

---

# mdmon

## Supervision des métadonnées md externes

Le kernel 2.6.27 ajoute le support des métadonnées externes. Ces métadonnées externes impliquent que l'espace utilisateur gère toutes les mises à jours des métadonnées. La responsabilité du kernel est de notifier l'espace utilisateur quand un événement de métadonnées se produit, tels qu'une erreur disque. Le kernel, dans des cas importants, attend l'espace utilisateur pour prendre une action sur ces notifications.

## Mises à jours des métadonnées

mdmon est un service de mise à jours des métadonnées. mdmon requête l'espace de nom sysfs à la recherche de changements dans les attributs `array_state`, `sync_action` et `state`. Quand un changement est détecté il appelle un handler par type de métadonnées pour effectuer les modifications sur les métadonnées. Les actions sont les suivantes :

**array\_state - inactif** Efface le bit dirty pour le volume et laisse l'array stoppé

**array\_state - write pending** Définit le bit dirty pour l'array puis met `array_state` à actif. Les écritures sont bloquées tant que les écritures de l'espace utilisateur sont actives

**array\_state - active-idle** Le timer safe mode a expiré pour l'array et définit l'état pour nettoyer les écritures de block dans l'array

**array\_state - clean** Efface le bit dirty pour le volume

**array\_state - read-only** État initial de tous les array au démarrage. mdmon prend une des 3 actions suivantes :

1/ Transition de l'array vers read-auto conservant le bit dirty effacé si le handler de métadonnées détermine que l'array n'a pas besoin de resynchro ou autres modifications

2/ Transition de l'array vers actif si le handler de métadonnées détermine qu'une resynchro ou autres manipulations est nécessaire.

3/ Laisser l'array read-only si le volume est marqué à ne pas superviser.

**sync\_action - resync-to-idle** Notifie l'handler de métadonnées qu'une resynchro peut avoir fini. Si un processus de resynchro est en attente avant d'être complété, cet événement permet au handler de métadonnées de vérifier la resynchro

**sync\_action - recover-to-idle** Un spare peut avoir fini la reconstruction donc indique au handler l'état de chaque disque. C'est l'opportunité de l'handler d'effacer les bits out-of-sync et le status dégradé du volume.

**<disk>/state - faulty** Une erreur disque déclenche une série d'événements. D'abord, notifie le handler qu'un disque a échoué, et notifie le kernel qu'il peut débloquer les écritures qui étaient dépendant de ce disque. Ensuite le disque est mis à removed+ de l'array membre. Finalement le disque est marqué en erreur dans tous les autres array membre dans le conteneur.

## Conteneurs

Les formats de métadonnées externe, comme DDF, diffèrent des formats de métadonnées MD natifs dans le sens qu'ils définissent un jeu de disques et une série de sous-array dans ces disques. Les métadonnées en comparaison définissent une relation 1 : 1 entre un jeu de périphériques block et un array RAID. Par exemple, pour créer 2 array à différents niveaux RAID dans un simple jeu de disques, les métadonnées MD nécessitent que les disques soient partitionnés puis chaque array peut être créé avec un sous-jeu de ces partitions. Les formats externes supportés effectuent cette opération disques en interne.

Les périphériques conteneur maintiennent simplement les références de tous les disques membre et permet aux outils mdmon de déterminer quels array actifs appartiennent à quel conteneur. Certaines commandes de gestion d'array comme la suppression ou l'ajout de disque sont seulement valides au niveau du conteneur. Les conteneurs sont identifiés dans `/proc/mdstat` avec une version de métadonnées

---

"external :<metadata name>". Les périphériques membres sont identifiés par "external :/<container device>/<member index>", ou "external :-<container device>/<member index>" si l'array est lecture seule.

## Options

**CONTAINER** Le périphérique conteneur à superviser. Peut être un chemin complet comme /dev/md/container, ou un simple nom de périphérique md.

**-foreground** Ne lance pas en tâche de fond

**-takeover** Instruit mdmon de remplacer un service mdadm déjà en cours d'exécution

**-all** Trouve tous les conteneurs actifs et démarre le monitoring sur chacun d'entre-eux

## Démarrage et arrêt

mdmon doit être lancé quand un système de fichier dans le périphérique monitoré est monté avec des considérations spéciale quand le système de fichier root est monté depuis un périphérique mdmon supervisé. Noter qu'en général mdmon est nécessaire même si le système de fichier est monté en lecture seule. et certains systèmes de fichier restent en écriture dans ces circonstances, par exemple pour rejouer un journal après un arrêt non propre.

Quand l'array est assemblé par le code initramfs, mdadm démarre automatiquement mdmon si requis. Cela signifie que mdmon doit être installé dans l'initramfs et doit il doit y avoir un système de fichier en écriture dans lequel mdmon peut créer un pid et un socket. Le système de fichier particulier à utiliser est donné à mdmon à la compilation. défaut : /run/mdadm.

Une fois le système root instancié avec un pivot\_root, mdmon devrait être lancé avec **-all -takeover** pour que le mdmon lancé dans l'initramfs puisse être remplacé.

À l'arrêt, mdmon ne devrait pas être terminer avec les autres processus. Vu qu'il maintien un fichier ouvert dans /dev, il n'est pas possible de démonter /dev si c'est un système de fichier séparé

## Exemples

Terminer le mdmon en cours d'exécution est terminé et une nouvelle instance est démarré :

**mdmon -all-active-arrays -takeover**