

# Laboratoire 3

## INFO3201 - Architecture d'ordinateur

Prof. Andy Couturier

Automne 2024

**Remise : 27 novembre**

### Livrable

Le laboratoire doit être soumis sous forme de rapport via un dépôt Git hébergé sur la plateforme `git.couturier.prof`. Le dépôt doit être intitulé «INFO3201-Laboratoire3».

Seuls les formats de fichiers suivants sont acceptés :

- **OpenDocument Text (ODT)** : Formats créés avec LibreOffice Writer.
- **LaTeX** : Soumettez le fichier source `.tex` et veillez à ce qu'il compile correctement avec `pdflatex` ou `lualatex`. Assurez-vous d'inclure les images et autres fichiers nécessaires à la compilation dans votre dépôt Git.

Points importants à noter :

- **Bonus** : Utiliser LaTeX octroie un bonus automatique de 1% sur la note globale dans la composante devoir/laboratoire du cours, permettant de rattraper des points perdus jusqu'à un maximum de 100%.
- **Inadmissibilité des formats non libres** : Aucun document en formats propriétaires ne sera accepté (note de 0).

### Structure du rapport

Voici la structure proposée pour votre rapport :

- **Page titre**

- Titre du rapport
- Vos noms
- Sigle et nom du cours
- Nom du département
- Nom de la faculté
- Nom de l'université
- Date de soumission

- **Introduction**

- Présentez un bref aperçu de l'objectif du laboratoire, ainsi que son importance.
- L'introduction devrait être en mesure d'expliquer le laboratoire à quelqu'un d'externe qui est hors contexte.

- **Résultats et discussions**

- Inclure ce qui est demandé comme contenu dans les exercices.

- **Conclusion**

- Résumez les principaux apprentissages obtenus du laboratoire.
- Discutez de l'impact des observations sur la compréhension du fonctionnement des modules.

- **Références**

Points importants à noter :

- Vous avez la possibilité d'adapter les titres des sections et la structure du rapport. L'exemple fourni sert uniquement à illustrer le contenu minimal requis, mais vous bénéficiez d'une grande liberté en ce qui concerne le format.
- Si vous décidez d'intégrer le logo de l'université dans votre rapport, assurez-vous de respecter scrupuleusement les normes graphique de l'université (voir ici).

## Exercice 1

Dans cet exercice, l'objectif est de se familiariser avec les concepts du RAID 0 et du RAID 1.

## Test avec un disque unique

Vous pouvez surveiller l'état d'un ensemble RAID en temps réel avec cette commande :

```
sudo watch -n 0.1 "mdadm --detail /dev/md0"
```

Cette commande, au début, affichera de manière répétée une erreur. Toutefois, une fois que vous aurez configuré un ensemble RAID, elle commencera à afficher son statut en temps réel, avec des mises à jour toutes les 0,1 secondes.

Vous pouvez simuler un disque lent en utilisant les commandes suivantes :

```
dd if=/dev/zero of=./slow-disk0.img bs=1M count=100
sudo mknod -m 0660 /dev/loop0 b 7 0
sudo losetup /dev/loop0 ./slow-disk0.img
sudo bash -c 'echo "0 `blockdev --getsize /dev/loop0` delay /dev/loop0 0 50"
→ | dmsetup create dm0'
```

La première commande utilise l'outil `dd` pour créer un fichier d'image de disque se nommant `slow-disk0.img`, lequel est rempli intégralement de zéros. Ce fichier est généré en lisant depuis `/dev/zero`, un dispositif spécial qui produit un flux continu de zéros. La commande spécifie une taille de bloc de 1 mégaoctet (`bs=1M`) et un nombre total de 100 blocs à écrire (`count=100`). Cela donne lieu à un fichier de 100 mégaoctets au total. Le fichier ainsi créé peut ensuite être utilisé pour simuler un disque virtuel ou tout autre type de stockage dans les étapes suivantes du processus.

La deuxième commande utilise `mknod` pour créer un périphérique de type bloc appelé `/dev/loop0`, qui est destiné à être utilisé comme un dispositif en boucle (loop device). Le terme "boucle" dans "dispositif en boucle" découle de la manière dont le système redirige les opérations d'entrée/sortie. En fait, le système d'exploitation fait une sorte de "bouclage" interne : au lieu d'envoyer les opérations directement à un matériel physique, il les redirige vers un fichier sur le système de fichiers. Ce fichier est traité comme s'il était un périphérique physique, créant ainsi un "circuit" fermé où les E/S reviennent à l'origine, c'est-à-dire au fichier lui-même.

