur plusieurs serveurs : Lustre, GlusterFS, etc.
- **Bases de données (SQL)**
- **Objet**
  - Usage Cloud : *manipulation par des méthodes d'accès spécifiques*
  - Plus contraint par POSIX
  - Nombreuses solutions, choix en fonction des besoins
    - ★ Notamment solutions NoSQL : MongoDB, Cassandra, Redis, . . .

# Matériel [1]

- Serveurs classiques
  Exemple : Dell R730XD avec 12 disques de 8 To = 96 To pour 11 400 €
- Serveur + baie attachée via carte RAID (*Direct-Attached Storage*)
  Exemple : Dell PowerVault MD 1400 avec 12 disques 8 To = 96 To pour 9 000 €
- *Appliances* NAS (*Network-Attached Storage*) : solution intégrée, exporte l'espace de stockage via des protocoles standards (NFS, CIFS, iSCSI)
  Exemple : Western Digital My Cloud NAS : 4 To pour 190 €
- SAN (Storage Area Network) = réseau local dédié au stockage, souvent avec une technologie réseau spécifique (Fibre Channel). Réservé aux très gros datacenters
  Exemple : stockage à bandes magnétiques : `https://youtu.be/IDgXa0ioVTs`
- Software-Defined Storage : réseau de serveurs classiques, aggrégation via logiciel qui assure la répartition et une vue unifiée (système de fichiers distribué, stockage objet, etc.)

1. Tous les coûts et toutes les capacités datent de 2016

# Dell R740XD2

Maximum 26 disques, 364 To

# Technologies de disques durs

- Compromis entre capacité, performances, coût
- Disques durs (HDD, *Hard Disk Drive, Hard Disk, Hard Drive*)
  - Disques magnétiques + têtes de lecture (pannes assez fréquentes)
  - Deux formats : 3.5", 2.5" (SFF, serveurs sans gros besoins)
  - Capacité max : dépend du format (3.5" : 10 To ; 2.5" : 4 To)
  - Performances :
    - Dépend de la vitesse de rotation (de 5 400 à 15 000 tr/min)
    - Bon en lecture séquentielle, mauvais en lecture aléatoire
  - Coût plutôt faible :
    - 8 To, 7 200 rpm : 441 €$\rightsquigarrow$ 55 €/To
    - 300 Go, 15 000 rpm : 144 €$\rightsquigarrow$ 480 €/To
- Solid-State Drives (SSD)
  - Électronique, pas de composants en mouvement
  - Performances très bonnes, quel que soit le type d'accès
  - Coût élevé :
    - 400 Go : 300 €$\rightsquigarrow$ 750 €/To
    - 1.6 To : 1 500 €$\rightsquigarrow$ 937 €/To
- De plus en plus : mix SSD + HDD, selon les types de données

# Explorer la topologie : lsblk

```
# lsblk
NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda      8:0    0 558.9G  0 disk
|-sda1   8:1    0   3.7G  0 part [SWAP]
|-sda2   8:2    0  14.9G  0 part /
|-sda3   8:3    0  22.4G  0 part
|-sda4   8:4    0     1K  0 part
`-sda5   8:5    0   518G  0 part /tmp
sdb      8:16   0 558.9G  0 disk
sdc      8:32   0 558.9G  0 disk
`-sdc1   8:33   0 558.9G  0 part
sdd      8:48   0 558.9G  0 disk
`-sdd1   8:49   0 558.9G  0 part
sde      8:64   0 558.9G  0 disk
`-sde1   8:65   0 558.9G  0 part
sdf      8:80   0 186.3G  0 disk
`-sdf1   8:81   0 186.3G  0 part
```

```
# lsblk -tS
NAME ALIGNMENT MIN-IO OPT-IO PHY-SEC LOG-SEC ROTA SCHED RQ-SIZE  RA WSAME NAME HCTL       TYPE VENDOR   MODEL            REV TRAN
sda          0    512      0     512     512    1 cfq       128 128    0B sda  0:0:0:0    disk SEAGATE  ST600MM0088      TT31
sdb          0    512      0     512     512    1 cfq       128 128    0B sdb  0:0:1:0    disk SEAGATE  ST600MM0088      TT31
sdc          0    512      0     512     512    1 cfq       128 128    0B sdc  0:0:2:0    disk SEAGATE  ST600MM0088      TT31
sdd          0    512      0     512     512    1 cfq       128 128    0B sdd  0:0:3:0    disk SEAGATE  ST600MM0088      TT31
sde          0    512      0     512     512    1 cfq       128 128    0B sde  0:0:4:0    disk SEAGATE  ST600MM0088      TT31
sdf          0    512      0     512     512    0 cfq       128 128    0B sdf  0:0:5:0    disk TOSHIBA  PX02SSF020       A4AF
```

