https://www.coincoin.fr.eu.org/?Tuning-MySQL

Tuning MySQL

- 1- Blog-Notes - Au boulot -

Date de mise en ligne : mardi 24 avril 2007

Date de parution : 24 avril 2007

But de l'article

Il s'agit de fusionner plusieurs instances de bases de données MySQL vers une seule, en modifiant les structures, collations, etc ...

Problématiques rencontrées

– il est impossible de faire un `insert select *` sur plusieurs instances. On parle bien ici d'instances, pas de schéma. L'équivalent de l'Oracle DBLink n'étant utilisable qu'à partir de MySQL 5.2 .
– Après avoir contourné ce point, nous avons été confronté à un état de Deni de Service sur la machine durant les opérations d'instanciation
– De fait des multiples insertions "one shot", nous avons ensuite eu quelques pics d'activité réseau.

Solutions mises en place

1. dblink sous MySQL

L'instance destination que nous appellerons **WEBN est montée en InnoDB** par défaut, pour les besoins transactionnels à venir. Des réplications des instances initiales, que nous appellerons **WEBI, WEBM**, entres autres depuis les masters avant migration sur la machine de destination. Premier test concluant : une réplication d'InnoDB vers **MyISAM** est possible. Un lien symobolique (frm/MYI/MYD) est ensuite réalisé depuis les réplis WEBI et WEBM vers le répertoire data de la WEBN.
– avantages : tous les schémas de tout le serveur sont visibles depuis une seule instances.
– inconvénient, il faut impérativement éteindre les instances, donc couper les réplications, durant les opérations concurrentes sur les même s fichiers, créant des locks filesystem et/ou des incohérences de compteurs.

2. Accélération des insertions de masse

L'opération devant durer un minimum de temps, il convient de traiter environ 60 Go de données, nous adaptons quelques paramètres MySQL :

– **désactivation de l'autoextend InnoDB** : gain en perf de 20 % , on s'affranchit des risques de full disk.

```
innodb_data_file_path          = ibdata1:2000M;ibdata2:2000M; .....
```

– **désactivation des transactions** sur WEBN, on ne fera que des inserts, cette machine n'est pas encore en production. Gain de 20 % en temps d'insertion

```
innodb_flush_log_at_trx_commit = 0
```

– **désactivation des logs de réplication** pour ses futures slaves. De toute façon les ordres ne pourront être

répliquées, les différentes slaves WEBI et WEBM n'étant pas montées. Ce la évitera une montée en I/O inutile.

```
innodb_log_archive              = 0
```

– possibilité non mise en oeuvre mais encore envisagée : **désactivation du cache SQL** (`query_cache_size`)

Après toutes ces opérations, nous avons été confronté à une non tenu des disques. Durant la phase d'insertion, tous les disques étaient au maximum de leur capacité, tous les processus passaient en iowait, la machine était inexploitable. La configuration matérielle était la suivante :

```
Power supply #1 : Ok Power supply #2 : Ok  System        : ProLiant DL380 G4 Processor: 0    : Intel Xeon
: 3200 MHz      : Ok Processor: 1    : Intel Xeon    : 3200 MHz      : Ok Disk Configuration:  Module #:
 1:          512 MB:        Ok Module #:      2:          512 MB:        Ok Module #:      3:          1024
MB:        Ok Module #:      4:          1024 MB:        Ok Module #:      5:          1024 MB:        Ok
Module #:      6:          1024 MB:        Ok  Smart Array 6i in Slot 0    Controller Status: OK    Cache
Status: OK    Battery Status: OK  Slot=0    Total Cache Size: 128 MB  logicaldrive 1 (203 GB, RAID 1+0): Ok
1:2     72.8 GB / 15000 Rps     OK 1:3     72.8 GB / 15000 Rps     OK 1:4     72.8 GB / 15000 Rps     OK 1:5
    72.8 GB / 15000 Rps     OK 2:0     72.8 GB / 15000 Rps     OK 2:1     72.8 GB / 15000 Rps     OK
```

