

Etats "HARD" et "SOFT"

La base de données va se fragmenter au fil des insertions/suppression d'éléments, et le volume des données va devenir plus faible que le volume sur disque (dans `/var/lib/mongo`). Il est possible de surveiller cet écart de consommation, et même la réduire.

⚠ Important !

Avant toute opération, faite un shinken-backup complet du serveur impacté ou avec les options qui permet de sauvegarder données

Récupérer l'espace disque

Deux méthodes options existent pour le compactage de la base :

- Compactage in-place de la base
- Faire une sauvegarde de la base et restauration dans une autre base

La première option ne nécessite pas le montage d'une autre base ni de transfert de données. C'est globalement plus simple, mais pendant que la base se compacte elle devient indisponible, ce qui peut provoquer un long temps d'indisponibilité. De plus, suivant le moteur de base utilisé les contraintes et les résultats sont variables:

- **MMapV1**: le compactage sera efficace en termes de récupération d'espace, mais pendant le compactage, le double du volume de données sera utilisé, il faut donc prévoir assez d'espace disque
- **Wired Tiger**: le compactage est moins efficace, le moteur n'arrivant pas à récupérer tout l'espace perdu, mais par contre il se fait in-place sans consommer plus d'espace disque

Il est fortement recommandé de migrer sur le moteur Wired Tiger, qui permet d'avoir de meilleures performances et un espace disque consommé plus faible (moins de fragmentation et compression de données).

- Le script de vérification de la fragmentation permet de donner le moteur de données utilisé.
- Par contre, l'opération nécessitera un arrêt de la base mongoDB, donc vous devrez planifier une interruption de votre supervision.

Méthode 1 : Compactage dans la base

Il est important de noter que pendant que le script va fonctionner, la base se bloquer et les opérations ne pourront pas se faire. Nous vous conseillons de faire cette opération avec Shinken éteint.

Le script de compactage est [launch_mongo_compaction.py](#)

Attention: ce script ne fonctionne que sur un shinken supérieur à la v02.07.00

Il se lance de la manière suivante:

```
launch_mongo_compaction.py -H IP_SERVER --ssh-user=shinken --connection-method=ssh --ssh-keyfile=~shinken/.ssh/id_rsa
```

- **-H**: adresse publique du serveur à vérifier
 - ou localhost si vous souhaitez vérifier le serveur sur lequel s'exécute le script
- **--connection-method=ssh**: à n'utiliser que pour les serveurs distants afin de créer un tunnel SSH vers le serveur et protéger les accès
- **--ssh-user**: utilisateur à utiliser pour les serveurs distants pour les connexions SSH
- **--ssh-keyfile**: clé SSH à utiliser pour les serveurs distants

i ATTENTION: Pour rappel, lancer cette commande va bloquer la base pendant l'opération, qui va dépendre de la vitesse du serveur, notamment de la vitesse de ses disques, car toutes les données vont être lues/réécrites. Il faut que Shinken soit arrêté pendant cette opération.

Au lancement, il va parcourir les bases de données, et va les compacter. Voici une exécution du script:

```
? Unknown Attachment
```

Pour information, la base de données de ce lancement faisait 600Mo sur disque pour un volume de données en base de 1.5Go.

Méthode 2: sauvegarde/restauration de la base dans une autre base de donnée, migration de MMapV1 vers Wired Tiger si nécessaire

i Il est important de noter que pendant la restauration des données, il ne faut pas que Shinken fonctionne. Cette opération doit donc se faire à froid.

C'est également la méthode pour migrer de moteur de données au sein d'un serveur: mongo ne supporte pas de changer de moteur in-place, il faut donc procéder à une restauration de données dans une nouvelle base avec le nouveau moteur.

Afin de sauvegarder la base, il faut prévoir suffisamment d'espace disque. Il faut ensuite lancer:

```
shinken-backup --addons --configuration --sla --user
```

Restauration sur un nouveau serveur

Il faut ensuite transférer la sauvegarde sur le nouveau serveur et lancer la restauration:

```
shinken-restore
```

Restauration sur le même serveur

Si vous souhaitez réutiliser le même serveur, il va falloir:

- Arrêter mongo
- Supprimer ses données
- S'assurer que le format de données est bien WiredTiger
- Restaurer les données

Arrêter MongoDB:

```
/etc/init.d/mongod stop
```

Supprimer les données:

```
rm -rf /var/lib/mongo/*
```

Migration vers Wired Tiger (si nécessaire)

Pour s'assurer que le format de données est WiredTiger, il faut que l'entrée suivante soit dans le fichier /etc/mongod.conf:



Le format du fichier `mongod.conf` doit être le nouveau format de mongo

ATTENTION: le fichier `/etc/mongod.conf` doit être dans le nouveau format de configuration (du yaml). Si la chaîne `wiredTiger` n'est pas présente (et commentée) alors c'est que vous avez l'ancien format, et devez migrer vos paramètres.

Pour ce faire, vous avez le fichier `/etc/mongod.conf.rpmnew` qui est présent avec le nouveau format, vous n'avez qu'à:

- sauvegarder votre ancien fichier
- copier le fichier `/etc/mongod.conf.rpmnew` en `/etc/mongod.conf`
- reportez vos paramètres spécifiques si vous en aviez modifiés

```
storage:  
  engine: wiredTiger
```

Puis relancer la base:

```
/etc/init.d/mongod start
```

Ensuite il faut lancer l'import, qui va se faire avec le nouveau format de données, et avec un espace disque consommé minimal:

```
shinken-restore
```