lvmthin

Provisionning léger lvm

Les blocks dans un volume logique standard sont alloués quand le LV est créé, mais les blocks dans un provisionning léger sont alloués quand ils sont écris. À cause de celà, un LV thin provisioned a une taille virtuelle, et peut être plus grand que l'espace physique disponible. La quantité de stockage physique fournie pour les LV thin provisioned peut être augmentée ultérieurement en cas de besoin.

Les blocks dans un LV standard sont alloués (durant la création) dans le VG, mais les blocks dans un LV thin sont alloués (durant l'utilisation) depuis un "thin pool LV" spécial. Ce pool contient des blocks de stockage physique, et les blocks dans les LV thin référencent simplement les blocks dans le pool.

Un thin pool LV doit être créé avant que les LV thin puissent y être créés. Un thin pool LV est créé en combinant 2 LV standard : Un LV de données qui maintient les blocks pour les LV thin, et un LV de métadonnées, qui maintient les blocks de données associés avec les LV thin.

Les snapshots de LV thin sont inefficaces parce que les blocks de données communs à un LV thin et un de ses snapshots sont partagés. Les snapshots peuvent être plus de LV thin ou d'autre snapshots thin. Les blocks communs à des snapshots récursifs sont également partagés dans le thin pool. Il n'y a pas de limite ou de dégradation des séquences de snapshots.

Vu que les LV thin ou les LV snapshot y sont écris, ils consomment des blocks de données dans le pool. À mesure que les blocks de données diminuent dans le pool, il peut être nécessaire d'en fournir. Cela peut être en étandant le pool. Supprimer les LV thin ou les snapshots du pool peut également libérer des blocks dans le pool. Cependant, supprimer les LV n'est pas toujours une manière efficace de libérer de l'espace parce que la quantité est limité au nombre de blocks non partagés avec d'autres LV dans le pool.

L'allocation de block incrémental de pools thin peut causer les LV à devenir fragmentés. Les LV standard évitent généralement en allouant tous les blocks en une fois durant la création.

Termes Thin

ThinDataLV Grand LV créé dans un VG, utilisé par le pool thin pour stocker les blocks ThinLV

ThinMetaLV Petit LV créé dans un VG, utilisé pour suivre l'utilisation des blocks de données

ThinPoolLV Combinaison de ThinDataLV et ThinMetaLV, contenant les ThinLV et les SnapLV

ThinLV Créé depuis le ThinPoolLV, apparaît blanc après la création

SnapLV Snapshot créé depuis le ThinPoolLV, apparaît comme un snapshot d'un autre LV après la création

Utilisation Thin

La méthode première pour utiliser le provisionning thin est :

1. Créer ThinDataLV Créer un LV qui va maintenir les données du pool thin

lvcreate -n pool0 -L 10G vg0

2. Créer ThinMetaLV Créer un LV qui va maintenir les métadonnées

lvcreate -n pool0meta -L 1G vg0

3. Créer le ThinPoolLV Combiner les données et métadonnées en un LV pool

lvconvert –type thin-pool –poolmetadata vg0/pool0meta vg0/pool0

4. Créer un ThinLV Créer un nouveau LV thin depuis le pool. Ce LV est créé avec une taille virtuelle

lvcreate -n thin1 -V 1T –thinpool vg0/pool0

lvcreate -n thin2 -V 1T –thinpool vg0/pool0

5. Créer un SnapLV Créer des snapshots d'un ThinLV ou SnapLV existant. Ne pas spécifier -L, –size pour créer un snapshot.

lvcreate -n thin1s1 -s vg0/thin1

lvcreate -n thin1s2 -s vg/thin1

lvcreate -n thin1s1s1 -s vg/thin1s1

6. Activer un SnapLV Les snapshots thin sont créés avec le flag "activation skip", indiqué par l'attribut "k". Utiliser -K avec lvchange ou vgchange pour activer les snapshots avec l'attribut "k"

lvchange -ay -K vg0/thin1s1

Syntaxe alternative pour spécifier le type thin-pool

La syntaxe pour créer un pool est :
lvconvert –type -thin-pool –poolmetadata VG/ThinMetaLV VG/ThinDataLV
Un LV existant est convertis en un pool thin en changeant son type en thin-pool. Une syntaxe alternative peut être utilisée pour la même opération :
lvconvert –thinpool VG/ThinDataLV –poolmetadata VG/ThinMetaLV
Le type thin-pool est déduit par lvm ; l'option –thinpool n'est pas un alias pour –type thin-pool. L'utilisation de l'option –thinpool ici est différente de l'utilisation de l'option –thinpool en créant un LV thin, où il spécifie le pool dans lequel le LV thin est créé.

