

InterfaceUsage Switch SNMPv1v2

Sommaire

- Contexte
- Paramétrage
 - Données utilisées provenant du modèle
 - Données communes pour les checks du modèle
 - Données spécifiques pour ce check
 - Données DFE (Duplicate Foreach)
 - Données utilisées provenant du check
 - Données globales
 - Propriétés de l'hôte
- Résultat
 - Exemple
 - Interprétation
 - Statut
 - Résultat
 - Résultat Long
- Métriques

Les fichiers de log du broker sont situés dans le dossier `/var/log/shinken/`. Pour plus d'informations, consultez la page [Fichiers Logs](#).

Démarrage du broker

```
[YYYY-MM-DD HH:MM:SS] INFO : [broker] [START-DAEMON] The daemon (version=02.08.01-release.fr) is now
started as a daemon (detached from any shell) with pid=15412
[YYYY-MM-DD HH:MM:SS] INFO : [broker] Using the local log file '/var/log/shinken/brokerd.log'
[YYYY-MM-DD HH:MM:SS] INFO : [broker] Printing stored debug messages prior to our daemonization
[YYYY-MM-DD HH:MM:SS] INFO : [broker] [ SYSTEM ] System resource number of open files is
set to (soft:131070 / hard:131070 ) (set at system max values)
[YYYY-MM-DD HH:MM:SS] INFO : [broker] [ SYSTEM ] System resource number of process/threads is
set to (soft:unlimited / hard:unlimited ) (set at system max values)
[YYYY-MM-DD HH:MM:SS] INFO : [broker] Starting HTTP daemon
[YYYY-MM-DD HH:MM:SS] INFO : [broker]
|-----|
[YYYY-MM-DD HH:MM:SS] INFO : [broker] broker is
starting
[YYYY-MM-DD HH:MM:SS] INFO : [broker]
|-----|
```

Avec affichage:

- Du fichier de log défini dans sa configuration (broker.ini)
- Du nombre de processus/threads maximum autorisé par le système pour ce démon
- Du nombre de fichiers ouverts maximum autorisé par le système

Chargement d'une configuration

Début du chargement d'une configuration d'architecture (ou configuration de changement d'architecture)

Quand on reçoit le premier envoi de configuration d'architecture (avec nos modules, spare/non spare, les premiers Schedulers auxquels se connecter, etc.) on va avoir la ligne suivante:

```
[YYYY-MM-DD HH:MM:SS] INFO : [ broker-master ] [ CONFIGURATION ] ----- Loading the new
configuration from the arbiter
```

Lorsque l'on va recevoir de nouveaux envois (nouvel Arbiter, ou bien simplement le reste des Schedulers par exemple), on va avoir:

```
[YYYY-MM-DD HH:MM:SS] INFO : [ broker-master ] [ CONFIGURATION ] ----- Loading a configuration
update from the arbiter
```

Notification d'un nouveau démarrage d'arbiter

Dans le cas où un Arbiter a redémarré, on aura la ligne suivante:

```
[YYYY-MM-DD HH:MM:SS] INFO : [ broker-master ] [ CONFIGURATION ] The arbiter send us a new configuration: [configuration_uuid=060e70df204a61be70f75c0622e118, arbiter=arbiter-master, date=20-10-2020 15:37:27]
```

Cas d'un démon recevant un nouveau démon spare ou une assignation d'un démon master

Dans le cas où un master reçoit le nom de son démon spare, on aura:

```
[YYYY-MM-DD HH:MM:SS] INFO : [ broker-master ] [ CONFIGURATION ] [ MASTER ] My spare daemon is now "broker-spare"
```

Dans le cas où un spare reçoit le nom de son démon master, on aura:

```
[YYYY-MM-DD HH:MM:SS] INFO : [ broker-spare ] [ CONFIGURATION ] [ SPARE ] I am now the spare of the master daemon "broker-master"
```

Réception des liens vers d'autres démons (schedulers, arbiters, ...)

