

Les logs du Scheduler

Fichiers+Logs

Sommaire

- Démarrage du démon
- Réception et chargement de la configuration du démon
 - Quand un Scheduler reçoit une nouvelle configuration de l'Arbiter, il log
 - Quand un Scheduler est en train de charger sa nouvelle configuration, il log
 - Quand un Scheduler a fini de charger la nouvelle configuration reçu, il log :
- Demande et génération des broks initiaux
- Communication entre Schedulers
 - Quand un Scheduler distant n'a pas reçu de configuration de l'Arbiter
 - Quand un Scheduler distant reçoit une configuration de l'Arbiter
 - Quand un Scheduler distant reçoit une nouvelle configuration de l'Arbiter
- Requête d'export des données
 - Avec les noms des éléments
 - Sans les noms des éléments
- Erreurs possibles
 - Le Scheduler est en phase de démarrage
 - Le Scheduler ne gère pas encore de configuration
 - La configuration interdit l'export de données
 - Impossible d'exporter les noms des éléments car le mot de passe d'accès n'a pas été configuré
 - Accès refusé suite à l'utilisation d'un mauvais mot de passe
- Erreur de cohérence des périodes de maintenance
- Exécution de commandes externes reçues d'un Receiver
- Exemples
 - Échanges par paquet de taille limité des Broks avec le Broker
 - Log de performance de la boucle du scheduler
 - Surcharge serveur en activité disque, ralentissant l'écriture des logs
 - Logs d'erreur concernant des objets d'exécution de check qui restent en mémoire dans le Scheduler
 - Logs de WARNING concernant la suppression d'objets "notification" défectueux
 - Logs de WARNING concernant les notifications qui ne sont pas envoyées à cause des périodes de notification
 - Log d'ERROR quand on charge depuis la rétention un nombre trop important de vérifications sur les hôtes ou les checks
 - Options du support Shinken
 - Forcer l'étalement des checks au démarrage

Démarrage du démon

Au démarrage le démon affiche plusieurs logs contenant ses informations dont :

- Ses limites système en nombre de fichiers ouvrables, et nombre de threads/processus

```
[YYYY-MM-DD HH:MM:SS] INFO      : [ scheduler-name ] [ SYSTEM          ] System resource number of open files
is set to      (soft:1024      / hard:1024      ) (from parameter max_file_descriptor_limit)
[YYYY-MM-DD HH:MM:SS] INFO      : [ scheduler-name ] [ SYSTEM          ] System resource number of processes
/threads is set to (soft:unlimited / hard:unlimited ) (set at system max values)
```

- Sur quelle(s) interface(s) le démon écoute

S'il écoute sur toutes les interfaces:

```
[YYYY-MM-DD HH:MM:SS] INFO      : [ scheduler-name ] [schedulerdaemon] The daemon listens on all network
interfaces.
```

S'il écoute sur une interface précise:

```
[YYYY-MM-DD HH:MM:SS] INFO      : [ scheduler-name ] [schedulerdaemon] The daemon listens on the IP_INTERFACE:
PORT_INTERFACE network interface.
```

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

Réception et chargement de la configuration du démon

Les logs suivants permettent de suivre le chargement de la configuration de supervision entre l'Arbiter les Schedulers jusqu'aux interfaces : webui / livestatus / livedata

Il existe 2 types d'identifiants de configuration (représentation de la configuration)

- **configuration_uuid**: uuid de configuration totale généré par l'Arbiter
- **configuration_part_id**: id de la partie de configuration gérée par un Scheduler

Quand un Scheduler reçoit une nouvelle configuration de l'Arbiter, il log

```
[YYYY-MM-DD HH:MM:SS] INFO : [ scheduler-name ] [schedulerdaemon] [ CONFIGURATION ] [ NEW ] Configuration received [ configuration_part_id=configuration_part_id, configuration_uuid=configuration_uuid, arbiter=arbiter_name, date=creation_date, ]
```

- **configuration_part_id**: id de la partie de configuration spécifiquement gérée par ce Scheduler (unique pour chaque Scheduler)
- **configuration_uuid**: uuid créé lors du démarrage de l'Arbiter qui correspond donc à l'id de la configuration géré par l'Arbiter
- **creation_date**: date du démarrage de l'Arbiter
- **arbiter_name**: nom de l'Arbiter qui a créé cette configuration

