
lvmcache

Cache LVM

Le type de volume logique cache utilise un LV petit et rapide pour améliorer les performances d'un LV grand et lent. Il stocke les blocks fréquemment utilisé dans le LV rapide. Il est appelé le pool cache, et le grand LV est appelé l'origine. À cause des requis de dm-cache (le pilote kernel), lvm split le LV pool dans 2 périphériques - le LV de données de cache et LV de métadonnées de cache. Le LV de données de cache contient les blocks de données copiées. Le LV de métadonnées de cache maintient les informations d'audit qui spécifie où les blocks sont stockés. Les utilisateurs devraient être familiarisés avec ces LV s'ils souhaitent créer le meilleur LV caché possible. Tous les LV associés doivent être dans le même VG.

Utilisation du cache

La méthode principale pour utiliser un LV de type cache :

0. créer OriginLV

Crée un LV ou identifie un LV origine existant

lvcreate -n OriginLV -L LargeSize VG SlowPVs

Exemple :

lvcreate -n lvol0 -L 100G vg

1. créer CacheDataLV

1. créer CacheDataLV

Crée le LV de métadonnées de cache. Ce LV maintient les blocks depuis OriginLV. La taille de ce LV est la taille du cache.

Exemple :

lvcreate -n Cache0meta -L 12M vg /dev/fast

2. créer CachePoolLV

Combine les 2 LV en un LV pool.

lvconvert -type cache-pool --poolmetadata vg/cache0meta vg/cache0

3. Créer CacheLV

Crée un LV cache en liant le LV pool au LV origine.

lvconvert -type cache --cachepool vg/cache0 vg/lvol0

Suppression du cache

Un LV pool peut être déconnecté d'un LV Cache, laissant un LV pool inutilisé, et un LV origine sans cache :

lvconvert --splitcache vg/cache0

Supprimer un LV pool sans supprimer son LV origine lié :

lvremove vg/CachePoolLV

Une commande alternative déconnecte également le pool du cache, et supprime le pool :

lvconvert --uncache vg/cacheLV

Supprimer un LV cache supprime le LV origine et le LV pool lié

lvremove vg/CacheLV

Tolérance d'erreurs dans un LV pool.

0. Créer un LV origine :

lvcreate -L 10G -n lv1 vg /dev/slow_devs

1. Créer un cache LV data RAID1 :

lvcreate --type raid1 -m 1 -L 1G -n cache1 vg /dev/fast1 /dev/fast2

2. Créer un cache LV métadonnée RAID1 :

lvcreate --type raid1 -m 1 -L 8M -n cache1meta vg /dev/fast1 /dev/fast2

3. Créer un LV pool combinant le LV data et le LV metadata :

lvconvert --type cache-pool --poolmetadata vg/cache1meta vg/cache1

4. Créer un LV cache en combinant le LV pool et le LV origine

lvconvert --type cache --cachepool vg/cache1 vg/lv1

Mode de cache

Le mode de cache par défaut est "writethrough". Ce mode s'assure que toute donnée écrite sera stockée dans le pool et dans l'origine. La perte d'un périphérique associé avec le pool dans ce cas ne signifie pas la perte d'une donnée.

Un autre mode de cache, "writeback" retarde l'écriture des blocs de données du pool dans l'origine. Ce mode augmente les performances, mais la perte d'un périphérique du pool peut résulter de pertes de données.

Avec l'option `--cachemode`, le mode peut être défini en créant un cache, ou changé sur un cache existant.

Exemples

Créer un LV origine :

lvcreate -L 10G -n lv1 vg /dev/slow

Créer un cache LV data :

vcreate -L 1G -n cache1 vg /dev/fast

Créer un cache LV metadata

lvcreate -L 8M -n cache1meta vg /dev/fast

Créer un cache LV pool

lvconvert --type cache-pool --poolmetadata vg/cache1meta vg/cache1

Créer le cache :

lvconvert --type cache --cachepool vg/cache1 --cachemode writethrough vg/lv1

Stratégie de cache

Le sous-système de cache a des paramètres additionnels par LV : la stratégie de cache à utiliser. 3 stratégies sont actuellement disponibles : `smq` est la stratégie par défaut, `m` est une ancienne implémentation, et `cleaner` est utilisé pour forcer le cache à vider toutes les écritures en cache sur l'origine

La stratégie mq a des paramètres. Les défaut sont choisis pour être convenable pour la majorité des systèmes, mais dans des circonstances spéciales, changer ces paramètres peuvent améliorer les performances.

Avec les options `–cachepolicy` et `–cachesettings`, la stratégie de cache et les paramètres peuvent être définis en créant un cache, ou changé dans un cache existant.

Exemple

Changer la stratégie d'un cache existant :

```
lvchange –cachepolicy mq –cachesettings 'migration_threshold=2048 random_threshold=4' vg/lv1
```

Dans `lvm.conf` :

```
allocation/cache_policy  
allocation/cache_settings
```

Taille de chunk

La taille de blocks de données gérés par un pool peut être spécifié avec l'option `–chunksize` à la création. La valeur doit être un multiple de 32KiO entre 32KiO et 1GiO.

Utiliser une taille de chunk qui est trop grande réduit l'utilité du cache. Cependant, une taille trop petite surcharge le travail de gestion des chunks qui sont mappés dans le cache.

Pour afficher la taille de chunk du cache :

```
lvs -o+chunksize VG/CacheLV
```

Dans `lvm.conf` :

```
cache_pool_chunk_size
```

Pour connaître la valeur par défaut :

```
lvmconfig –type default allocation/cache_pool_chunk_size
```

Automatic pool metadata LV

Un LV data peut être convertit au LV pool sans spécifier de LV métadonnée. LVM va automatiquement créer un LV métadonnées depuis le même VG.

Créer un nouveau cache sans un LV origine existant

Un LV cache peut être créé en utilisant un pool sans un LV origine existant. Un nouveau LV origin est créé et lié au pool en une étape `lvcreate –type cache -L LargeSize -n CacheLV –cachepool VG/CachePoolLV VG SlowPVs`

Création de LV pool en une seule étape

Un LV pool peut être créé en une seule commande, au lieu d'utiliser `lvconvert` sur des LV existant :

```
lvcreate –type cache-pool -L CacheSize -n CachePoolLV VG FastPVs
```

Convertir des LV existant en type cache

Quand un LV origine existant est convertis en LV cache, le pool spécifié peut être un LV normal, au lieu d'un LV pool. Dans ce cas, lvm convertis le LV normal en cache. Un LV metadata peut optionnellement être spécifié.

lvcreate -n OriginLV -L LargeSize VG

lvcreate -n CacheDataLV -L CacheSize VG

lvconvert --type cache --cachepool VG/CacheDataLV VG/OriginLV

C'est équivalent à

lvcreate -n OriginLV -L LargeSize VG

lvcreate -n CacheDataLV -L CacheSize VG

lvconvert --type cache-pool VG/CacheDataLV

lvconvert --type cache --cachepool VG/CachePoolLV VG/OriginLV