

Poller - \$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-poller
 - 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
 - Statistiques générales sur l'exécution des checks
 - Charge du Poller
 - Utilisation du CPU
 - Utilisation de la mémoire (RAM)
 - Le nombre de processus dans la file d'attente du processeur
 - 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 Poller présente un grand nombre de statistiques de performances qui permettent de visualiser le travail effectué par le Poller. Le Poller a pour rôle d'exécuter les checks.

Il est ainsi souvent un des premiers démons que l'on regarde lorsqu'on veut visualiser combien de checks sont exécutés. Aussi, pour dimensionner correctement une installation Shinken Entreprise, il est important de pouvoir visualiser combien de checks un Poller est capable d'exécuter pour pouvoir décider d'ajouter ou retirer des Pollers de la configuration.

Les checks du Poller fournis dans le pack Shinken fournissent donc un grand nombre de données sur les performances du Poller.

Statut	Nom de check	Résultat	Résultat Long
	Poller - poller-lab-validationsw1-France - Performance	OK Poller statistics - The poller has not launched any check in the last 60 seconds. - Poller load is unavailable, please wait for data. - Average CPU running time of the checks is unavailable, please wait for data. [65% normal] Average server RAM usage. [6] Processes in the CPU Queue. normal	

Paramétrage

Le check utilise la ligne de commande suivante :

```
$PLUGINSDIR$/check_shinken -H "$HOSTADDRESS$" -p "$ARG2$" --shinkenversion "$SHINKENVERSION$" -t poller -m $ARG1$ --active_poller_latency "$_HOSTACTIVE_POLLER_LATENCY$" --check_tolerate "$_HOSTPOLLER_NB_CHECK_IN_TIMEOUT_TOLERATES$" --timeout "$_HOSTCHECK_SHINKEN_TIMEOUT$" -w "$_HOSTTHRESHOLD_CPU_STOLEN_WARNING$" -c "$_HOSTTHRESHOLD_CPU_STOLEN_CRITICAL$"
```

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-poller**

Nom	Modifiable sur	Unité	Défaut	Valeur par défaut à l'installation de Shinken	Description
POLLER_PORT	l'Hôte (Onglet Données)	---	0	7771	Port utilisé pour établir la connexion avec le Poller.
NB_CHECK_IN_TIMEOUT_TOLERATE	l'Hôte (Onglet Données)	---	0	0	Nombre de checks en timeout provoquant une sortie en erreur du check.
POLLER_NB_CHECK_IN_TIMEOUT_TOLERATE	l'Hôte (Onglet Données)	---	\$_HOSTNB_CHECK_IN_TIMEOUT_TOLERATE\$	\$_HOSTNB_CHECK_IN_TIMEOUT_TOLERATES\$	Nombre de checks en timeout provoquant une sortie en erreur du check.

ACTIVE_POLLER_LATENCY	l'Hôte (Onglet Données)	seconde	0.5	0.5	Latence de connexion au-delà de laquelle le check sort en erreur.
-----------------------	------------------------------	---------	-----	-----	---

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
POLLER_LIST	l'Hôte (Onglet Données)	---	poller-master\$(\$_HOSTPOLLER_PORT\$)\$	poller-master\$(\$_HOSTPOLLER_PORT\$)\$	Liste de Pollers. Check(s) impacté(s) : - Poller - \$KEY\$ - Running Well - Poller - \$KEY\$ - Performance Note : Cette valeur remplacera dans la commande la valeur \$ARG2\$

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
	Poller - poller-lab-validationsw1-France - Performance	[OK] Poller statistics - The poller has not launched any check in the last 60 seconds. - Poller load is unavailable, please wait for data. - Average CPU running time of the checks is unavailable, please wait for data. [65% normal] Average server RAM usage. [6] Processes in the CPU Queue.	