Exemple Scheduler réception d'une nouvelle configuration

```
[YYYY-MM-DD HH:MM:SS] INFO : [ scheduler-name ] [schedulerdaemon] [ CONFIGURATION ] [ NEW ] Configuration received [ configuration_part_id=1280, configuration_uuid=e551f7f93f2d45bfafae77fc302db7a2, arbiter=arbiter-master1, date=15-05-2020 15:13:38 ]
```

✗ Dans le cas où le scheduler n'est pas de la même version que l'arbiter et que l'option `*mismatch_version_error*` est activé sur l'arbiter:

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ scheduler-name ] Incompatible daemon version : Your Arbiter daemon is in version [XX.XX.XX-release.fr culmulative-patch-YY] while this daemon is in version [XX.XX.XX-release.fr culmulative-patch-YY]. Refusing this configuration.
```

⚠ Dans le cas où le scheduler n'est pas de la même version que l'arbiter et que l'option `*mismatch_version_error*` est désactivé sur l'arbiter:

```
[YYYY-MM-DD HH:MM:SS] WARNING : [ scheduler-name ] Incompatible daemon version : Your Arbiter daemon is in version [XX.XX.XX-release.fr culmulative-patch-YY] while this daemon is in version [XX.XX.XX-release.fr culmulative-patch-YY].
```

Quand un Scheduler est en train de charger sa nouvelle configuration, il log

Le log suivant apparaît si le Scheduler n'a pas fini de charger sa configuration est qu'il reçoit une demande de statut.

```
[YYYY-MM-DD HH:MM:SS] INFO : [ scheduler-name ] [scheduler][0] Someone asks to get the raw stats (daemon Health) but the scheduler is not initialized
```

Quand un Scheduler a fini de charger la nouvelle configuration reçu, il log :

```
[YYYY-MM-DD HH:MM:SS] INFO : [ scheduler-name ] [schedulerdaemon] [ CONFIGURATION ] [ NEW ] Loaded in [loading_time]s => [ configuration_part_id=configuration_part_id, configuration_uuid=configuration_uuid, date=creation_date, author=arbiter_name ]
```

- **configuration_part_id**: id de la partie de configuration géré par ce Scheduler (unique pour chaque Scheduler)
- **configuration_uuid**: uuid crée lors du démarrage de l'Arbiter qui correspond donc à l'id de la configuration géré par l'Arbiter
- **creation_date**: date du démarrage de l'Arbiter
- **arbiter_name**: nom de l'Arbiter qui a créé cette configuration

- **loading_time:** temps de chargement de la configuration

Exemple Scheduler chargement de la nouvelle configuration

```
[YYYY-MM-DD HH:MM:SS] INFO    : [ scheduler-name ] [ schedulerdaemon ] [ CONFIGURATION ] [ NEW ] Loaded in
[1.31168293953]s => [ configuration_part_id=1280 configuration_uuid=e551f7f93f2d45bfafae77fc302db7a2,
author=arbiter-master1, date=15-05-2020 15:13:38]
```

Demande et génération des broks initiaux

Les modules de Broker ont besoin des Broks dit "initiaux" afin d'avoir une image complète des éléments de supervision (*hôtes, checks, mais aussi timeperiod et commandes*). Quand une nouvelle configuration est chargée par le Scheduler, le Broker le détecte et demande une génération des Broks initiaux. Ceci est visible par la log suivant :

```
[YYYY-MM-DD HH:MM:SS] INFO    : [ scheduler-name ] [ INITIAL BROKS    ] [ REGISTERING ] [ broker-master ] The
Broker is registering for initial broks generation. ( Currently 1 registered )
```

La génération va commencer.

```
[YYYY-MM-DD HH:MM:SS] INFO    : [ scheduler-name ] [ INITIAL BROKS    ] [ GENERATING ] [ TIMING ] Generating
initial broks for configuration [shard_id=256, scheduler=scheduler-name,
configuration_uuid=60b36ae8df5f4090bd7ef5d576ee1a15, arbiter=arbiter-name, architecture=architecture-name,
date=26-08-2022 16:13:27, active=True] (currently have 36 hosts and 463 services)
```

Pendant que la génération des broks initiaux se fait, d'autres demandes vont se rajouter à la première si on a plusieurs Brokers.

Une fois la génération finie, on aura le log suivant :

