

Scheduler - \$KEY\$ - Performance

Sommaire

Contexte

Paramétrage

- Données utilisées provenant du modèle
 - Données communes pour les checks du modèle
 - Provenant du modèle shinken
 - Provenant du modèle shinken-daemon
 - Provenant du modèle shinken-scheduler
 - Données spécifiques pour ce check
 - Les données DFE (Duplicate Foreach)
- Données utilisées provenant du check
- Données globales

Résultat

- Exemple
- Interprétation
 - Statut
 - Les vérifications communes
 - Les vérifications spécifiques
 - Résultat
 - Résultat Long
 - Description des erreurs
 - Erreur d'un démon bloqué, qui doit être redémarré
 - Un démon spare n'est assigné sur aucun démon master
 - Problème de conflits d'Arbiters
 - Les serveurs ne sont pas à la même heure
 - Erreur de surcharge des disques de logs
 - Le démon a bloqué une tentative de chargement d'objet malveillant
 - Le démon est en cours d'arrêt
 - La dernière connexion de l'Arbiter remonte à trop longtemps
 - Le démon a chargé une configuration enregistrée
 - Erreur de vol de CPU

Métriques

Contexte

La supervision d'un démon Scheduler présente un grand nombre de statistiques de performances qui permettent de visualiser le travail d'ordonnancement effectué par le Scheduler, ainsi que les statistiques des Pollers qui viennent se connecter à celui-ci.

Le démon Scheduler va effectuer tout le travail d'ordonnancement, et c'est à lui que vont s'adresser (*en autres*) les démons Poller et Reactionner pour récupérer les checks et les notifications à effectuer. Son bon fonctionnement est donc vital au bon fonctionnement de l'architecture Shinken. Aussi, pour dimensionner correctement une installation Shinken Entreprise, il est important de pouvoir visualiser combien de checks ses Pollers satellites peuvent traiter, ainsi que leurs utilisations CPU et RAM.

Les checks du Scheduler fournis dans le pack Shinken proposent donc un grand nombre de données sur les performances du Scheduler et de ses Pollers.

Statut	Nom de check	Résultat	Résultat Long																																											
		[OK] The scheduler is running well. Scheduler performance : - Average scheduler CPU usage: 0% - In the last 5 minutes, the executed checks had an average full waiting time (in Scheduler + Poller) of 0.0s before being launched - Retention save: - was successfully saved at 2023-10-25 15:45:34 (age=42s) - save time : 743ms. - Retention load: - was at 2023-10-25 15:45:34 - load time : 9s Scheduler Satellite : <table border="1"> <thead> <tr> <th rowspan="2">poller name</th> <th rowspan="2">realm</th> <th rowspan="2">tags</th> <th rowspan="2">checks todo</th> <th rowspan="2">checks done</th> <th colspan="2">CPU</th> <th>RAM</th> <th>LOAD</th> </tr> <tr> <th>available</th> <th>used by the daemon poller</th> <th>% used on the server</th> <th>on the server</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td>None</td> <td>0.00/s</td> <td>0.00/s</td> <td>Resources available</td> <td>0% (of 3 core)</td> <td>normal 60% (limit :95%)</td> <td>normal 3 Processes in the queue (limit : 12)</td> </tr> <tr> <td colspan="3">Total</td> <td>0.00/s</td> <td>0.00/s</td> <td colspan="4"></td> </tr> </tbody> </table> No reactionner found for this scheduler! Please wait reactionners will connect to the scheduler. Scheduled checks per second classified by causes : <table border="1"> <thead> <tr> <th>Causes</th> <th>Checks per second</th> </tr> </thead> <tbody> <tr> <td>dependency</td> <td>0.00/s</td> </tr> <tr> <td>retry</td> <td>0.00/s</td> </tr> <tr> <td>force</td> <td>0.00/s</td> </tr> <tr> <td>schedule</td> <td>0.00/s</td> </tr> <tr> <td>Total</td> <td>0.00/s</td> </tr> </tbody> </table>		poller name	realm	tags	checks todo	checks done	CPU		RAM	LOAD	available	used by the daemon poller	% used on the server	on the server			None	0.00/s	0.00/s	Resources available	0% (of 3 core)	normal 60% (limit :95%)	normal 3 Processes in the queue (limit : 12)	Total			0.00/s	0.00/s					Causes	Checks per second	dependency	0.00/s	retry	0.00/s	force	0.00/s	schedule	0.00/s	Total	0.00/s
poller name	realm	tags	checks todo						checks done	CPU		RAM	LOAD																																	
				available	used by the daemon poller	% used on the server	on the server																																							
		None	0.00/s	0.00/s	Resources available	0% (of 3 core)	normal 60% (limit :95%)	normal 3 Processes in the queue (limit : 12)																																						
Total			0.00/s	0.00/s																																										
Causes	Checks per second																																													
dependency	0.00/s																																													
retry	0.00/s																																													
force	0.00/s																																													
schedule	0.00/s																																													
Total	0.00/s																																													