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
 - 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
Le temps de connexion à la base de données : • Si c'est supérieur à THRESHOLD_CPU_STOLEN_CRITICAL (<i>par défaut : 10%</i>)	CRITIQUE
Le temps de connexion à la base de données : • Si c'est supérieur à THRESHOLD_CPU_STOLEN_WARNING (<i>par défaut : 5%</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

Statistiques générales sur l'exécution des checks

La première statistique remontée par le check est le nombre de checks effectués par seconde par le Poller supervisé.

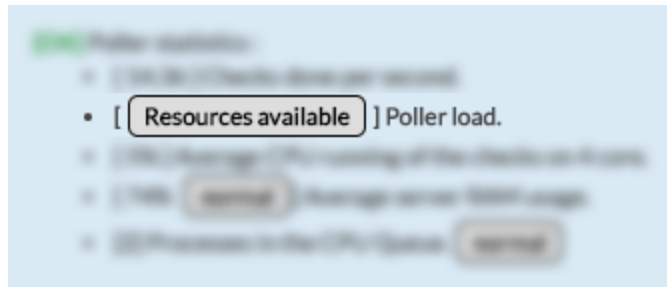
Dans cet exemple, le Poller effectue 1.9 check par seconde en moyenne, tous checks confondus.

[OK] Poller statistics :

- **[1.9]** Checks done per second.

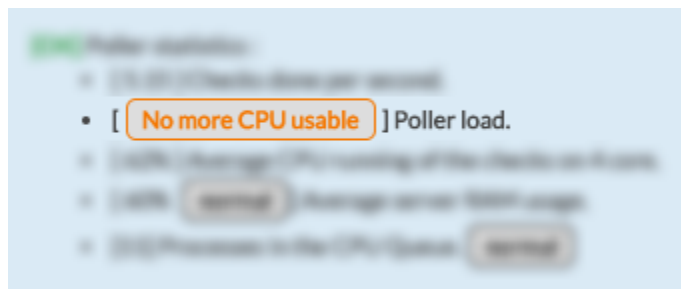
Charge du Poller

Le check du Poller fournit aussi la charge du Poller. Il s'agit d'un indicateur général indiquant si le Poller peut encore supporter des checks supplémentaires, ou s'il est chargé au maximum. Cet indicateur n'est pas lié aux autres indicateurs de performances de la machine (File d'attente CPU, mémoire).



Une pastille orange précédant la mention "Poller load" signifie que le Poller ne peut plus prendre de checks supplémentaires. Cette pastille indique que le Poller fonctionne à la vitesse maximale qui lui est permise par le processeur.

C'est donc un signe indiquant qu'il faudrait ajouter un Poller supplémentaire dans l'architecture Shinken.

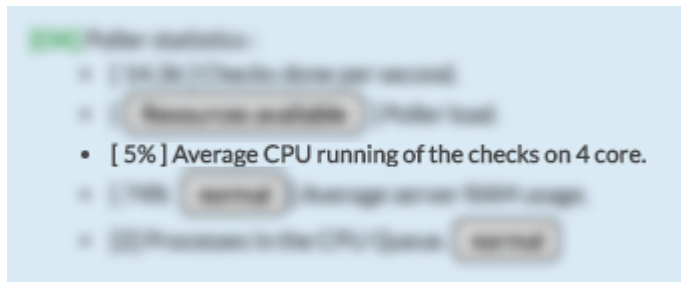


Utilisation du CPU

Le Poller se limite intelligemment selon différents critères pour éviter de surcharger inutilement la machine sur laquelle il est exécuté. Le premier critère de limitation est l'utilisation du CPU de la machine par le démon (*et l'exécution des checks*).

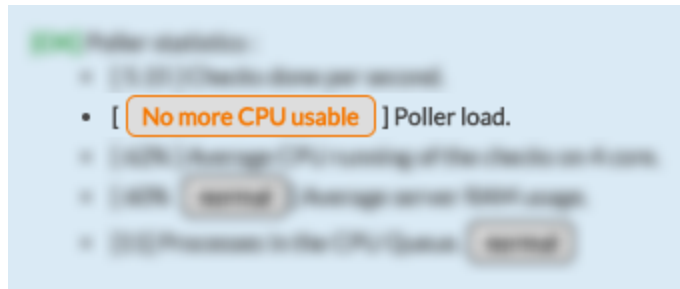
Si le Poller se rend compte que le CPU de la machine sur laquelle il est exécuté est surchargé, il se régule pour exécuter moins de checks et éviter de rendre la machine inutilisable.