# Analyser et mesurer les performances

- Métriques
  - Débit en [ lecture, écriture ] X [ séquentiel, aléatoire (*random*) ]
  - iops (Input/Output Operations Per Second)
- Outils d'analyse
  - `iostat` : statistiques par *device* (`iostat -cdx 1`)
  - `pidstat -d 1`, `iotop` : statistiques par processus
  - `blktrace` : trace des I/O sur un *device* (`btrace /dev/sda`)
  - `strace` : trace les appels systèmes d'un processus
- Outils de mesure de performances
  - `dd` : tests séquentiels basiques, sur des fichiers ou des *devices*
    écriture : `dd if=/dev/zero of=file1 bs=1024k count=1k`
    lecture : `dd if=file1 of=/dev/null bs=1024k`
  - `hdparm` : mesure directe sur un disque (+ effet du cache)
  - `fio` : outil très complet et assez complexe (beaucoup de paramètres, avec une forte influence sur les résultats)

# Utilisation de fio

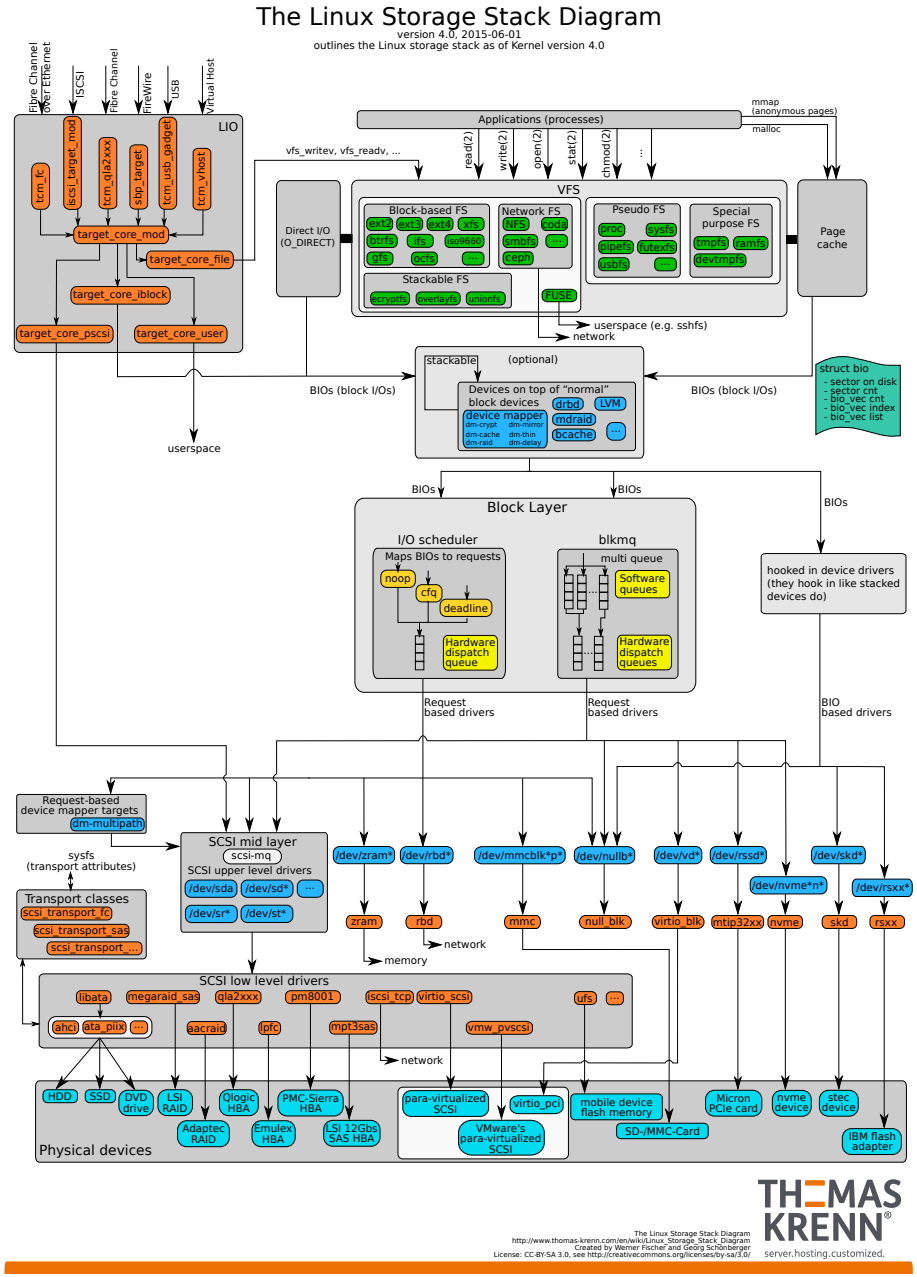