```
[YYYY-MM-DD HH:MM:SS] INFO    : [ scheduler-name ] [ INITIAL BROKS    ] [ GENERATING ] [ DONE ] [ Elapsed
time=0.042s ] 241 initial broks are generated for 2 brokers:
[YYYY-MM-DD HH:MM:SS] INFO    : [ scheduler-name ] [ INITIAL BROKS    ] [ GENERATING ] [ DONE ] - broker-
master
[YYYY-MM-DD HH:MM:SS] INFO    : [ scheduler-name ] [ INITIAL BROKS    ] [ GENERATING ] [ DONE ] - broker-
secondary
```

Et les Brokers seront prévenus qu'ils peuvent désormais revenir télécharger les Broks :

```
[YYYY-MM-DD HH:MM:SS] INFO    : [ scheduler-name ] [ INITIAL BROKS    ] [ GENERATING ] [ DONE ] [ broker-
master ] The broker is warned that the generation is done and can be GET.
[YYYY-MM-DD HH:MM:SS] INFO    : [ scheduler-name ] [ INITIAL BROKS    ] [ GENERATING ] [ DONE ] [ broker-
secondary] The broker is warned that the generation is done and can be GET.
```

Communication entre Schedulers

Quand un Scheduler distant n'a pas reçu de configuration de l'Arbiter

Exemple Scheduler réception d'une nouvelle configuration

```
[YYYY-MM-DD HH:MM:SS] INFO    : [ scheduler-name ] [ CONFIGURATION ] The scheduler-distant has not yet
received any configuration from Arbiter.
```

Quand un Scheduler distant reçoit une configuration de l'Arbiter

Exemple Scheduler réception d'une nouvelle configuration

```
[YYYY-MM-DD HH:MM:SS] INFO    : [ scheduler-name ] [ CONFIGURATION ] The scheduler-distant change it's configuration from configuration_uuid to new_configuration_uuid
```

- **configuration_uuid**: uuid créé lors du démarrage de l'Arbiter qui correspond donc à l'id de la configuration gérée par l'Arbiter
- **new_configuration_uuid**: uuid créé lors du démarrage de l'Arbiter qui correspond donc au nouvel id la configuration gérée par l'Arbiter

Exemple Scheduler réception d'une nouvelle configuration

```
[YYYY-MM-DD HH:MM:SS] INFO    : [ scheduler-name ] [ CONFIGURATION ] The scheduler-distant change it's configuration from 6ddbcbdd9260e40d9a8a48e1eabc875a5 to 32ab5f3457fb4c3fbe2415d873ae199e
```

Quand un Scheduler distant reçoit une nouvelle configuration de l'Arbiter

Exemple Scheduler réception d'une nouvelle configuration

```
[YYYY-MM-DD HH:MM:SS] INFO    : [ scheduler-name ] [ CONFIGURATION ] The scheduler-distant received a new configuration (uuid=073c072c6d524b38bb4c08b1fdfa7f89)
```

Requête d'export des données

La commande `shinken-scheduler-export-data` invoquée depuis le serveur de l'Arbiter permet de questionner les Schedulers pour extraire des données sur les éléments supervisés.

Lorsque cette commande est utilisée, les logs qui suivent permettent de voir la nature de la réponse faite par un Scheduler.

Avec les noms des éléments

```
[YYYY-MM-DD HH:MM:SS] INFO    : [ scheduler-name ] [ EXPORT DATA ] full export of XXX elements in X.XXXs from Arbiter IPXXX.XXX.XXX.XXX
```

Sans les noms des éléments

```
[YYYY-MM-DD HH:MM:SS] INFO    : [ scheduler-name ] [ EXPORT DATA ] anonymous export of XXX elements in X.XXXs from Arbiter IPXXX.XXX.XXX.XXX
```

Erreurs possibles

Le Scheduler est en phase de démarrage

```
[YYYY-MM-DD HH:MM:SS] WARNING: [ scheduler-name ] [ EXPORT DATA ] Request from Arbiter IPXXX.XXX.XXX.XXX fails. Scheduler is not ready (initialisation ongoing)
```

Le Scheduler ne gère pas encore de configuration

```
[YYYY-MM-DD HH:MM:SS] WARNING: [ scheduler-name ] [ EXPORT DATA ] Request from Arbiter IPXXX.XXX.XXX.XXX fails. Scheduler is not ready (waiting for configuration)
```

La configuration interdit l'export de données

```
[YYYY-MM-DD HH:MM:SS] WARNING: [ scheduler-name ] [ EXPORT DATA ] Request from Arbiter IPXXX.XXX.XXX.XXX fails. Export is disabled by configuration parameter scheduler__export_data__enabled
```

Impossible d'exporter les noms des éléments car le mot de passe d'accès n'a pas été configuré