La troisième commande fait usage de `losetup` pour attacher le fichier image `slow-disk0.img` au dispositif en boucle `/dev/loop0`. Cette opération permet de traiter le fichier d'image comme s'il s'agissait d'un véritable périphérique de stockage.

La quatrième commande utilise `dmsetup` pour créer un mappage de dispositif nommé `dm0` qui applique un délai artificiel aux opérations d'entrée/sortie effectuées sur le dispositif en boucle `/dev/loop0`. En construisant un tableau de mappage avec la commande `echo`, on spécifie que pour toute l'étendue du dispositif (déterminée grâce à `blockdev --getsize /dev/loop0`), un délai de 50 millisecondes doit être introduit pour chaque opération.

Vous pouvez créer un système de fichier EXT4 sur le disque lent avec la commande suivante :

```
sudo mkfs.ext4 /dev/mapper/dm0
```

Vous pouvez monter le système de fichier avec la commande suivante :

```
sudo mount -o sync /dev/mapper/dm0 /mnt
```

Vous pouvez afficher les appareils de stockage du système et leur structure avec la commande suivante :

```
lsblk
```

Vous pouvez afficher le niveau d'utilisation des systèmes de fichiers montés avec la commande suivante :

```
df -h
```

En utilisant le script bash fourni dans le dossier **Codes** du répertoire du laboratoire, vous pouvez effectuer un test de performance du disque avec ces commandes :

```
chmod +x ./bench_disk.sh # Ajout des permissions d'exécution sur le script  
sudo ./bench_disk.sh /mnt 80M
```

Vous pouvez rediriger la sortie du script dans le fichier **test1** en utilisant la syntaxe suivante :

```
sudo ./bench_disk.sh /mnt 80M > test1
```

Quand vous avez terminé les tests, vous pouvez nettoyer votre environnement avec les commandes suivantes pour vous préparer à la prochaine série de tests :

```
sudo umount /mnt  
sudo dmsetup remove dm0  
sudo losetup -D
```

La commande **sudo dmsetup remove dm0** va retourner une erreur si le disque est encore occupé avec des opérations d'entrées/sorties. Il suffit d'attendre un peu et de relancer la commande jusqu'à ce que ça fonctionne.

## Test avec deux disques en RAID 0

Vous pouvez effectuer un test similaire au précédent avec deux disques lents identiques configurés en RAID 0 en utilisant la série de commandes suivante :

```
# Création du disque lent 0
dd if=/dev/zero of=./slow-disk0.img bs=1M count=100
sudo mknod -m 0660 /dev/loop0 b 7 0
sudo losetup /dev/loop0 ./slow-disk0.img
sudo bash -c 'echo "0 `blockdev --getsize /dev/loop0` delay /dev/loop0 0 50"
→ | dmsetup create dm0'

# Création du disque lent 1
dd if=/dev/zero of=./slow-disk1.img bs=1M count=100
sudo mknod -m 0660 /dev/loop1 b 7 1
sudo losetup /dev/loop1 ./slow-disk1.img
sudo bash -c 'echo "0 `blockdev --getsize /dev/loop1` delay /dev/loop1 0 50"
→ | dmsetup create dm1'

# Création d'un ensemble RAID de type 0
sudo mdadm --create --verbose /dev/md0 --level=0 --raid-devices=2
→ /dev/mapper/dm0 /dev/mapper/dm1

# Création d'un système de fichier sur l'ensemble RAID
sudo mkfs.ext4 /dev/md0

# Montage du système de fichier et affichage de sa capacité
sudo mount -o sync /dev/md0 /mnt
df -h

# Test de performance
sudo ./bench_disk.sh /mnt 80M

# Nettoyage
sudo umount /mnt
sudo mdadm --stop /dev/md0
sudo dmsetup remove dm0
sudo dmsetup remove dm1
sudo losetup -D
```

## Questions

1. Comparez les performances d'un disque unique avec celles d'une configuration RAID 0. Pour ce faire, présentez un tableau indiquant le débit en lecture et en écriture pour

les quatre tests effectués dans le script `bench_disk.sh`. Ensuite, réalisez un second tableau de même structure, mais cette fois-ci montrant les IOPS en lecture et en écriture. Procédez à une analyse de ces résultats et interprétez les différences observées. Le comportement constaté est-il conforme à vos attentes ? Justifiez votre réponse.

