ext4

4ème système de fichier étendu

Fonctionnalités

Un système de fichier formatté en ext4 peut avoir certains flags de fonctionnalités embarqués :

64bits Active les systèmes de fichiers supérieurs à 2^32 blocks. Cette fonctionnalité est active automatiquement si nécessaire.

bigalloc Active l'allocation de blocks clusterisés, pour que l'unité d'allocation soit une puissance de 2 nombres de blocks. C'est à dire, chaque bit dans le bitmap d'allocation de block indique désormais si un cluster est utilisé ou non, où un cluster est par défaut composé de 16 blocks. Cette fonctionnalité peut diminuer le temps passé à allouer des blocks et réduit la fragmentation, spécialement pour les grands fichiers. La taille peut être spécifiée en utilisant mke2fs -C.

dir_index Utilise des b-trees hashés pour accélérer la recherche de nom dans les grands répertoires.

dir_nlink Autorise plus de 65000 sous-répertoires par répertoire

encrypt Chiffrement au niveau du système de fichier des blocks de données et des noms de fichier. les métadonnées d'inode (timestamps, taille de fichier, user/groupe, etc) ne sont pas chiffrés

ext_attr Active l'utilisation d'attributs étendus.

extent Permet le mappage des numéros de blocks logiques pour un inode particulier vers des blocks physique dans le périphériques de stockage à stocker en utilisant un extent tree, qui est une structure de données plus efficace que le schéma de block indirect traditionnel utilisé par ext2 et ext3. Cela diminue la charge de block de métadonnées, améliore les performances et diminue le besoin de lancer e2fsck sur le système de fichier.

extra_isize Réserve un quantité d'espace spécifique dans chaque inode pour les métadonnées étendues tels que les timestamps nanoseconde et la date de création de fichier, même si le kernel n'a pas besoin d'un tel espace. Sans cette option, le kernel réserve de l'espace quand il en a besoin. Utile quand la taille d'inode fait au moins 256 octets.

filetype Active le stockage des information de type de fichier dans le entrées de répertoire.

flex_bg Permet aux métadonnées de groupes par block (bitmaps d'allocation et tables d'inodes) d'être placés n'importe où. De plus, mke2fs plase ces métadonnées ensemble en commençant au premier groupe de block de chaque "flex_bg group". La taille du groupe flex_bg peut être spécifié avec l'option -G

has_journal Créé un journal qui s'assure de la consistance du système de fichier même entre les arrêt non-propre. Équivalent à utiliser l'option -j avec mke2fs ou tune2fs.

huge_file Permet des fichiers supérieurs à 2Tio

inline_data Permet de stocker des données dans l'inode et dans les zones d'attributs étendus.

journal_dev Active dans le superblock dans un périphérique journal externe. La taille de block pour le journal externe doit être la même que le système de fichier.

large_file Définis automatiquement quand un fichier supérieur à 2Gio est créé.

meta_bg Permet de redimensionner un système de fichiers online sans réserver explicitement de l'espace pour agrandir la taille des descripteurs de groupe de block. Également utilisé pour redimensionner des systèmes de fichier supérieurs à 2^32 blocks. Non recommandé à la création d'un système de fichier.

mmp Fournis une protection de montage multipoint. Utile dans les environnements de stockage partagés

project Fournis le support des quotas de projet. Avec cette fonctionnalité, le project ID de l'inode est géré quand le système de fichier est monté.

quota Créé des inodes quota (inode #3 pour usrquota, #4 pour grpquota) et les définis dans le superblock. Avec cette fonctionnalité, les quotas sont activés automatiquement quand le système de fichier est monté.

resize_inode indique que de l'espace a été réservé pour que la table de descripteur de groupe de block puisse être étendue en redimmensionnant un système de fichier monté. L'opération de redimensionnement online est géré par le kernel, piloté par resize2fs. Nécessite également sparse_super.

sparse_super Indique que les copies backup du superblock et les descripteurs de groupe de block sont présent seulement dans quelques groupes de block, pas tous.

sparse_super2 Indique qu'il y a seulement 2 sauvegardes de superblock et des descripteurs de groupe de block. Les groupes de block utilisés pour les sauvegardes sont stockés dans le superblock, mais typiquement, sera localisé au début du groupe de block #1, et un dans le dernier groupe de block dans le système de fichier. Version extrême de sparse_super et est conçus pour permettre qu'un plus grand pourcentage du disque ait des blocks contigus disponible pour les fichiers.

uninit_bg Indique que les descripteurs de groupe de block sont protégés en utilisant des checksums, permettant à mke2fs de créer un système de fichier sans initialiser tous les groupes de block. Accélère la création des systèmes de fichier.