```
[YYYY-MM-DD HH:MM:SS] WARNING: [ scheduler-name ] [ EXPORT DATA ] Request from Arbiter IPXXX.XXX.XXX.XXX fails. The daemon__export_data__password parameter is missing or void, that is forbidden for not anonymous request
```

Accès refusé suite à l'utilisation d'un mauvais mot de passe

```
[YYYY-MM-DD HH:MM:SS] WARNING: [ scheduler-name ] [ EXPORT DATA ] Request from Arbiter IPXXX.XXX.XXX.XXX fails. The password is not matching "daemon__export_data__password" parameter
```

Erreur de cohérence des périodes de maintenance

Les périodes de maintenance ont une incohérence entre non identifiée qui consiste à ce qu'une période de maintenance va être démarrée deux fois, ce qui va poser soucis lors de son arrêt (l'hôte/check sera dans un état incohérent). Pour l'instant, nous n'avons pas trouvé la source de l'incohérence, mais nous avons mis des protections pour les éviter et corriger.

Au chargement de la rétention, on aura une entrée "ERROR":

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ scheduler-name ] [ DOWNTIME-INCOHERENCY ] The host Linux has bad downtime values (saved number of downtime=1, actual=0). We are fixing the values. Please report it to the support.
```

Lors du second démarrage de la période de maintenance:

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ scheduler-name ] [ DOWNTIME-INCOHERENCY ] The downtime [Downtime id=1596532528600972 active=active type=fixed start=Tue Aug 4 11:16:06 2020 - end=Tue Aug 4 12:15:06 2020] on server got activated twice. This is a bug and MUST be reported to support for investigation. Thanks.
```

Dans les deux cas, il faut récupérer les logs du Scheduler, et les donner en analyse au support Shinken.

Exécution de commandes externes reçues d'un Receiver

Lorsque le Receiver envoie des commandes externes au Scheduler, il est possible que ce dernier n'ait pas reçu de configuration de la part de l'Arbiter et qu'il ne soit pas en capacité d'exécuter ces commandes.

```
[YYYY-MM-DD HH:MM:SS] INFO : [ scheduler-name ] [ BUS COMMANDS ] Get external commands (like recheck, set acknowledge, etc) from the Receiver receiver-master but i am not ready. Waiting for configuration from Arbiter.
```

Dans le cas où, les commandes ont pu être exécutées :

```
[YYYY-MM-DD HH:MM:SS] INFO : [ scheduler-name ] [ BUS COMMANDS ] Running 200 external commands (like recheck, set acknowledge, etc) received from the Receiver receiver-master
```

Les commandes externes différentes des mise à jour de statut sont également listées :

```
[YYYY-MM-DD HH:MM:SS] INFO : [ scheduler-name ] [ EXTERNAL COMMAND ] received command [ COMMAND_NAME ] on [ ELEMENT ] with args [ OPTIONAL_PARAMETERS ]
```

Les valeurs les plus courantes pour COMMAND_NAME sont les suivantes :

COMMAND_NAME	Type d'élément	Action
--------------	----------------	--------

Check acknowledge	check	Définit une prise en compte sur un check
Check acknowledge deletion	check	Supprime une prise en compte sur un check
Create check downtime	check	Définit une période de maintenance sur un check
Delete check downtime	check	Supprime une période de maintenance sur un check
Schedule immediate check	check	Force l'exécution de la vérification du statut d'un check
Host acknowledge	host	Définit une prise en compte sur un hôte
Host acknowledge deletion	host	Supprime une prise en compte sur un hôte
Create host downtime	host	Définit une période de maintenance sur un hôte
Delete host downtime	host	Supprime une période de maintenance sur un hôte
Schedule immediate host	host	Force l'exécution de la vérification du statut d'un hôte

Exemples

Mise en place d'une prise en compte sur un check

```
[2023-03-30 14:10:28] INFO : [ scheduler-supdesup2 ] [ EXTERNAL COMMAND ] received command [ Check acknowledge ] on [ Int-II - CLOUD - int-google-1/Load Average SSH ] with args [ 2,True,True,bmourgues,En cours de résolution ]
```

Retrait d'une prise en compte sur un check

```
[2023-03-30 14:12:18] INFO : [ scheduler-supdesup2 ] [ EXTERNAL COMMAND ] received command [ Check acknowledge deletion ] on [ Int-II - CLOUD - int-google-1/Load Average SSH ] with args [ ]
```

Mise en place d'une période de maintenance sur un check

