

Graphite: définition de la précision du stockage des données

Les précisions de stockages

Graphite, comme toutes les bases de métrologie, stocke ses données avec diverses précisions: plus les données sont éloignées, moins elles sont précises, et ce afin de gagner en performance et surtout espace disque.

A noter que chaque métrique (par exemple `cpu_idle` sur le check Cpu de l'hôte serveur-1) aura son propre fichier `.wsp` présent dans `/opt/graphite/storage/whisper` (dans un sous répertoire HOST-UUID/CHECK-UUID).

Définition des précisions de stockage

La définition des différentes précisions se fait via le fichier `/opt/graphite/conf/storage-schemas.conf`.

Il va s'organiser en bloc. Chacun va avoir:

- **pattern**: une règle d'expression régulière d'application du bloc.
 - **Attention**: vu que les métriques sont stockées sous forme de uuid dans Graphite, seuls les noms de métriques seront utilisables ici, pas les noms d'hôtes ou de checks
- **priority**: une priorité d'application du bloc, plus la valeur est haute, moins il est prioritaire
- **retentions**: une règle de rétention: suite de couple *précision:durée*, qui donne la précision d'un point, et la durée de stockage pour cette précision.

Précision livrée par défaut à partir de la V02.06.02

La précision livrée par défaut depuis la 02.06.02 est la suivante:

```
[ma_precision]
pattern = .*
retentions = 60s:7d,300s:30d,1800s:180d,86400s:547d
```

Va appliquer pour tous les métriques la règle de stockage suivante:

- un point toutes les 60s, sur 7 jours
- un point toutes les 5min, sur 30 jours
- un point par 30min, sur 3 mois
- un point par jour, sur 1 an et demi

A noter qu'un point coûte 12 octets:

- 4 pour la date (taille d'un int32)
- 8 pour la valeur (taille d'un double64)

Ici un fichier créé avec cette rétention aura donc la taille:

- 60s:7d:
 - nombre de points: $7 \text{ j} * 86400 \text{ s} / 60 \text{ s} = 10080 \text{ points}$
 - taille = $12 \text{ o} * 10080 = 120 \text{ ko}$
- 300s:30d:
 - nombre de points : $30 \text{ j} * 86400 / 300 \text{ s} = 8640 \text{ points}$
 - taille = $12 \text{ o} * 8640 = 101 \text{ Ko}$
- 1800s:180d:
 - nombre de points : $180 \text{ j} * 86400 / 1800 \text{ s} = 8640 \text{ points}$
 - taille = $12 \text{ o} * 8640 = 101 \text{ Ko}$
- 86400s:547d
 - nombre de points : $547 \text{ j} * 86400 / 86400 \text{ s} = 547 \text{ points}$
 - taille = $12 \text{ o} * 547 = 6 \text{ Ko}$

Soit une Taille totale = **328Ko**, par métrique présente sur le disque (le fichier `.wsp`).

La précision est définie à la création d'un métrique

Attention: il est important de noter que la précision d'un fichier de métrique est prise en compte à la création du fichier.

Ainsi, un changement de la configuration de Graphite ne va pas redimensionner tous les métriques présents, mais seulement les nouveaux fichiers créés.

Précision livrée par défaut AVANT la V02.06.02

La précision livrée par défaut avant la 02.06.02 était la suivante:

```
[ma_precision]
pattern = .*
retentions = 60s:7d,300s:30d,3600s:180d,86400s:650d
```

Va appliquer pour tous les métriques la règle de stockage suivante:

- un point toutes les 60s, sur 7 jours
- un point toutes les 5min, sur 30 jours
- un point par heure, sur 6 mois
- un point par jour, sur 2 ans

A noter qu'un point coûte 12 octets:

- 4 pour la date (taille d'un int32)
- 8 pour la valeur (taille d'un double64)

Ici un fichier créé avec cette rétention aura donc la taille:

- 60s:7d:
 - nombres de points: $7 \text{ j} * 86400\text{s} / 60\text{s} = 10080 \text{ points}$
 - taille= $12\text{o} * 10080 = 120\text{ko}$
- 300s:30d:
 - nombre de points : $30 \text{ j} * 86400 / 300\text{s} = 8640 \text{ points}$
 - taille = $12\text{o} * 8640 = 103\text{Ko}$
- 3600s:180d:
 - nombre de points : $180 \text{ j} * 86400 / 3600\text{s} = 4320\text{points}$
 - taille = $12\text{o} * 4320 = 52\text{Ko}$
- 86400s:650d
 - nombre de points : $650 \text{ j} * 86400 / 86400\text{s} = 650 \text{ points}$
 - taille = $12\text{o} * 650 = 8\text{Ko}$

Soit une Taille totale = **283Ko**, par métrique présente sur le disque (le fichier .wsp).