Avec pour conséquence :

```
avg-cpu:  %user   %nice %system %iowait  %steal   %idle           7.23    0.00    2.49   32.42    0.00
57.86  Device:    rrqm/s wrqm/s   r/s   w/s   rsec/s   wsec/s avgrq-sz avgqu-sz await  svctm  %util
cciss/c0d0   0.00 9781.19 17.82 1408.91     0.00     0.00     0.00     5.90    4.37   0.70  99.41
```

– une métrologie précédente a montré qu'il valait mieux multiplier le nombre d'axes de disques plutôt que multiplier les axes logiques. On notera la présence d'un cache disque additionnel et d'un connecteur dual-channel pour répartir les I/O.

```
 Cache Status: OK    Accelerator Ratio: 50/50 (read/write)
```

– Nous adaptons quelques paramètres systèmes :

- méthode d'écriture disque brutale, permanente et identique entre le moteur InnoDB et le filesystem : **OsyncDSync**
  ```
  innodb_flush_method             = O_DSYNC
  ```
- désactivation de l'accounting
  ```
  /dev/cciss/c0d0p7       /BDD            xfs     noatime,osyncisdsync        1 2
  ```
- changement de la méthode par défaut du scheduler : **deadline**
  ```
  cat /sys/block/cciss\!c0d0/queue/scheduler noop anticipatory [deadline] cfq
  ```

Pour l'avoir au boot, cela s'intègre dans la configuration de grub :

```
cat /etc/grub/grub.conf default=0 timeout=10 boot=/dev/cciss/c0d0 title Linux (/boot/vmlinuz-2.6.15.4)
kernel (hd0,0)/boot/vmlinuz-2.6.15.4 ro root=/dev/cciss/c0d0p1 elevator=deadline
```

Malgré tous ces ajouts, la manipulation n'aboutissait toujours pas, pour la même raison que précédemment. Néanmoins, ça "vivait" un peu plus longtemps. Nous touchons au but. Nous choisissons alors d'upgrader la configuration matérielle :

```
Power supply #1 : Ok Power supply #2 : Ok  System        : ProLiant DL380 G4 Processor: 0    : Intel Xeon
```

```
: 3200 MHz      : Ok Processor: 1    : Intel Xeon    : 3200 MHz      : Ok Disk Configuration:  Module #:
 1:          512 MB:        Ok Module #:      2:          512 MB:        Ok Module #:      3:          1024
MB:        Ok Module #:      4:          1024 MB:        Ok Module #:      5:          1024 MB:        Ok
Module #:      6:          1024 MB:        Ok  MSA500 G2 at WEBN_1    Controller Status: OK    Cache Status:
OK    Battery Status: OK  Smart Array 6400 in Slot 1    Controller Status: OK    Cache Status: NotConfigured
   Battery Status: OK  Smart Array 6i in Slot 0    Controller Status: OK    Cache Status: OK    Battery
Status: OK  Slot=1    Total Cache Size: 128 MB     Specified device does not have any logical drives  Slot=0
   Total Cache Size: 128 MB  logicaldrive 1 (203 GB, RAID 1+0): Ok  1:2     72.8 GB / 15000 Rps     OK 1:3
  72.8 GB / 15000 Rps     OK 1:4     72.8 GB / 15000 Rps     OK 1:5     72.8 GB / 15000 Rps     OK 2:0
72.8 GB / 15000 Rps     OK 2:1     72.8 GB / 15000 Rps     OK  chassisname = WEBN_1  logicaldrive 1 (237 GB,
RAID 1+0): Ok  1:1     36.4 GB / 15000 Rps     OK 1:2     36.4 GB / 15000 Rps     OK 1:3     36.4 GB / 15000
Rps     OK 1:4     36.4 GB / 15000 Rps     OK 1:5     36.4 GB / 15000 Rps     OK 1:6     36.4 GB / 15000 Rps
    OK 1:7     36.4 GB / 15000 Rps     OK 1:8     36.4 GB / 15000 Rps     OK 1:9     36.4 GB / 15000 Rps
OK 1:10    36.4 GB / 15000 Rps     OK 1:11    36.4 GB / 15000 Rps     OK 1:12    36.4 GB / 15000 Rps     OK
1:13    36.4 GB / 15000 Rps     OK 1:14    36.4 GB / 15000 Rps     OK
```