Automatic pool metadata LV

Un LV thin de données peut être convertis en un LV pool thin sans spécifier de LV metadata. LVM créé automatiquement un LV metadata dans le même VG
lvcreate -n pool0 -L 10G vg0
lvconvert –type thin-pool vg0/pool0

Spécifier des périphériques pour les LV de données et de métadonnées

Les LV de données et de métadonnées dans un thin pool sont mieux créés sur des périphériques physiques séparés. Pour cela, spécifie les nom de périphérique à la fin de la ligne lvcreate. Cela peu être utile pour utiliser des périphériques rapides pour les métadonnées
lvcreate -n pool0 -L 10G vg0 /dev/sdA
lvcreate -n pool0meta -L 1G vg0 /dev/sdB
lvconvert –type thin-pool –poolmetadata vg0/pool0meta vg0/pool0

Tolérance aux panne en utilisant RAID

Pour la tolérance aux panne, utiliser un raid pour le pool de données et de métadonnées. C'est recommandé pour les métadonnées
lvcreate –type raid1 -m 1 -n pool0 -L 10G vg0 /dev/sdA /dev/sdB
lvcreate –type raid1 -m 1 -n pool0meta -L 1G vg0 /dev/sdC /dev/sdD
lvconvert –type thin-pool –poolmetadata vg0/pool0meta vg0/pool0

LV metadata spare

La première fois qu'un LV pool est créé, lvm crée un lv metadata space dans le VG. Ce comportement peut être contrôlé avec l'option –poolmetadataspare y|n.

Pour créer le LV pmspare (pool metadata spare), lvm crée d'abord un LV avec un nom par défaut, par ex lvol0, puis le convertit en un LV caché avec le suffix _pmspare, par ex, lvol0_pmspare. Un pmspare est conservé dans un VG à utiliser pour tout pool thin. Le pmspare LV ne peut pas être créé explicitement, mais peut être supprimé explicitement.

Exemple

lvcreate -n pool0 -L 10G vg0
lvcreate -n pool0meta -L 1G vg0
lvconvert –type thin-pool –poolmetadata vg0/pool0meta vg0/pool0

Vérification et réparation des métadonnées

Si un thin pool metadata est endommagé, il peut être réparé. Vérifier et réparer un thin pool metadata est analogue à lancer fsck sur un système de fichier.
Quand un thin pool LV est activé, lvm lance la commande thin_check pour vérifier les métadonnées dans le LV de métadonnées du pool.

lvm.conf thin_check_executable peut être définis avec une chaîne vide ("") pour désactiver l'étape thin_check. Ce n'est pas recommandé.

lvm.conf thin_check_options Contrôle les options de commande utilisées pour la commande thin_check

Si la commande thin_check trouve un problème avec les métadonnées, le LV pool n'est pas activé, et les métadonnées doivent être réparées.

Les commandes de réparation simple ne réussissent pas toujours. La réparation avancée peut nécessiter d'éditer les métadonnée du pool thin et les métadonnées lvm.
lvconvert –repair VG/ThinPoolLV
La réparation effectue les étapes suivantes :

1. **Créer une nouvelle copie réparée des métadonnées** lvconvert lance la commande thin_repair pour lire les métadonnées endommagées dans le LV de métadonnées existant, et écrit une nouvelle copie réparée dans le LV pmspare du VG.
2. **Remplacer le thin pool metadata LV** Si l'étape 1 est réussie, le LV de métadonnées du pool thin est remplacé avec le LV pmspare contenant les métadonnées corrigées. L'ancien devient visible avec le nouveau nom ThinPoolLV_tmetaN (où N est 0, 1, ...).