La précision est définie à la création d'un métrique

Attention: il est important de noter que la précision d'un fichier de métrique est prise en compte à la création du fichier.

Ainsi, un changement de la configuration de Graphite ne va pas redimensionner tous les métriques présents, mais seulement les nouveaux fichiers créés.



La précision n'est pas modifiée lors d'une mise à jour

Attention: il est important de noter que la précision n'est pas modifiée lors d'une mise à jour de Shinken Enterprise.

Précision spécifique à un hôte

Il est possible de modifier la précision spécifiquement à un ou plusieurs hôtes.

Il faut ajouter un bloc dans le fichier `/opt/graphite/conf/storage-schemas.conf`

```
[specifique]
pattern = b6f742865f074584a605d31116833f3d.*.*
retentions = 1m:1h,5m:2h,1h:30d,1d:1y,8640m:3y
```

Le paramètre `pattern` est découpé comme suit : ID HOTE.ID CHECK.NOM DU METRIQUE



Le pattern doit être l'ID créé par Shinken

Dans notre exemple, tous les checks et tous les métriques de checks de l'hôte "shinken" (ID : b6f742865f074584a605d31116833f3d) obtiennent la rétention "specifique"



La précision est définie à la création d'un métrique

Attention: il est important de noter que la précision d'un fichier de métrique est prise en compte à la création du fichier ou un lancement de la commande `whisper-resize.py`

Ainsi, un changement de la configuration de Graphite ne va pas redimensionner tous les métriques présents, mais seulement les nouveaux fichiers créés.

Pour la prise en compte du nouveau paramètre :

- Vous pouvez supprimer le dossier/fichier contenant les métriques pour qu'ils se régénèrent avec le nouveau paramètre dans `/opt/graphite/storage/whisper/ID HOTE/ID CHECK/NOM DU METRIQUE.wsp`
- Vous pouvez utiliser le script suivant :
 - Pour tous les checks d'un hôte :
 - `/usr/bin/whisper-resize.py /opt/graphite/storage/whisper/ID HOTE/*/*.wsp 1m:1h,5m:2h,1h:30d,1d:1y,8640m:3y;`
 - Pour tous les métriques d'un check :
 - `/usr/bin/whisper-resize.py /opt/graphite/storage/whisper/ID HOTE/ID CHECK/*.wsp 1m:1h,5m:2h,1h:30d,1d:1y,8640m:3y;`
 - Pour un métrique spécifiquement :
 - `/usr/bin/whisper-resize.py /opt/graphite/storage/whisper/ID HOTE/ID CHECK/NOM DU METRIQUE.wsp 1m:1h,5m:2h,1h:30d,1d:1y,8640m:3y;`

Une fois le fichier créé ou régénéré, pour avoir confirmation de la bonne prise en compte de ce nouveau bloc, dans le fichier `/opt/graphite/storage/log/carbon-cache/carbon-cache-a/creates.log` vous aurez les lignes suivantes :

```
29/06/2021 16:41:44 :: new metric b6f742865f074584a605d31116833f3d.92182da3c28c5193c0e36b4a68b5e896.
connexion_time matched schema specifique
29/06/2021 16:41:44 :: new metric b6f742865f074584a605d31116833f3d.92182da3c28c5193c0e36b4a68b5e896.
connexion_time matched aggregation schema default_average
29/06/2021 16:41:44 :: creating database file /opt/graphite/storage/whisper/b6f742865f074584a605d31116833f3d
/92182da3c28c5193c0e36b4a68b5e896/connexion_time.wsp (archive=[(60, 60), (300, 24), (3600, 720), (86400,
365), (518400, 182)] xff=0.01 agg=average)
```

Modifier la précision sur tous les éléments d'une installation de Shinken en cours de fonctionnement

Script de migration massif

Si vous utilisez Shinken depuis quelques temps et que vous souhaitez modifier la précision des métriques sans perdre les anciennes valeurs, vous pouvez utiliser le script `whisper-resize.py`

Dans un premier temps, modifier à votre convenance le paramètre `retentions` du bloc "everything_1min" dans le fichier `/opt/graphite/conf/storage-schemas.conf`

Pour que le paramètre soit pris en compte, il faut redémarrer carbon-cache :

```
/etc/init.d/carbon-cache restart
```

Une fois `carbon-cache` redémarré il faut créé le script `mass_modif_retention_graphite.sh` qui va modifier tous les fichiers `.wsp` que vous avez :

```
#!/bin/bash

FOLDER="/opt/graphite/storage/whisper"
RETENTION="XXXXXX"

for i in $(find $FOLDER/$1 -iname "*.wsp"); do
    if [ -a $i ];
    then
        /usr/bin/whisper-resize.py $FOLDER/$i $RETENTION;
        /usr/bin/chown apache:apache $FOLDER/$i
    fi;
done
```