Options de montage

journal_dev=devnum/journal_path=path Quand les numéros majeur/mineur du périphérique de journal externe a été changé, ces options permettent à l'utilisateur de spécifier le nouvel emplacement du journal.

norecovery|noload Ne charge pas le journal au montage. Noter que si le système de fichier n'a pas été démonté proprement, ne pas rejouer le journal peut engendrer des inconsistances dans le système de fichier.

data={journal|ordered|writeback} Spécifie le mode de journalisation. Les métadonnées sont toujours journalisés :

journal Toutes les données sont envoyées au journal avant d'être écrit dans le système de fichier

ordered mode par défaut. Toutes les données sont écrites dans le système de fichier avant que ses métadonnées soient envoyées au journal

writeback L'ordre des données n'est pas préservé - les données peuvent être écrites dans le système de fichier après que ses métodonnées aient été envoyées au journal. Plus performant et garanti l'intégrité du système de fichier, mais d'anciennes données peuvent apparaître dans les fichiers après un crash et une récupération journal.

commit=nrsec Synchronise toutes les données et métadonnées chaque nrsec secondes. Défaut : 5. 0=défaut.

oldalloc|orlov Utilise l'ancien allocateur ou l'allocateur Orlov pour les nouveaux inode. Orlov est le défaut.

[no]user_xattr Active les attributs étendus

[no]acl Support pour les ACL posix

bsddf|minixdf Définis le comportement pour l'appel système statfs. minixdf retourne dans le champ f_blocks le nombre total de blocks, bsddf soustrait les blocks utilisé par ext4 et non disponible pour le stockage.

debug Affiche des informations de debuggage à chaque remontage

errors={continue|remount-ro|panic} Définis le comportement si une erreur est rencontrée

data_err=ignore|abord Affiche simplement un message d'erreur si une erreur se produit (ignore), ou annule le journal (abort)

grpid|bsdgroups et nogrpid|sysvgroups Définissent quel GID a un fichier nouvellement créé. avec grpid, il prend le GID du répertoire dans lequel il a été créé ; sinon (le défaut), il prend le fsgid du processus courant, sauf si le setgid est mis, auquel cas il prend le gid du répertoire parent et obtient le setgid si c'est un répertoire lui-même.

sb=n Au lieu du block 1, utilise le block n comme superblock. Peut être utile quand le système de fichier a été endommagé.

nouid32 Désactive les UID et GID 32bits. Pour compatibilité avec les anciens kernel.

grpquota|usrquota|quota|noquota active le support des quotas

usrjquota=aquota.user|grpjquota=aquota.group|jqfmt=vfsv0 Support pour les quotas journalisés (quota version 2). jqfmt=vfsv0 active les quotas journalisés. Pour les quotas journalisés les options de montage usrjquota=aquota.user et grpjquota=aquota.group sont requis pour indiquer au système de quota quels bases de quota utiliser. Les quotas journalisés ont l'avantage que même après un crash aucune vérification de quota n'est requis.

journal_checksum Active le cheksum des transactions du journal. Permet au code de récupération dans e2fsck et le kernel de détecter les corruptions dans le kernel.

journal_async_commit Les blocks émis peuvent être écris sur le disque sans attendre les blocks descripteur.

barrier=0 / barrier=1 / barrier / nobarrier Active l'utilisation de barrières d'écriture dans le code jbd. Ces barrières forcent l'ordre sur disque correcte des commits journaux, rendant les cache d'écritures disque volatiles plus sûres, au prix d'une pénalité de performances.