Si la réparation fonctionne, le LV pool thin et ses LV thin peuvent être activés, et le LV contenant les métadonnées du pool thin endommagé peut être supprimé. Il peut être utile de déplacer le nouveau LV de métadonnées vers un meilleur PV.

Si la réparation ne fonctionne pas, le LV du pool thin et ses LV thin sont perdus.

Si les métadonnées sont restaurées manuellement avec thin_repair directement, le LV de métadonnées du pool peuvent être échangé avec un autre LV contenant les nouvelles métadonnées :

Activation des snapshots thin

Quand un LV snapshot thin est créé, il a le flag "activation skip" par défaut. Ce flag est indiqué par l'attribut 'k' affiché par lvs
lvs vg/thin1s1

Ce flag indique que le LV snapshot n'est pas activé par des commande d'activation normales. Ce comportement ne s'applique pas aux commandes de désactivation.

Un LV snapshot avec l'attribut 'k' peut être activé en utilisant -K (ou –ignoreactivationskip) en plus de l'option standard -ay (ou –activate y)

Commande pour activer un LV snapshot thin :
lvchange -ay -K VG/SnapLV
Le flag persistant "activation skip" peut être désactivé durant lvcreate, ou ultérieurement avec lvchange en utilisant l'option -kn (ou –setactivationskip n). Il peut être activé avec -kv (ou –setactivationskip y).

Quand le flag 'activation skip' est supprimé, les commandes d'activation normales vont activer le LV, et l'option -K n'est pas nécessaire

La commande pour créer un LV snapshot sans le flag :
lvcreate -kn -n SnapLV -s VG/ThinLV
Commande pour supprimer le flag d'un LV snapshot :
lvchange -kn VG/SnapLV

lvm.conf auto_set_activation_skip Contrôle le paramètre d'activation utilisé par lvcreate

Supprimer les LV pool thin, les LV thin et les snapshots

Supprimer un LV thin et ses snapshots associés retourne les blocks utilisés au LV pool thin. Ces blocks sont réutilisés pour d'autres LV thin et snapshots.

Supprimer un LV pool thin supprime le LV de données, et le LV de métadonnées et retourne l'espace au VG.

La suppression de LV pool thin, LV thin et snapshots avec lvremove, ne peuvent pas être récupérés avec vgcfgrestore. vgcfgbackup ne récupère pas les métadonnées du pool.

Gérer manuellement l'espace libre d'un LV pool thin

L'espace disponible dans le LV pool thin peut être affiché avec la commande lvs. L'espace libre peut être ajouté en étendant le LV pool thin.

Commande pour étendre l'espace du pool :
lvextend -L Size VG/ThinPoolLV

Exemple

Doubler l'espace physique d'un pool :
lvextend -L+10G vg0/pool0
D'autres méthodes pour augmenter l'espace dans le pool inclus de supprimer un LV thin et des snapshots associés, ou lancer fstrim dans le système de fichier utilisant un LV thin.

Gérer manuellement l'espace de métadonnées d'un LV pool thin

L'espace de métadonnées disponible dans un LV pool thin peut être affiché avec la commande lvs -o+metadata_percent

Commande pour étendre l'espace de métadonées du pool :
lvextend –poolmetadatasize Size VG/ThinPoolLV

Utiliser fstrim pour augmenter l'espace libre dans un LV pool thin

Supprimer des fichiers dans un système de fichier au dessus d'un LV thin ne rajoute pas généralement d'espace libre dans le pool. Lancer manuellement fstrim peut retourner de l'espace dans le pool qui a été utilisé par des fichier supprimés. fstrim ne fonctionne pas si le LV pool thin a le mode discards à ignore.

Exemple

Un pool thin a 10G d'espace physique, et un LV thin qui a une taille virtuelle 100G. Écrire un fichier 1G dans le système de fichier réduit l'espace libre dans le pool thin de 10% et augmente l'utilisation virtuelle du système de fichier de 1%. Supprimer le fichier de 1G restore les 1% virtuel du système de fichier, me ne restaure pas les 10% physiques. la commande fstrim restore cet espace physique au pool thin.