```
[2023-03-30 14:13:30] INFO : [ scheduler-supdesup2 ] [ EXTERNAL COMMAND ] received command [ Create check downtime ] on [ Int-II - CLOUD - int-google-1/Load Average SSH ] with args [ 1680178443,1680181983,True,0,0,bmourgues,Mise à jour en cours ]
```

Retrait d'une période de maintenance sur un check

```
[2023-03-30 14:15:42] INFO : [ scheduler-supdesup2 ] [ EXTERNAL COMMAND ] received command [ Delete check downtime ] on [ Downtime id=1680178410781868 active=active type=fixed start=Thu Mar 30 14:14:03 2023 - end=Thu Mar 30 15:13:03 2023 on Int-II - CLOUD - int-google-1/Load Average SSH ] with args [ ]
```

Check forcé

```
[2023-03-30 11:01:04] INFO : [ scheduler-supdesup2 ] [ EXTERNAL COMMAND ] received command [ Schedule immediate check ] on [ Int-II - FRANCE - integration-1/Broker - broker-france-webui - Memory consumption ] with args [ 1680166864 ]
```

Échanges par paquet de taille limité des Broks avec le Broker

Quand le broker demande les broks au Scheduler, ce dernier va avoir dans ses logs:

```
[YYYY-MM-DD HH:MM:SS] INFO : [ scheduler-name ] [ GIVE BROKS ] [ broker-master ] Sending 5 broks (1.3kB)
```

Si on se retrouve dans un cas où la limite du **broks_packet_size** est atteinte (sur les broks initiaux par exemple), on va avoir la ligne suivante:

```
[YYYY-MM-DD HH:MM:SS] INFO : [ scheduler-name ] [ GIVE BROKS ] [ broker-master3 ] Sending 2588 broks (413.4 kB) [chunk, still 1698 to send]
```

En cas de problème de communication, il se peut qu'un paquet de broks soit perdu, dans ce cas, une réémission est faite pour éviter cette perte, on aura un WARNING suivant:

```
[YYYY-MM-DD HH:MM:SS] WARNING: [ scheduler-name ] [ GIVE BROKS ] [ broker-master ] Packet number did mismatch "100d0ccf12de4665bf04f2150dcc97d5" != "0ca27bc3ea5440358c1194b5b7c3b4f4" : Re-sending broks (6.3kB)
```

Log de performance de la boucle du scheduler

Dans la boucle du Scheduler, les logs ci-dessous permettent de connaître le temps d'exécution passé pour chaque action.

```
[YYYY-MM-DD HH:MM:SS] INFO : [ scheduler-name ] [ SCHEDULER TIME ] [ === Loop start === ] [ Loop number=7885 ] ===-----  
[YYYY-MM-DD HH:MM:SS] INFO : [ scheduler-name ] [ SCHEDULER TIME ] took [ 0.005 ]s to schedule checks, consume results and create broks  
[YYYY-MM-DD HH:MM:SS] INFO : [ scheduler-name ] [ SCHEDULER TIME ] took [ 0.005 ]s to update items context ( downtimes, acknowledge, flapping, root_problems, business values )  
[YYYY-MM-DD HH:MM:SS] INFO : [ scheduler-name ] [ SCHEDULER TIME ] took [ 0.005 ]s to clean data ( cache, zombies, proxy, stats, notifications list )  
[YYYY-MM-DD HH:MM:SS] INFO : [ scheduler-name ] [ SCHEDULER TIME ] took [ 0.005 ]s to check environment and update stats ( satellite_thread, time, retention, orphan, modules )  
[YYYY-MM-DD HH:MM:SS] INFO : [ scheduler-name ] [ SCHEDULER TIME ] [ === Loop stop === ] [ Loop number=7884 ] [PERF] [ 0.082 ]s
```

Afin d'avoir toutes les informations de DEBUG sur les performances du scheduler (boucle principale) il faut le lancer avec:

```
SHINKEN_LOG_SCHEDULER_RECURRENT_TIMES_FLAG=1 /etc/init.d/shinken-scheduler -d start
```

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 si c'est faisable 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:

```
[YYYY-MM-DD HH:MM:SS] WARNING : [ LOGGER ]  
[YYYY-MM-DD HH:MM:SS] WARNING : [ LOGGER ]  
-----  
[YYYY-MM-DD HH:MM:SS] WARNING : [ LOGGER ] [ WRITING ] The log writes time is very high (1.87s). Please look  
at your log disk performance.  
[YYYY-MM-DD HH:MM:SS] WARNING : [ LOGGER ]  
-----  
[YYYY-MM-DD HH:MM:SS] WARNING : [ LOGGER ]
```

Logs d'erreur concernant des objets d'exécution de check qui restent en mémoire dans le Scheduler

Dans le cas où le Scheduler détecte que des objets d'exécution de checks ne sont pas bien nettoyés dans l'index de lancement (par rapport au temps).

Ceci peut résulter en deux logs d'ERROR :

- Signifie qu'un objet a été consommé, mais qu'il est peut-être encore présent dans l'index

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ scheduler-name ] [ SCHEDULING ] [ JOB-EXECUTION FAST INDEX ] The check XXXX  
is zombie without being cleaned (name=YYYY, was indexed at SSSSS). Please report to your support.
```

- Suite au premier dans la minute suivante, le Scheduler corrigera l'objet en trop.

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ scheduler-name ] [ JOB-EXECUTION FAST INDEX ] [ SCHEDULING ] [ JOB-EXECUTION  
FAST INDEX ] [PERF: lock aquired in 0.000s, done in 0.001s] Did remove 9 forgotten execution from 2 past  
seconds indexed (total number of seconds inside=63).
```

Cette situation ne devrait pas être présente, donc nous vous conseillons d'activer des logs de debug plus détaillé concernant ce mécanisme via la variable d'environnement **SHINKEN_LOG_SCHEDULER_JOB_EXECUTION_FAST_INDEX_FLAG**

```
SHINKEN_LOG_SCHEDULER_JOB_EXECUTION_FAST_INDEX_FLAG=1 /etc/init.d/shinken-scheduler -d restart
```

Nous vous conseillons de fournir ces logs détaillés à votre support pour analyse.

Logs de WARNING concernant la suppression d'objets "notification" défectueux

Sur des installations datant d'avant la 02.03.03-U01, des objets "notifications" pouvaient rester indéfiniment en mémoire dans le Scheduler, mais sans être lancés.

- Ces objets de notification sont anciens et défectueux et ils sont vus comme "late" par le check du Scheduler.
- Une mise à jour via une version ou un patch, permet de supprimer ces objets.
- Ceci sera visible sur le premier démarrage du Scheduler avec le log suivant:

```
[YYYY-MM-DD HH:MM:SS] WARNING: [ scheduler-name ] [RETENTION] [myhost] The notification [Notification  
557481183 status:scheduled command:VOID ref:unknown t_to_go:Tue Feb 2 07:49:19 2021 (creation=Fri Apr 28 02:  
19:56 2017) (is_master=False) (on myhost)] was detected as invalid (bug from the old code), and was dropped.
```

Logs de WARNING concernant les notifications qui ne sont pas envoyées à cause des périodes de notification

```
[YYYY-MM-DD HH:MM:SS] WARNING: [ scheduler-name ] [ NOTIFICATION      ] The notification for NOM DE L'HOTE OU DU CHECK was not send because notification_period NOM DE LA PERIODE DE TEMPS do not provide a date in the next 366 days ( either in the past or nothing is defined, or days are excluded ).
```

Ce log sera affiché qu'une fois par jour au max par éléments afin d'évité de stature les logs.

Log d'ERROR quand on charge depuis la rétention un nombre trop important de vérifications sur les hôtes ou les checks

Afin d'éviter un crash du Scheduler après le chargement d'une mauvaise rétention avec un check ou un hôte qui possède trop de vérifications, on ignore les vérifications et on log le message suivant :

```
[YYYY-MM-DD HH:MM:SS] ERROR   : [ scheduler-name ] [ RETENTION ] [ ANALYSE ] [ CHECKS_IN_PROGRESS ] [ monhote /check The check have too much executions (503 > warning limit=100). This can be a bug, please contact your support with a backup of your retention for verification. [ uuid=b3e7d3307d2211eca52d080027940ca8-f9c118de7d2311ec80c7080027940ca8 ]
```

Options du support Shinken

Forcer l'étalement des checks au démarrage

Si, suite à une demande du support, l'option pour forcer l'étalement des checks a été activée, le log suivant apparaîtra lors du premier chargement d'une configuration par le Scheduler:

```
[YYYY-MM-DD HH:MM:SS] INFO    : [ scheduler-name ] [ MAINTENANCE ] [ CHECK SPREAD OUT ] Scheduler will force all check spread out
```