Paramétrage

Le check utilise la ligne de commande suivante :

```
$PLUGINSDIR$/check_shinken -H "$HOSTADDRESS$" -p "$ARG2$" --shinkenversion "$SHINKENVERSION$" -t scheduler -m $ARG1$ -l "lck-$LASTSERVICECHECK$"
--passive_poller_latency "$_HOSTPASSIVE_POLLER_LATENCY$" --timeout "$_HOSTCHECK_SHINKEN_TIMEOUT$" -w "$_HOSTTHRESHOLD_CPU_STOLEN_WARNING$" -c "$_HOSTTHRESHOLD_CPU_STOLEN_CRITICAL$" --scheduler_too_old_retention_save_margin "$_HOSTSCHEDULER__RETENTION__RENTENTION-IS-TOO-OLD-AFTER_X_MINUTES$"
```

Données utilisées provenant du modèle

Données communes pour les checks du modèle

Provenant du modèle **shinken**

Nom	Modifiable sur	Unité	Défaut	Valeur par défaut à l'installation de Shinken	Description

CHECK_SHINKEN_TIMEOUT	l'Hôte (Onglet Données)	seconde	3	3	Temps maximum durant lequel le check va attendre une réponse du démon.
-----------------------	------------------------------	---------	---	---	--

Provenant du modèle **shinken-daemon**

Nom	Modifiable sur	Unité	Défaut	Valeur par défaut à l'installation de Shinken	Description
THRESHOLD_CPU_STOLEN_CRITICAL	l'Hôte (Onglet Données)	%	10	10	Seuil de CPU volé sur une machine virtuelle supervisée par VMware avant de déclencher un critique.
THRESHOLD_CPU_STOLEN_WARNING	l'Hôte (Onglet Données)	%	5	5	Seuil de CPU volé sur une machine virtuelle supervisée par VMware avant de déclencher un warning.

Provenant du modèle **shinken-scheduler**

Nom	Modifiable sur	Unité	Défaut	Valeur par défaut à l'installation de Shinken	Description
PASSIVE_POLLER_LATENCY	l'Hôte (Onglet Données)	seconde	0.5	0.5	Latence de connexion entre le Scheduler et les Pollers passifs au-delà de laquelle le check est en erreur.
SCHEDULER__RETENTION__RETENTION-ISTOO-OLD-AFTER_X_MINUTES	l'Hôte (Onglet Données)	minute	5	5	Temps additionnel de marge ajouté à l'intervalle de sauvegarde de la rétention avant qu'une rétention ancienne ne soit déclarée trop vieille et retourne en WARNING.
SCHEDULER_PORT	l'Hôte (Onglet Données)	---	7768	7768	Configuration du port de communication avec le Scheduler.
LASTSERVICECHECK	Non modifiable	---	---	---	Date au format timestamp de la dernière exécution du check.  Cette variable est évaluée uniquement par le Scheduler

Données spécifiques pour ce check

Pas de données spécifiques pour ce check.

Les données DFE (Duplicate Foreach)

Nom	Modifiable sur	Unité	Défaut	Valeur par défaut à l'installation de Shinken	Description
SCHEDULER_LIST	l'Hôte (Onglet Données)	---	scheduler-master\${_HOSTSCHEDULER_PORT}\$	scheduler-master\${_HOSTSCHEDULER_PORT}\$	Liste des Schedulers (<i>Multi-démon</i>) Check(s) impacté(s) : - Scheduler - \$KEY\$ - Running Well - Scheduler - \$KEY\$ - Performance

Données utilisées provenant du check

Pas de données provenant du check pour ce check.

Données globales

Nom	Modifiable sur	Unité	Défaut	Valeur par défaut à l'installation de Shinken	Description
HOSTADDRESS	l'Hôte (Onglet Général)	---	Nom de l'hôte	Nom de l'hôte	Adresse de l'hôte

Résultat

Exemple

Statut

Nom de check

Résultat

Résultat Long

[OK] The scheduler is running well.

Scheduler performance :

- Average scheduler CPU usage: 0%

- In the last 5 minutes, the executed checks had an average full waiting time (in Scheduler + Poller) of 0.0s before being launched

- Retention save:

- was successfully saved at 2023-10-25 15:45:34 (age=42s)

- save time : 743ms.