Étendre automatiquement le LV pool thin

Le service lvm dmeventd (lvm2-monitor) supervise l'utilisation de données des LV pool thin et les étends quand l'utilisation atteind un certain niveau. L'espace libre nécessaire doit exister dans le VG pour étendre les LV pool thin. Superviser et étendre les LV pool thin sont contrôlés indépendamment.

Supervision

Quand un LV pool thin est activé, dmeventd va le superviser par défaut. La commande pour démarrer et stopper la supervision dmeventd d'un LV pool thin :

Le status de supervision actuel peut être affiché avec la commande lvs -o+seg_monitor

autoextension

dmeventd devrait être configuré pour étendre les LV pool thin avant que tout l'espace soit utilisé. Les alertes sont émises via syslog quand l'utilisation d'un pool thin atteind 80%, 85%, 90% et 95%. Le point auquel dmeventd étend un pool, et la quantité sont contrôlés avec 2 paramètres de configuration :

lvm.conf thin_pool_autoextend_threshold % qui défins quand le pool devrait être étendu. À 100 désactive l'extension automatique. La valeur minimum est 50.

lvm.conf thin_pool_autoextend_percent Définis la quantité d'espace à ajouter, en pourcentage de la taille courante.

Désactivation

Il y a plusieurs manières où l'extension des pools thin peut être empêché.

- Si le service dmeventd ne fonctionne pas, aucun monitoring ou extension automatique ne va se produire
- Même quand dmeventd fonctionne, toute la supervision peut être désactivée avec le paramètre de monitoring dans lvm.conf
- Pour activer ou créer un LV pool thin sans interagir avec dmeventd, utiliser l'option –ignoremonitoring peut être utilisé.
- Définir thin_pool_autoextend_threshold à 100 désactive l'extension automatique.

Exemple

Si thin_pool_autoextend_threshold vaut 70 et thin_pool_autoextend_percent vaut 20, quand un pool excède 70% d'utilisation, il est étendu de 20%. Pour un pool de 1G, en utilisant 700M cela déclenche une redimmensionnement à 1,2G. Quand l'utilisation excède 840M, le pool sera étendu à 1,44G, et ainsi de suite.

Épuisement de l'espace

Proprement géré, l'espace de données des pool thin devraient être étendus avant que tout l'espace soit utilisé. Si l'espace est déjà épuisé, il peut être toutefois être étendus.

Le comportement d'un pool thin plein est configurable avec l'option –errorwhenfull y|n de lvcreate ou lvchange. Cette option ne s'applique seulement pour les écritures.

La commande pour changer le comportement d'un pool thin complet :
lvchange –errorwhenfull {y|n} VG/ThinPoolLV

lvm.conf error_when_full Contrôle l'erreur dans le fichier de configuration

Le paramètre courant d'un LV pool thin peut être affiché avec
lvs -o+lv_when_full
Le paramètre errorwhenfull n'affecte pas les paramètre de supervision et autoextension, et ces derniers n'affectent pas non plus le paramètre errorwhenfull. C'est seulement quand la supervision/autoextension ne sont pas effectifs que le pool thin devient plein et que le paramètre errorwhenfull est appliqué.

Le défaut est 'n'. Les écritures des LV thin sont acceptés et mis en queue, dans l'attente d'espace disponible. Une fois l'espace étendu, les écritures en attente sont traitées.

En attendant d'être étendus, le pool thin met les écriture en queue pour 60 secondes (le défaut). Si l'espace n'a pas été étendus après ce délai, les écritures en queue retournent une erreur à l'appelant, par ex le système de fichier. Cela peut résulter en une corruption du système de fichier pour les systèmes de fichier non-journalisés qui peut nécessiter fsck. Quand un pool thin retourne des erreurs d'écriture à un LV thin, un système de fichier est sujet à la perte de données utilisateur non-synchronisés.

Le timeout de 60 secondes peut être changé ou désactivé avec l'option du module kernel dm-thin-pool no_space_timeout.