Quand un démon reçoit une liste de démons (pour se connecter par exemple), on aura un affichage du genre:

```
[YYYY-MM-DD HH:MM:SS] INFO : [ broker-master ] [ CONFIGURATION ] The arbiter asked us to remove daemons:
[YYYY-MM-DD HH:MM:SS] INFO : [ broker-master ] [ CONFIGURATION ] - REMOVED scheduler : [name=scheduler-master ] [shard_id= 256] [uri=http://localhost:7768/]
[YYYY-MM-DD HH:MM:SS] INFO : [ broker-master ] [ CONFIGURATION ] The arbiter send us new daemons:
[YYYY-MM-DD HH:MM:SS] INFO : [ broker-master ] [ CONFIGURATION ] + ADDED scheduler : [name=scheduler-spare ] [shard_id= 256] [uri=http://localhost:8768/]
```

Ici par exemple:

- La shard **256** n'est plus gérée par le scheduler **scheduler-master**
- Mais désormais par son spare **scheduler-spare**

Application par le démon de la propriété satellitemap (remapping d'adresse pour gérer un VLAN)

Quand un démon a un paramètre satellitemap, il va changer l'adresse d'un autre démon par une autre (pour par exemple gérer le cas où il tourne dans un vlan avec un plan d'adressage particulier). Cette application se voit via le log suivant:

```
[YYYY-MM-DD HH:MM:SS] INFO : [ broker-master ] [ CONFIGURATION ] [ SATELLITEMAP ] Replacing the daemon scheduler-secondary to address:port from localhost:8768 => 192.168.1.124:8768 as defined in our daemon .cfg file (satellitemap property)
```

Ici le Scheduler scheduler-secondary est passé de l'adresse **localhost:8768** à **192.168.1.124:8768**.

Un tour de boucle

Début d'un tour

```
[YYYY-MM-DD HH:MM:SS] INFO: [broker-master ] [BROKER TIME ] [ === Loop start === ] =====
=====
```

Récupération des broks des schedulers et arbiters

L'Arbiter envoie ses broks vers le broker:

```
[YYYY-MM-DD HH:MM:SS] INFO : [broker-master ] [RECEIVE BROKS] [ arbiter ] [PERF] [ 0.000 ]s -
Add 1 broks into INTERNAL queue (new size=18) and the EXTERNAL queue (new size=18)
[YYYY-MM-DD HH:MM:SS] INFO : [broker-master ] [RECEIVE BROKS] [ arbiter ]
----- 1 composed of: architecture_export_map=1
```

Le broker récupère les broks depuis un Scheduler:

```
[YYYY-MM-DD HH:MM:SS] INFO : [broker-master ] [GET BROKS ] [ scheduler-master ] [PERF] [ 0.007 ]s -
Add 16 broks into INTERNAL queue (new size=16) and the EXTERNAL queue (new size=16)
[YYYY-MM-DD HH:MM:SS] INFO : [broker-master ] [GET BROKS ] [ scheduler-master ]
----- 16 composed of: host_check_result=10, host_next_schedule=6
```

Avec pour les deux cas:

- Affichage du nombre de broks récupérés sur le démon, et affichage de la taille des files d'attente une fois rajoutés
- Affichage du type de broks récupérés, ainsi que leur nombre

Quand on a une erreur de transfert qui faisait perdre des broks dans le passé, on a cette entrée dans les logs:

```
[YYYY-MM-DD HH:MM:SS] WARNING: [broker-master] [ GIVE BROKS ] [ broker-master ] Did fail to transfert broks
from the scheduler "scheduler-master": [[Connexion error to https://192.80.10.220:7768/ : Operation timed
out after 120000 milliseconds with 802816 out of 22791250 bytes received]. THESESES BROKS ARE LOST AND CANNOT
BE RETRIEVED
```

Elle disparaîtra quand on aura bien testé le mécanisme de reprise sur erreur dans une future version.

Envoi des broks aux modules externes

Statut des files d'envoi

```
[YYYY-MM-DD HH:MM:SS] INFO: [broker-master ] [MANAGE BROKS ] [ EXTERNAL MODULE ] => Number of "Broks Sets"
not eaten in MODULE queues (WebUI5-ha): 11 (WebUI3-ha): 11 (WebUI7-ha): 11 (WebUI4-ha): 11 (WebUI8-ha): 11
(WebUI1-ha): 11 (WebUI2-ha): 11 (WebUI6-ha): 11
```

À chaque tour de boucle, le broker envoie 1 ensemble de broks à chaque WebUI. 1 ensemble est composé d'autant de broks qu'il a reçus dans le tour.

Si le nombre est plus gros que 1, c'est que les WebUIs mettent du temps à digérer les ensembles.

- C'est courant au démarrage, car les broks initiaux sont longs à être traités
- Mais cela ne devrait pas arriver après

Préparation des Broks pour l'envoi

```
[YYYY-MM-DD HH:MM:SS] INFO: [broker-master ] [MANAGE BROKS ] [ PREPARING BROKS ] [PERF] [ 0.001 ]s,
preparing broks lists for INTERNAL and EXTERNAL modules
```

Chaque tour de boucle le broker préparer les listes d'envoi avec les nouveaux broks reçus.