- Retention load:

- was at 2023-10-25 15:45:34

- load time : 9s

Scheduler Satellite :

 Scheduler - 
Performance

poller name	realm	tags	checks todo	checks done	CPU		RAM	LOAD
					available	used by the daemon poller	% used on the server	on the server
		None	0.00/s	0.00/s	Resources available	0% (of 3 core)	normal 60% (limit :95%)	normal 3 Processes in the queue (limit : 12)
Total			0.00/s	0.00/s				

No reactionner found for this scheduler! Please wait reactionners will connect to the scheduler.

Scheduled checks per second classified by causes :

Causes	Checks per second
dependency	0.00/s
retry	0.00/s
force	0.00/s
schedule	0.00/s
Total	0.00/s

Interprétation

Statut

Il peut prendre quatre valeurs **OK** / **CRITIQUE** / **ATTENTION** / **INCONNU** .

- Le statut va dépendre du retour de sonde et de la configuration spécifique du check pour les données suivantes :
 - THRESHOLD_CPU_STOLEN_CRITICAL
 - THRESHOLD_CPU_STOLEN_WARNING
 - SCHEDULER_RETENTION_RENTENTION-IS-TOO-OLD-AFTER_X_MINUTES
 - POLLER_NB_CHECK_IN_TIMEOUT_TOLERATE
 - CHECK_SHINKEN_TIMEOUT
- Voici un tableau récapitulatif du statut attendu suivant le retour de sonde.

Les vérifications communes

Situation	Statut
Démon bloqué et qui doit être redémarré	CRITIQUE
Démon spare non assigné sur un démon master	CRITIQUE
Problème de conflits d'Arbiters	CRITIQUE
Les serveurs ne sont pas à la même heure	CRITIQUE
Erreur de surcharge des disques de logs	ATTENTION
Démon qui bloque une tentative de chargement d'objet malveillant	ATTENTION

Démon est en cours d'arrêt	ATTENTION
Dernière connexion de l'Arbiter remonte à trop longtemps	ATTENTION
Le démon a chargé une configuration enregistrée, en attendant d'être contacté par l'Arbiter	ATTENTION

Les vérifications spécifiques

Situation	Statut
En fonction du pourcentage de CPU volé : Si c'est supérieur à THRESHOLD_CPU_STOLEN_CRITICAL(<i>par défaut : 10%</i>)	CRITIQUE
En fonction du pourcentage de CPU volé : Si c'est supérieur à THRESHOLD_CPU_STOLEN_WARNING(<i>par défaut : 5%</i>)	ATTENTION
Le temps de connexion à la base de données : Si c'est supérieur à SCHEDULER_RETENTION_RENTENTION-IS-TOO-OLD-AFTER_X_MINUTES (<i>par défaut : 5%</i>)	ATTENTION
Le temps de connexion à la base de données : Si c'est supérieur à POLLER_NB_CHECK_IN_TIMEOUT_TOLERATE (<i>par défaut : 0</i>)	ATTENTION
Si la sonde n'a pas eu de réponse avant le temps maximum Si supérieur à CHECK_SHINKEN_TIMEOUT (<i>par défaut : 3 sec</i>)	INCONNU

Résultat

Renvoi au format texte :

- Le nombre de checks effectués par seconde par le Poller supervisé.

Résultat Long

Dans le Résultat long du check "Poller - Performance", deux tableaux présentent des statistiques sur les temps d'exécution des checks dans le Poller.

- Le premier tableau affiche les 5 checks consommant le plus de temps CPU parmi l'ensemble des checks exécutés sur le Poller. Dans ce tableau, pour chaque check sont affichés le nom du check, l'hôte sur lequel il est accroché, et le temps d'exécution du check.

Top 5 checks :		
Host	Check	Cpu time
Enterprive DEV 1 - MASTER 1	Cpu	809ms
Enterprive DEV 1 - MASTER 1	Net	789ms
Enterprive DEV 1 - MASTER 1	Mem	769ms
Enterprive DEV 1 - MASTER 1	Io	769ms
SUPDESUP - V02.03.03-U01	VM Tools	409ms

- Un deuxième tableau présente la répartition du temps d'exécution des checks. Dans le tableau en exemple, on voit que les checks exécutés sur ce Poller s'exécutent majoritairement en moins de 50ms.

