Optimisation des entrées/sorties pour les systèmes POSIX

Travaux Dirigés/Pratiques

Loïc Tortay (& Éric Legay),

École Informatique IN2P3/Groupe Calcul,

10 février 2012

Heuristiques pour le RAID

- Performances en IOps utile :
  - RAID-1/10:
    - IOps/disque * nombre de disques pour les lectures
    - IOps/*mirroré* pour les écritures
  - RAID-0/5/6/50/60 :
    - IOps/disque * nombre de disques, si I/O > *chunk* ou *stripe*
    - IOps/disque, si I/O < *chunk/strip*
- RAID-[56] & nombre de disques vs *Uncorrectable Bit Error Rate* :
  - BER * #bits/disque * #disques dans un volume RAID

    ⇒ erreur *certaine* avec des gros disques, besoin de vérification de la parité (ou checksum) lors des lectures (DDN, ZFS, Btrfs, ...)
  - RAID-5 habituellement acceptable si nombre de disques < 8
  - RAID-6 si plus de 8 disques ou disques > 500 Go

Heuristiques pour le RAID (2/2)

- Pour le RAID-5/6, préférer les nombres de *disques de **données*** type 2^n ou à la rigueur pairs (+ *disques de parité*), en particulier : 8+P+Q, 8+P, 4+P
- Essayer d'avoir une taille de *strip/chunk* et/ou de *stripe* pertinente, par exemple I/O de 1Mo ⇒ *stripe* de 1 Mo (plus facile à dire qu'à faire)

Exemples & sources

- `/srv/data/posix-io` (README dans chaque répertoire)
  - `ent` : exemples I/Os disque avec les différentes méthodes présentées
  - `hfile` : exemple I/Os disque avec consommation CPU
  - `misc` : meat.c, scpc.c
  - `net` : exemples I/Os réseau, `sendfile` & `select` (*man select_tut*)
  - `rr` : lecture/écriture aléatoires paramétrées
  - `stap` : scripts SystemTap
- Glibc & SUSv3/SUSv4 : `-D_XOPEN_SOURCE=600` mais fonctions certaines Linux indisponibles
- `intelnode`, sous-répertoires de `/mnt` (choisir un parmi les 7) :
  - ext4, xfs (1 x 450 Go)
  - r0-xfs, r0-ext4 (900 Go : 2 x 450 Go)
  - r10-xfs (900 Go : 2 x 2 x 450 Go)
  - r6-ext4 (3.6 To : 8 x 450 Go + P + Q)
  - r5-ext4 (2.5 To : 6 x 450 Go + P)

SystemTap (1)

- `cp -r /srv/data/posix-io ~`
- `cd posix-io ; for d in ent net rr ; do cd $d && make && cd .. ; done`

- Terminal 1 : `# cd ~/posix-io/stap`

  `# stap -v pagecache-hit-rate.stp `id -u $USER``
- Terminal 2 : `% DEST=/mnt/.../${USER}-zeros`

```
dd if=/dev/zero of=$DEST bs=1M count=8192
cd ~/posix-io/rr
/usr/bin/time -p ./rr $DEST
../ent/drop-from-cache $DEST
/usr/bin/time -p ./rr $DEST
```

SystemTap (2)

- Terminal 1 : `# stap -v pagecache-hit-rate.stp `id -u $USER``
- Terminal 2 : `# stap -v seek-seeks.stp `id -u $USER``
- Terminal 3 : `% DEST=/mnt/.../${USER}-zeros`

```
        % dd if=/dev/zero of=$DEST bs=1M count=8192
        % cd ~/posix-io/rr
(A)     % ../ent/drop-from-cache $DEST
        % /usr/bin/time -p ./rr $DEST
```

- Terminal 2 : `Ctrl+C`
- Recommencer en supprimant la ligne (A)
- ?

SystemTap (3)

- Terminal 1 : `# stap -v pagecache-hit-rate.stp `id -u $USER``
- Terminal 2 : `# stap -v seek-seeks.stp `id -u $USER``
- Terminal 3 : `% DEST=/mnt/.../${USER}; mkdir $DEST`

```
        % cd ~/posix-io/ent && cp * $DEST && cd $DEST
(A)     % ./t.sh
        % /usr/bin/time -p ./rr $DEST
```

- Terminal 2 : `Ctrl+C`
- Recommencer, à partir de (A) en remplaçant `./t.sh` par : `./t-drop.sh`
- ???