2. Réalisez des tests supplémentaires pour des configurations RAID 0 avec 4, 8 et 16 disques. Pour chaque configuration testée, documentez l'ensemble des commandes exécutées. Après chaque test, capturez l'écran affichant les détails de la commande `sudo mdadm --detail /dev/md0`. Assurez-vous également de fournir un fichier de sortie pour chaque test de performance, obtenu via le script `./bench_disk.sh`. Utilisez ensuite les données relatives au débit en lecture pour créer un graphique représentant la variation du débit en fonction du nombre de disques (1, 2, 4, 8, 16). Qu'observez-vous à partir de ce graphe ?
3. Configurez un RAID 1 avec deux disques de 100 Mo chacun et comparez-le à une configuration RAID 0 utilisant également deux disques de la même taille. Quel espace de stockage est disponible pour chaque configuration, et pourquoi ?

## Exercice 2

Dans cet exercice, l'objectif est de se familiariser avec la défaillance de disques dans les ensembles RAID 5 et du RAID 6.

Vous pouvez créer un ensemble RAID 5 simulé avec les commandes suivantes :

```
# Création du disque 0
dd if=/dev/zero of=./raid-disk0.img bs=1M count=100
sudo mknod -m 0660 /dev/loop0 b 7 0
sudo losetup /dev/loop0 ./raid-disk0.img
echo "0 `sudo blockdev --getsize /dev/loop0` linear /dev/loop0 0" | sudo
→ dmsetup create dm0

# Création du disque 1
dd if=/dev/zero of=./raid-disk1.img bs=1M count=100
sudo mknod -m 0660 /dev/loop1 b 7 1
sudo losetup /dev/loop1 ./raid-disk1.img
echo "0 `sudo blockdev --getsize /dev/loop1` linear /dev/loop1 0" | sudo
→ dmsetup create dm1

# Création du disque 2
dd if=/dev/zero of=./raid-disk2.img bs=1M count=100
sudo mknod -m 0660 /dev/loop2 b 7 2
sudo losetup /dev/loop2 ./raid-disk2.img
```

```

echo "0 `sudo blockdev --getsize /dev/loop2` linear /dev/loop2 0" | sudo
→ dmsetup create dm2

# Création du RAID array
sudo mdadm --create --verbose /dev/md0 --level=5 --raid-devices=3
→ /dev/mapper/dm0 /dev/mapper/dm1 /dev/mapper/dm2

# Affichage du status du RAID array en temps réel (à faire dans un autre
→ terminal)
sudo watch -n 0.1 "mdadm --detail /dev/md0"

# Création d'un système de fichier sur le RAID array
sudo mkfs.ext4 /dev/md0

# Montage du disque dans le système
sudo mount /dev/md0 /mnt
df -h

# Simulation de la panne
echo "0 `sudo blockdev --getsize /dev/loop0` error" | sudo dmsetup load dm0
sudo dmsetup resume dm0

# Tentative d'écriture au disque
sudo dd if=/dev/zero of=/mnt/test bs=1M count=100

# Création d'un nouveau disque pour le remplacement
dd if=/dev/zero of=./raid-disk3.img bs=1M count=100
sudo mknod -m 0660 /dev/loop3 b 7 3
sudo losetup /dev/loop3 ./raid-disk3.img
echo "0 `sudo blockdev --getsize /dev/loop3` linear /dev/loop3 0" | sudo
→ dmsetup create dm3

# Ajout du nouveau disque dans le RAID array
sudo mdadm /dev/md0 --add /dev/mapper/dm3

# Retrait du disque défectueux
sudo mdadm /dev/md0 --remove /dev/mapper/dm0

# Nettoyage
sudo umount /mnt
sudo mdadm --stop /dev/md0
sudo dmsetup remove dm0
sudo dmsetup remove dm1
sudo dmsetup remove dm2
sudo dmsetup remove dm3

```

```
sudo losetup -D
```

## Questions

1. Effectuez une expérience de perte d'un unique disque dans une configuration RAID 5, puis une perte de deux disques dans une configuration RAID 6. Pour chaque situation, prenez une capture d'écran affichant les détails fournis par la commande `sudo mdadm --detail /dev/md0` lorsque les disques sont défectueux. Ajoutez ces captures à votre rapport. Imaginez un scénario où un disque dur réel est défectueux. Quelles seraient les étapes à suivre pour résoudre ce problème ? La première étape consisterait à retirer le disque défectueux de l'ordinateur ou du serveur.
2. Effectuez une expérience de perte de deux disques dans une configuration RAID 5. Que se passe-t-il lorsqu'on perd plus d'un disque ? Pouvez-vous toujours écrire de nouveaux fichiers dans `/mnt` ? D'après le cours, existe-t-il une méthode pour récupérer les données de cet ensemble RAID ?