

07 - Optimisation des performances

Objectif

- Montrer qu'une optimisation sérieuse part d'une mesure, puis d'un changement ciblé et vérifiable.

1. Analyse mémoire — /proc

- `/proc` est un système de fichiers virtuel qui expose des informations du noyau.
- Beaucoup d'outils comme `top`, `ps` ou `free` lisent des informations dans `/proc`.
- Pour analyser un processus :

```
cat /proc/<PID>/maps  
cat /proc/<PID>/status
```

- `/proc/<PID>/maps` montre les zones mémoire mappées par le processus.

2. memstat

`memstat` met en forme des informations provenant de `/proc`.

Options utiles :

```
memstat -w  
memstat -p <PID>
```

- `-w` évite la coupure des lignes.
- `-p` cible un processus précis.

3. smem

- `top` peut être trompeur, notamment à cause des bibliothèques partagées et de la mémoire partagée entre processus.
- `smem` expose des valeurs comme RSS, PSS et USS.

```
smem  
smem -p <PID>  
smem --pie name -s rss
```

- Installer `python-matplotlib` permet à `smem` de générer des graphiques.

4. Benchmarks de systèmes de fichiers

- Le cours évoque des comparaisons Phoronix entre ext4, Btrfs, XFS, F2FS et NILFS2 sur SSD.
- Ces benchmarks permettent d'illustrer qu'un système de fichiers peut être excellent dans un scénario et moins bon dans un autre.
- À l'oral, il faut éviter de conclure : « XFS est toujours meilleur » ou « Btrfs est toujours plus lent ».
- La bonne question est : quel workload ? quels fichiers ? quel support ? quelle tolérance au risque ?

5. Choisir le filesystem approprié — Rappels pratiques

- XFS : souvent pertinent pour gros fichiers et stockage serveur.
- ext4 : valeur sûre, stable, généraliste.
- Btrfs : fonctionnalités avancées comme snapshots et compression.
- JFS : faible usage CPU et bon comportement sur certains profils.

6. Tuning des filesystems — atime / noatime

- Linux peut enregistrer le temps d'accès aux fichiers.
- C'est utile pour certains usages, mais cela génère des écritures supplémentaires.
- Désactiver l'enregistrement des accès dans `/etc/fstab` :

```
/dev/hda5 /home btrfs noatime,rw,user,exec,auto 1 1
```

6. Tuning des filesystems — Point d'attention

- Certaines applications peuvent dépendre de `atime`.
- Avant de désactiver, vérifier les usages : messagerie, sauvegardes, outils de conformité, scripts historiques.

7. Taille de bloc

- Une grande taille de bloc peut réduire la fragmentation et accélérer certaines opérations, mais elle peut aussi gaspiller de l'espace avec de nombreux petits fichiers.
- Le cours mentionne 4096 octets comme valeur souvent pertinente.
- Exemple :

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

7. Taille de bloc — Point d'attention

- Changer la taille de bloc nécessite généralement de reformater.

8. xinetd

- xinetd remplace l'ancien inetd/TCP Wrappers dans plusieurs environnements.
- Il peut autoriser ou interdire des connexions selon l'adresse IP, le domaine, l'heure, la charge, le nombre de connexions simultanées ou le débit de connexions entrantes.

8. xinetd — Configuration globale

```
defaults
{
    disable = yes
    instances = 20
    no_access = 0.0.0.0/0
    log_type = SYSLOG authpriv
    log_on_failure = HOST RECORD
    log_on_success = HOST USERID DURATION PID
    per_source = 4
}
includedir /etc/xinetd.d
```

8. xinetd — Configuration par service

```
service telnet
{
    disable = no
    flags = REUSE
    only_from = 172.16.0.0/16
    socket_type = stream
    wait = no
    user = root
    server = /usr/sbin/in.telnetd
}
```

8. xinetd — Point d'attention

- Telnet est un exemple historique, pas une recommandation de déploiement moderne.

9. Optimiser le démarrage — systemd-analyze

```
systemd-analyze  
systemd-analyze blame  
systemd-analyze critical-chain  
systemd-analyze plot > boot.svg
```

- Le graphique SVG aide à visualiser les dépendances et les unités lentes.

9. Optimiser le démarrage — Initramfs et noyau

- Alléger l'initramfs ou compiler un noyau personnalisé peut réduire le temps de démarrage, mais les gains ne sont pas toujours significatifs sur systèmes modernes.

9. Optimiser le démarrage — Démarrage précoce de services

- Certains services peuvent être activés pour démarrer tôt si l'on sait qu'ils seront toujours utilisés :

```
systemctl enable upower
```

9. Optimiser le démarrage — Point d'attention

- Ne pas casser les mécanismes d'activation socket ou D-Bus sans raison : ils existent aussi pour éviter de démarrer inutilement des services.