Ce qui signifie :

- ajout d'**un contrôleur quad RAID SCSI** avec un cache additionnel de 128 Mo de RAM
- ajout d'**un SAN** de 14 disques en RAID 1+0 avec un cache de 2 x 256 Mo de RAM et un stripsize forcé à 16 ko (la valeur par défaut est de 64 ko). Le firmware est mis à jour également :

  ```
  Hardware Revision: Rev A    Firmware Version: 1.52    Accelerator Ratio: 50/50 (read/write)    Read
  Cache Size: 256 MB    Write Cache Size: 256 MB    Total Cache Size: 512 MB    Battery Backed Cache
  Size: 512 MB    Non Battery Backed Cache Size: 0 MB    Battery Pack Count: 2    Battery Status: OK
  ```

- l'ajout d'un second logicaldrive via un autre port de la carte quad serait inutile, même s'il utiliserait 128 Mo de RAM supplémentaire car : ces 128 Mo sur le second connecteur du SAN ne sont utilisés qu'en spare. Par ailleurs, la carte PCI-Express est unique et on ne gagnerait rien au niveau carte-mère.

Suite à quoi, point de vue logique, nous avions un logicaldrive c0d0 contenant les datas de WEBN ; un logicaldrive c1d0 contenant les datas de WEBI et WEBM entres autres.

Fonctionnalités amusantes, durant les phases de `select insert` on observe des écritures intensives et permanentes sur c0d0, alors qu'à l'inverse, on voit bien un fonctionnement par pic sur c1d0 du fait du plus gros cache disque.

3. Activité réseau

Lors de la mise en place des réplications, 46 slaves (répartis sur un peu plus de 20 switchs) pour une seule master, toutes sur des switchs en GigaBit, il se trouve que le switch sur lequel était branchée la master a délivré prêt de 960 Mo/sec sur un port lorsque le `start slave` a été joué sur toutes les réplis WEBN, laissant peu de place aux 45 autres machines du même équipement réseau ...

https://www.coincoin.fr.eu.org/local/cache-vignettes/L400xH72/net2-426a4.png
Charge Réseau

Afin de palier à cela, nous avons mis en place la **compression MySQL** entre le master et ses slaves :

```
slave_compressed_protocol       = 1
```

Le débit permanent et régulier durant la phase de mise à jour s'est alors stabilisé aux alentours de 120 Mo/sec

https://www.coincoin.fr.eu.org/local/cache-vignettes/L400xH72/network-0c984.png

Charge Réseau

Reporting

https://www.coincoin.fr.eu.org/local/cache-vignettes/L400xH72/innodb-1433c.png
Remplissage InnoDB
https://www.coincoin.fr.eu.org/local/cache-vignettes/L400xH72/cpu-646c4.png
Charge CPU
https://www.coincoin.fr.eu.org/local/cache-vignettes/L400xH78/temp-95175.png
Température CPU
https://www.coincoin.fr.eu.org/local/cache-vignettes/L400xH78/load-35a9c.png
Load I/O
https://www.coincoin.fr.eu.org/local/cache-vignettes/L400xH78/read-ad32d.png
Read I/O
https://www.coincoin.fr.eu.org/local/cache-vignettes/L400xH78/write-52d04.png
Write I/O

Remerciements

– Gérard Corbehem pour son expertise HP
– Laurent Denel pour son expertise RAID
– Olivier Wulveryck pour son expertise MySQL