- Exemple : `fio --ioengine=libaio --direct=1 --gtod_reduce=1 --name=test --bs=4k --iodepth=32 --filename=/dev/sde --size=1000M --readwrite=write`
- Agir sur :
  - `--filename=/dev/sde` (fichier ou *device* à tester)
  - `--readwrite` : type d'accès (`read`, `write`, `randread`, `randwrite`)
  - `--size=1000M` : taille du test, affecte la durée et la stabilité (caches)
- Contrôler les caches : `sdparm --get/clear/set=RCD/WCE /dev/sde`
  - RCD = Read Cache Disable (par défaut 0, 1 désactive le cache)
  - WCE = Write Cache Enable (déf : 0 ; ne pas activer si non-ondulé)
- Valeurs typiques : (RCD=0, WCE=0)

| Test | HDD 7200 rpm | HDD 10k rpm | SSD |
|---|---|---|---|
| read, 1000M | 134 Mo/s | 206 Mo/s | 532 Mo/s |
| write, 1000M | 134 Mo/s | 206 Mo/s | 198 Mo/s |
| randread, 200M | 2.7 Mo/s, 681 iops | 4.2 Mo/s, 1056 iops | 532 Mo/s, 132k iops |
| randwrite, 200M | 2.5 Mo/s, 640 iops | 3.1 Mo/s, 781 iops | 198 Mo/s, 50k iops |