inode_readahead_blks=n contrôle le nombre maximum de blocks de table d'inodes que l'algorithme readahead de table d'inode pré-lit dans le cache. La valeur doit être une puissance de 2. Défaut : 32 blocks

stripe=n Nombre de blocks de système de fichiers que mballoc tente d'utiliser pour la taille d'allocation et l'alignement. Pour les systèmes RAID5/6, c'est le nombre de disques de données * la taille de chunk dans les blocks du système de fichier.

delalloc|nodelalloc Défère l'allocation de block jusqu'au moment de l'écriture

max_batch_time=usec Temps max d'attente pour des opérations de système de fichier additionnels à batcher ensemble avec une opération d'écriture synchrone. Vu qu'une opération d'écriture synchrone force un commit puis attend que l'opération d'E/S se complète, cela ne coûte rien et peut être un gros gain de débit, on attend un peut pour voir si une autre transaction peut se greffer dans l'écriture synchrone. L'algorithme utilisé est conçus pour la vitesse disque, en mesurant la quantité de temps qu'il prend pour finir un commit.

min_batch_time=usec Temps de commit minimum. Défaut : 0ms. Augmenter ce paramètre peut améliorer des charges synchrone multi-threadés sur les disques rapide, au prix d'une latence accrue.

journal_ioprio=prio Priorité E/S (de 0 à 7, 0 étant la priorité la plus élevée) qui devrait être utilisé pour les opérations d'E/S envoyés par kjournald2 durant une opération commit. Défaut : 3, qui est légèrement supérieur à la priorité E/S par défaut.

abort simule les effets de ext4_abort().

auto_da_alloc|noauto_da_alloc De nombreuses applications cassées n'utilisent pas fsync() en remplaçant les fichiers existant via des pattern. Si auto_da_alloc est activé, ext4 détecte les pattern replace-via-rename et replace-via-truncate et force toutes allocations de blocks retardés à être alloués au prochain commit journal, dans le mode data=ordered.

noinit_itable N'initialise pas les blocks de table d'inode non-initialisés en fond. Peut être utilisé par les CD d'installation pour que le processus d'installation puisse se compléter le plus vite possible ; le processus d'initialisation de table d'inode sera déférré à la prochaine fois que le système de fichier est monté

init_itable=n Le code itable attend n fois le nombre de millisecondes qu'il a pris pour remplir de 0 la table d'inode du groupe de block précédent. Celà minimise l'impact sur les performances système à l'initialisation des tables d'inode

discard/nodiscard Contrôle si ext4 devrait émettre des commandes discard/TRIM au périphérique block quand des blocks sont libérés. Utile pour des disques SSD et les LUN sparse/thinly-provisionned.

block_validity/noblock_validity Active la facilité kernel pour traquer les blocks de métadonnées du système de fichier dans les structures de données interne. Cela permet à l'allocateur multi-block et d'autres routines de rapidement localiser les extents qui peuvent chevaucher les blocks de métadonnées. Cette option est prévu pour du debugging.

dioread_lock/dioread_nolock Contrôle si ext4 devrait ou non utilise les locks de lecture DIO. Si l'option dioread_nolock est donné, alloue les extents non initialisés avant l'écriture du tampon et convertis l'extent en initialisé après que les E/S soient complétés. Cette approche permet d'éviter d'utiliser d'inode mutex, qui améliore l'évolutivité sur les stockages à haute vitesse. Cependant cela ne fonctionne pas avec les données journalisée et l'option dioread_nolock est ignorée.

max_dir_size_kb=n Limite la taille des répertoires pour que toute tentative de les étendre au-delà de cette limite en Kio génère une erreur ENOSPC. Utile dans les environnement en mémoire contrainte, où de très gros répertoires peut causer de sérieux problèmes de performances. (par exemple, s'il y a seulement 512Mo de mémoire disponible, un répertoire de 176Mo peut sérieusement monopoliser les ressources système.

i_version Active le support d'inode 64bits. Défaut : off.

Attributs de fichier

a Ajouter uniquement

A Pas de mise à jours du atime

d Pas de dump

D Mise à jours de répertoire synchrone

i immuable

S Mises à jours synchrone

u Non supprimable

j Données journalisées

e Format étendu