Nb Checks per CPU running time :					
0 - 50ms	50 - 100ms	100 - 200ms	200 - 300ms	300 - 400ms	+ 400ms
10473 (98%)	48 (0%)	1 (0%)	18 (0%)	98 (0%)	7 (0%)

Description des erreurs

Erreur d'un démon bloqué, qui doit être redémarré

- Si un démon est dans un état bloqué, il doit être redémarré. Si c'est le cas :
 - les checks seront en **ERROR** avec le message suivant,
 - il faut ouvrir un ticket au support pour analyser le blocage

[CRITICAL]

The daemon have a **lock**, it's **not working** and **MUST** be restarted.

Please contact your support to analyse the daemon logs:

- "Main loop" was locked more than 3600s
- Detected at 2021-12-03 08:21:55 [WATCH DOG]

Un démon spare n'est assigné sur aucun démon master

Un spare n'est pas assigné à un démon master :

Lorsqu'un démon Broker spare n'est assigné par aucun démon master, il est inutile, et remonte en erreur dans le check, tout comme dans la commande shinken-healthcheck, afin qu'il soit rapidement identifié et corrigé.

SPARE [CRITICAL]

No master is using this spare daemon → **UNUSED**

Problème de conflits d'Arbiters

- **Conflits d'Arbiters :**

Si le démon est contacté par des Arbiters qui ne sont pas sur la même architecture (*par exemple un Arbiter de Production et un autre de l'environnement de Testing*), le check sera mis en **CRITICAL** .

=> **Arbiters CONFLICT**

Architecture List :

1. Production [REDACTED]
 - arbiter-master : last connection 11s ago. Defined on the server with uuid d2a358b0-cae3-4ce8-bc3f-edd241627bac (/var/lib/shinken/server.uuid)
2. Testing [REDACTED]
 - arbiter-master : last connection 12s ago. Defined on the server with uuid 785e0227-455a-449d-b8f1-516897000e3b (/var/lib/shinken/server.uuid)

- Conflit d'Arbiters qui ont le même nom d'Architecture :

Comme dans le cas précédent, le démon est contacté par des Arbiters d'architectures différentes, mais qui ont le même nom. On sort également en **CRITICAL** mais en avertissant que les noms sont identiques, et en indiquant où changer le nom des architectures.

=> **Arbiters CONFLICT**

Architecture List :

1. Production [REDACTED]
 - arbiter-master : last connection 33s ago. Defined on the server with uuid d2a358b0-cae3-4ce8-bc3f-edd241627bac (/var/lib/shinken/server.uuid)
2. Production [REDACTED]
 - arbiter-master : last connection 10s ago. Defined on the server with uuid 785e0227-455a-449d-b8f1-516897000e3b (/var/lib/shinken/server.uuid)

NOTE:

Some architecture have the same name. We advise you to change it in the configuration of their architecture_export module.

Les serveurs ne sont pas à la même heure

Si le serveur n'est pas à la même heure que le serveur Arbiter (*qui fait office de référence*), une erreur **CRITICAL** sera levée, car des temps différents sur les différents serveurs va avoir des effets **désastreux** sur la cohérence des données de supervision.

=> Arbiters connection :

- Architecture Production :
 - **ERROR** arbiter-master => server times are different, time shift of 1 days 16h

Erreur de surcharge des disques de logs

En cas de disques trop lent sur le volume des logs, le check sera mis en **WARNING** avec l'erreur suivante.

[WARNING] The daemon have some issues.

=> **WARNING** - Writing logs on disk took too much time (worth time was 2.0s during the last minute)
Path: "/var/log/shinken/"

Le démon a bloqué une tentative de chargement d'objet malveillant

Il est possible qu'un démon puisse détecter et bloquer une tentative d'injection d'objet malveillant par le biais de l'une de ses routes.

Un message est remonté :

- le nombre total de ces tentatives que le démon a bloquées ce jour (*le compte commence à minuit*) ;
- pour chacune des tentatives (*maximum 3*) :
 - descriptif de l'objet que l'attaquant essaye de charger,
 - sa provenance de l'attaque, par exemple le nom de la route utilisée, et l'IP à la source de l'attaque,
 - sa date.

[WARNING] The daemon have some issues:

=> There were [1] security breaches blocked today (last 3):

- [hacker/attack] by [HTTP(s) call "Configuration reception from an Arbiter server" by IP=127.0.0.1] at [2022-02-21 14:44:39]

Le démon est en cours d'arrêt

Lorsque le démon est en cours d'arrêt, le check le signale, et les informations relatives aux modules ne sont plus disponibles :

[WARNING] The broker is performing a shutdown.