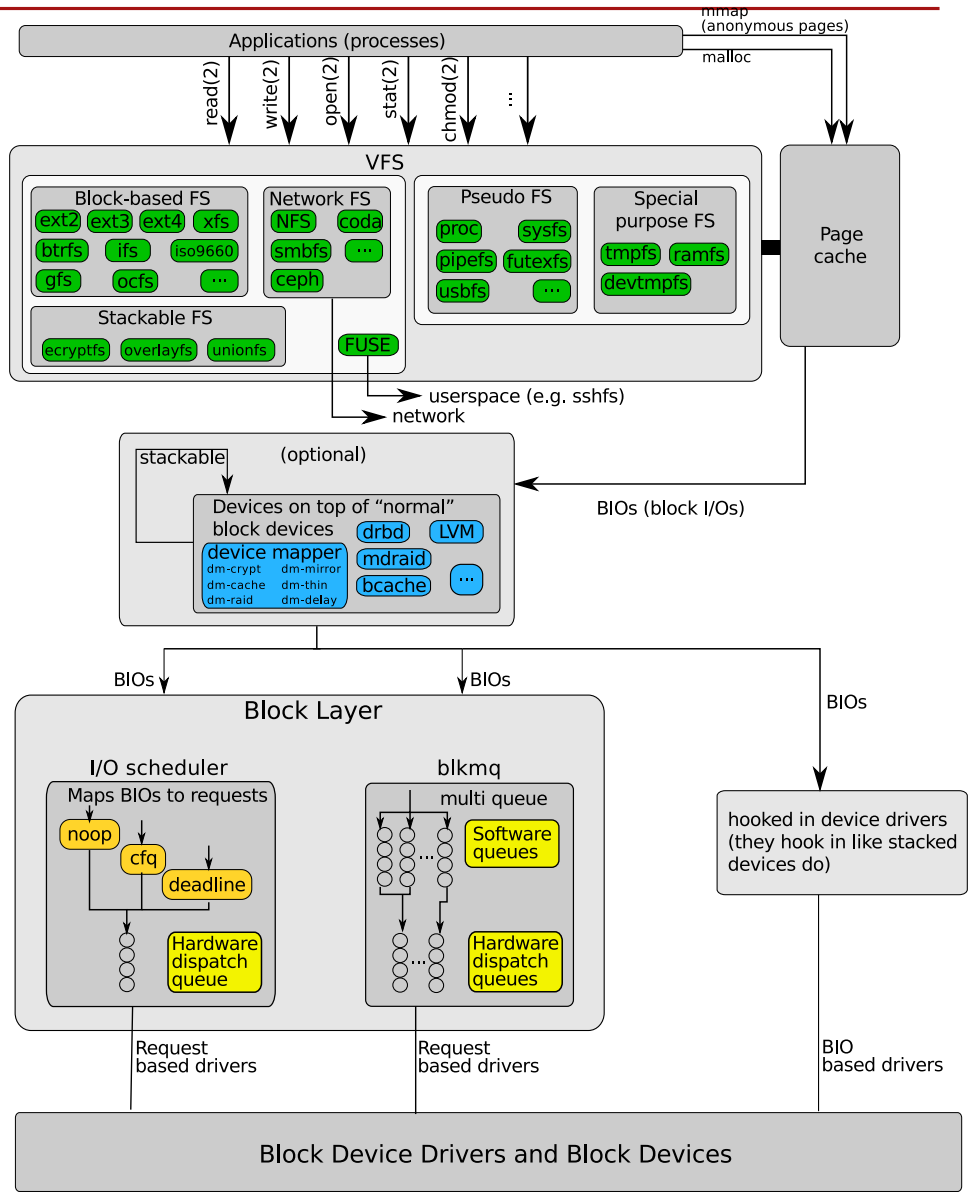
# Dans Linux

- Des systèmes de fichiers : ext4, BTRFS, etc.
- Des drivers de *block devices* pour contrôler les périphériques `/dev/sda`, `/dev/sdb`,...
- Des *block devices* intermédiaires (empilables) :
  - Utilisent un ou plusieurs *block devices*
  - Les agrègent
  - Exportent un ou plusieurs *block devices*
  - Exemples :
    - `mdraid` : RAID logiciel
    - `LVM` : gestion de volumes logiques

# Pile stockage dans Linux



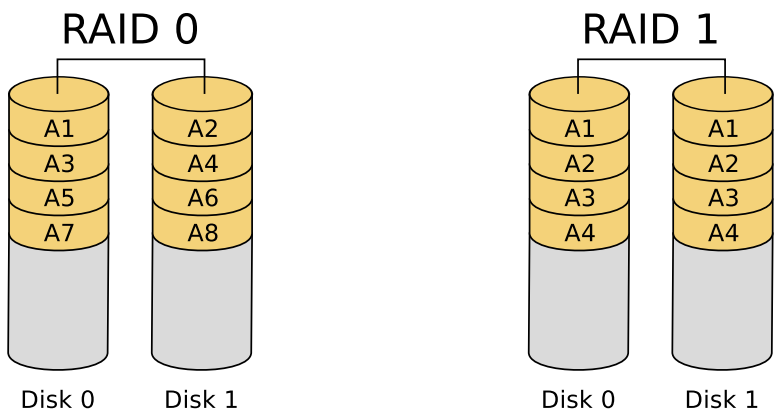
# Pile stockage dans Linux (simplifiée)