Envoi vers les modules externes

```
[YYYY-MM-DD HH:MM:SS] INFO: [broker-master ] [MANAGE BROKS ] [ EXTERNAL MODULE ] - PUSHED [ 0.331s,
limit=5.000s ]s, EXTERNAL queue evolution: [ 424 broks => 0 broks remaining ] [ 424 broks managed ]
[ Push average speed = 1928 broks/s]
```

Le broker a envoyé **424** broks en **0.331s**, et avait laissé une limite de temps de **5s** pour cet envoi (calcul basé sur la vitesse moyenne des derniers envois, ici **1928 broks/s**, et une marge de sécurité).

À noter: si le nombre de broks remaining est différent de zéro, ceci signifie que le broker a reçu des broks pendant la phase d'envoi, et qu'il les enverra le prochain tour.

Envoi des broks aux modules internes (sans leur propre processus)

Ces logs ne seront affichés que si le broker a des modules internes.

```
[YYYY-MM-DD HH:MM:SS] INFO: [broker-master ] [MANAGE BROKS ] [ INTERNAL MODULE ] - EXECUTED [ 0.239 ]s,
INTERNAL queue evolution: [ 424 broks => 238 broks remaining ] [ 424 broks managed ]
[YYYY-MM-DD HH:MM:SS] INFO: [broker-master ] ----- Details of
INTERNAL modules execution time: (Graphite-Perfdata=0.071s), (Simple-log=0.010s), (sla=0.086s)
```

Le broker a fourni **424** broks aux modules internes (ceux qui n'ont pas leur propre processus), en **0.239s** au total. Ici le nombre de broks remaining est différent de zéro, ceci signifie que le broker a reçu des broks pendant la phase d'envoi, et qu'il les enverra le prochain tour.

Il fournit ensuite le détail de temps de chaque module interne.

Récupération des commandes (demande de prise en compte, demande pour relancer une vérification, etc)

```
[YYYY-MM-DD HH:MM:SS] INFO: [broker-master ] [MODULES ] [ INTERNAL COMMANDS ] [PERF] [ 0.001 ]s Did
read 0 external commands (like recheck, set acknowledge, etc) from modules
```

Le broker récupère les commandes (comme une création de périodes de maintenance, etc.) et le temps que ceci lui a demandé.

Appel au modules internes chaque seconde

Chaque fin de tour, un appel est lancé vers les modules internes afin qu'ils puissent faire des actions spécifiques (par exemple vérifier un cache, vider leurs éléments pas encore envoyés, etc.)

```
[YYYY-MM-DD HH:MM:SS] INFO: [broker-master ] [MODULES ] [ TIME IN BROKER ] [PERF] [ 0.025 ]s All
modules "ticks" are done. Execution times by modules: (Graphite-Perfdata=0.001s), (sla=0.024s)
```

Avec:

- Le temps total
- Le temps passé par chaque module

Fin d'un tour

```
[YYYY-MM-DD HH:MM:SS] INFO: [broker-master ] [BROKER TIME ] [ === Loop stop === ] [PERF] [ 0.397 ]s
```

Le broker donne le temps qu'il a passé sur ce tour de boucle. Ce dernier doit rester sous la seconde sauf pendant la phase de réception d'une nouvelle configuration où il peut dépasser ce temps.

Surcharge serveur en activité disque, ralentissant l'écriture des logs

Si le serveur hébergeant le démon est surchargé en termes d'IO disques sur le volume qui héberge le fichier de log, alors ce dernier va mettre du temps à s'écrire et va ralentir tout le démon. Il faut alors dans la mesure du possible, isoler le volume des disques sur un disque moins chargé pour ne pas ralentir le démon.

En cas de soucis vous aurez dans les logs l'entrée suivante:

```
2020-05-04 00:00:51 WARNING : [ LOGGER ]
2020-05-04 00:00:51 WARNING : [ LOGGER ]
-----
2020-05-04 00:00:51 WARNING : [ LOGGER ] [ WRITING ] The log write time is very high (1.87s). Please look at
your log disk performance.
2020-05-04 00:00:51 WARNING : [ LOGGER ]
-----
2020-05-04 00:00:51 WARNING : [ LOGGER ]
```

Logs de chargement des modules

Les démons ont une phase de chargement des modules qui est décrite dans la page [HIDDEN - Logs de chargement des modules](#)