La dernière connexion de l'Arbiter remonte à trop longtemps

Si la dernière connexion de l'Arbiter remonte à trop de temps, le démon va lever un **WARNING** . Ceci peut être dû :

- Les Arbiters MASTER et SPARE sont réellement éteints
- Les Arbiters MASTER et SPARE sont en train d'envoyer des configurations à d'autres démons, et ne peuvent donc pas contacter ce démon pour l'instant.

=> Arbiters connection :

- Architecture *Production* :
 - arbiter-master => Missed connection from arbiter since 1 days 6h (> daemon check_interval * max_check_attempts)



Le temps pris en compte comme limite de dernière connexion est de *check_interval * max_check_attempts* du démon (*définis dans sa configuration*).

Les valeurs par défauts sont de 60s * 3, soit 3 minutes.

Le démon a chargé une configuration enregistrée

Lorsque le module *last-configuration-recorder* est actif, pour les démons de type

- Broker,
- Poller,
- Reactionner,
- Receiver et
- Scheduler,

Au démarrage, le démon charge la précédente configuration qu'il a reçue de l'Arbiter, en attendant que ce dernier le contacte.

- Il permet un redémarrage du démon **même si l'Arbiter ne peut pas joindre le démon** (*ex : coupure réseau*),
- Cela permet aussi de rendre le **redémarrage du démon plus rapide**, dans le cas où l'Arbiter **n'a pas changé de version** de configuration.

Un **WARNING** indique que le démon n'a pas encore été contacté par l'Arbiter.

Arbiters connection:

Architecture List :

1. Shinken-Production [loaded from retention]

WARNING

arbiter-master : loaded from retention 52s ago [waiting for Arbiter connection].

Erreur de vol de CPU

Seulement si la machine virtuelle est hébergée sur un hyperviseur VMWare

- Si la VM se fait voler trop de temps de calcul (CPU Stolen), le check sera mis en **WARNING** ou en **CRITIQUE** (*en fonction du taux de vol fixé par défaut ou indiqué par l'utilisateur*).
 - on peut avoir plus d'informations sur cet indicateur et comment réduire la perte de temps de la VM sur la page [Machine VMWare avec un fort taux de CPU Stolen \(%ready + %costop\)](#)

[WARNING] The daemon have some issues:

=> Your machine got **8% of CPU STOLEN** from the Hypervisor (*Type VMWare*)

→ On the VCenter search the data **CPU %ready + %costop**

→ Please have a look at the Shinken Enterprise documentation about advices to reduce it

[CRITICAL] The daemon have some issues:

=> Your machine got **20% of CPU STOLEN** from the Hypervisor (*Type VMWare*)

→ On the VCenter search the data **CPU %ready + %costop**

→ Please have a look at the Shinken Enterprise documentation about advices to reduce it

Métriques

Nom	Unité	Description
checks_todo_by_sec	---	Nombre de vérifications d'hôtes et de checks générées par seconde dans le Scheduler (<i>moyenne glissante calculée sur 1 min</i>).
nb_reactionners_in_overload	---	Nombre de Reactionners connectés à ce Scheduler en surcharge.

nb_pollers	---	Nombre de Pollers connectés à ce Scheduler.
nb_reactionners	---	Nombre de Reactionners connectés à ce Scheduler.
checks_done_by_sec	---	Nombre de résultats de vérification d'hôtes et de checks donnés par les Pollers par seconde (<i>moyenne glissante calculée sur 1 min</i>).
average_scheduler_cpu_usage	---	Durée d'un cycle de traitement du Scheduler. - Le maximum est à 100. - Plus cette valeur est haute plus cela indique une charge sur le Scheduler.
notifications_and_event_handlers_done_by_sec	---	Nombre de notifications & event handlers fait par les Reactionners par seconde (<i>moyenne glissante calculée sur 1 min</i>).
notifications_todo_by_sec	---	Nombre de notifications générées par seconde dans le Scheduler (<i>moyenne glissante calculée sur 1 min</i>).
save_retention_time	---	Durée en seconde de la dernière sauvegarde de rétention
event_handlers_todo_by_sec	---	Nombre d'event handlers générées par seconde dans le Scheduler (<i>moyenne glissante calculée sur 1 min</i>).
load_retention_time	---	Durée en seconde du dernier chargement de rétention
average_scheduler_cpu_estimate_d_overload	---	Estimation de la surcharge du Scheduler. - Si cette métrique est à zéro alors le Scheduler n'est pas en surcharge. - Si cette métrique est trop souvent supérieure à 0, c'est l'indication qu'il y a besoin d'un Scheduler supplémentaire.
nb_pollers_in_overload	---	Nombre de Pollers connectés à ce Scheduler en surcharge.