Il faut remplacer la variable RETENTION par celle que vous avez mis dans le fichier storage-schemas.conf avant exécution du script

Rendre le script exécutable

```
chmod +x mass_modif_retention_graphite.sh
```

Et le lancer en mettant une faible priorité tant au CPU qu'à la consommation disque:

```
ionice nice ./mass_modif_retention_graphite.sh
```

Suppression des anciennes données après migration

Le script va créer pour chaque fichier converti un fichier .bak qui est une sauvegarde de l'ancien fichier, ce qui signifie que la taille nécessaire à cette conversion va nécessiter beaucoup d'espace disque. Vous pouvez annuler la création de ce fichier en ajoutant le paramètre "--nobackup" au script. L'ancien fichier sera supprimé uniquement en cas de succès de la conversion.

Dans le script, remplacer cette ligne :

```
/usr/bin/whisper-resize.py $FOLDER/$i $RETENTION;
```

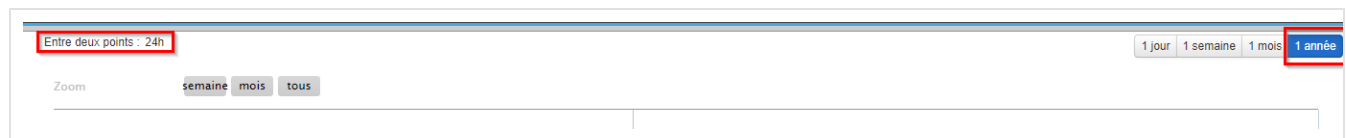
Par cette ligne :

```
/usr/bin/whisper-resize.py $FOLDER/$i $RETENTION --nobackup;
```

Visualisation des données avec la nouvelle précision

Une fois la conversion terminée vous pouvez voir le changement directement dans l'interface de visualisation de Shinken

- Les courbes seront lissées pour les anciennes données.
- En fonction de la rétention choisie, Graphite retourne la meilleure précision qu'il a pour mais uniquement pour l'ensemble des données demandées par l'utilisateur.
 - Par exemple, si vous avez choisi cette précision : 5m:90d,30m:180d,1h:3y et vous souhaitez voir les métriques depuis Shinken sur 1 année.
 - L'écart entre 2 points est de 24h sur toute la période
 - Même sur les 3 et 6 derniers mois, car pour avoir tous les points, Graphite a dû prendre la dernière précision, soit 1h sur 3ans.



- Si vous souhaitez afficher une période précise de précision (comme ici 3 mois) mais qui n'est pas listée par l'interface de Shinken, il faut pour l'instant l'afficher dans [Grafana](#) à la place, qui permet de choisir pour chaque dashboard la période demandée.

En cas d'erreur

Erreur dans le format du fichier storage-schemas.conf

Si vous avez cette erreur au redémarrage de carbon-cache c'est que la configuration d'un fichier n'est pas bonne :

```
/etc/init.d/carbon-cache restart
Stopping carbon-cache... [FAILED]
Starting carbon-cache...Traceback (most recent call last):
  File "/opt/graphite/bin/carbon-cache.py", line 30, in <module>
    run_twistd_plugin(__file__)
  File "/opt/graphite/lib/carbon/util.py", line 92, in run_twistd_plugin
    runApp(config)
  File "/usr/lib64/python2.7/site-packages/twisted/scripts/twistd.py", line 23, in runApp
    _SomeApplicationRunner(config).run()
  File "/usr/lib64/python2.7/site-packages/twisted/application/app.py", line 411, in run
    self.application = self.createOrGetApplication()
  File "/usr/lib64/python2.7/site-packages/twisted/application/app.py", line 489, in createOrGetApplication
    ser = plg.makeService(self.config.subOptions)
  File "/opt/graphite/lib/twisted/plugins/carbon_cache_plugin.py", line 21, in makeService
    return service.createCacheService(options)
  File "/opt/graphite/lib/carbon/service.py", line 127, in createCacheService
    from carbon.writer import WriterService
  File "/opt/graphite/lib/carbon/writer.py", line 34, in <module>
    schemas = loadStorageSchemas()
  File "/opt/graphite/lib/carbon/storage.py", line 139, in loadStorageSchemas
    except InvalidConfiguration, e:
NameError: global name 'InvalidConfiguration' is not defined
```

Erreur dans le format des précisions de rétention

Si vous avez cette erreur, ça signifie que votre définition de précision n'est pas bonne :

```
Traceback (most recent call last):
  File "/usr/bin/whisper-resize.py", line 45, in <module>
    for retentionDef in args[1:]
  File "/usr/lib/python2.7/site-packages/whisper.py", line 88, in parseRetentionDef
    (precision, points) = retentionDef.strip().split(':')
ValueError: too many values to unpack
```