Modifié depuis `https://www.thomas-krenn.com/en/wiki/Linux_Storage_Stack_Diagram`

# RAID (*Redundant Array of Independent Disks*)

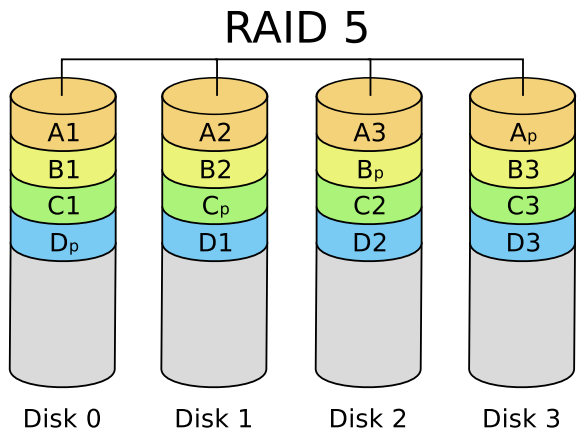
- Agréger plusieurs disques dans un espace de stockage unique pour augmenter la redondance et les performances
- Plusieurs types : RAID 0, RAID 1, RAID 5, RAID 6 (compromis différents)
- Implémentations :
  - RAID logiciel (driver `mdraid`, commande `mdadm`)
    - Utilise le CPU du serveur pour les calculs de parité
    - Plus facile à administrer, à monitorer, etc.
  - RAID matériel (dans le *firmware* de la carte RAID)
    - Performances généralement meilleures (inclut un cache)
- RAID 0 : volume agrégé par bandes
- RAID 1 : disques en miroir

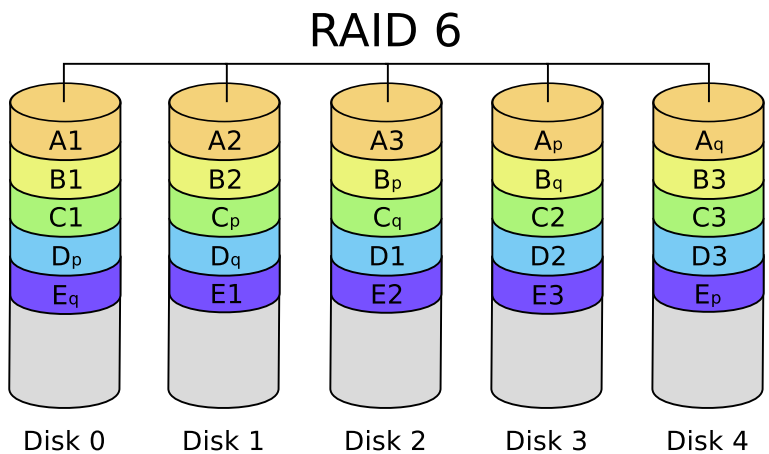



(Source des images : Wikipedia)

# RAID (*Redundant Array of Independent Disks*) (2)

- RAID 5 : volume agrégé par bandes à parité répartie
  $\rightsquigarrow$ Survivre à la défaillance d'un disque
- RAID 6 : idem RAID 5, mais plusieurs disques de parité
  $\rightsquigarrow$ Survivre à la défaillance de plusieurs disques



# LVM (Logical Volume Manager)

- Gérer les *partitions* de manière beaucoup plus flexible
  - Notamment redimensionner à volonté
- On configure des VG (*Volume Group*) qui :
  - Rassemblent des PV (*Physical Volumes*) : disques ou partitions
  - Et les découpent en LV (*Logical Volumes*) : *devices* qui seront utilisés pour créer des systèmes de fichiers, par exemple.
- Configuration la plus classique : un VG comprenant un seul PV (le disque du serveur) et découpé en LV selon les besoins

# LVM (Logical Volume Manager) (2)