Dans le check "*Poller - Performance*", la quantité de CPU utilisée par les checks est affichée dans le Résultat du check.



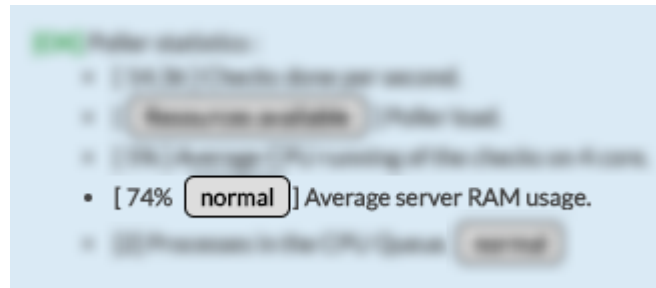
Si le CPU de la machine hébergeant le Poller est utilisé de manière intensive, le résultat du check "*Poller - Performance*" avertit de l'utilisation trop élevée du CPU. Dans ce cas, le Poller se limite et n'exécute plus de checks supplémentaires tant que l'utilisation du CPU est trop élevée.

La ligne indiquant la charge du Poller indique alors que le CPU est surchargé.

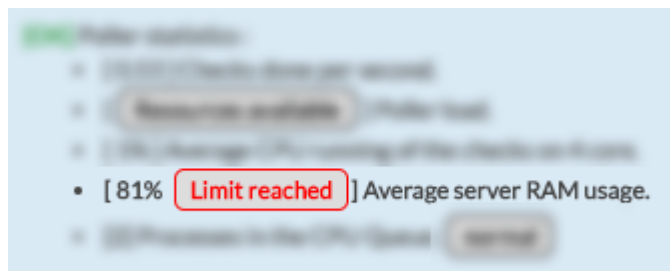


Utilisation de la mémoire (RAM)

Si le CPU n'est pas le facteur limitant pour le fonctionnement du Poller, l'état de la mémoire du serveur est alors vérifié. Dans le résultat du check "Poller - Performance", l'utilisation de la mémoire sur le serveur est indiquée.



Si l'utilisation de la mémoire (RAM) sur le serveur dépasse le seuil défini dans la configuration du Poller, le résultat du check "Poller - Performance" affiche un avertissement indiquant l'utilisation excessive de la mémoire. Lorsque cet avertissement est affiché, le Poller n'exécute plus de checks supplémentaires tant que l'utilisation de la mémoire est supérieure au seuil défini.



Comme mentionné précédemment, le réglage du seuil est effectué dans le fichier de configuration du Poller concerné.

/etc/shinken/pollers/mon_poller.cfg

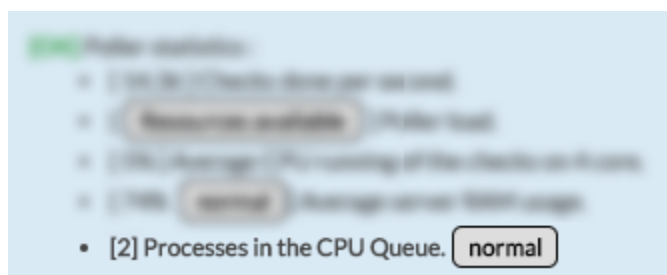
```
define poller {
...
# Percentage of used physical RAM beyond which the poller will not launch any new check
max_ram_percent      95
...
}
```

Le nombre de processus dans la file d'attente du processeur

Le Poller possède une dernière mesure de performance qu'il observe et selon laquelle il limite son activité. Après la vérification de l'utilisation du CPU du Poller et de la mémoire du serveur, le Poller vérifie le nombre de processus dans la file d'attente du processeur.

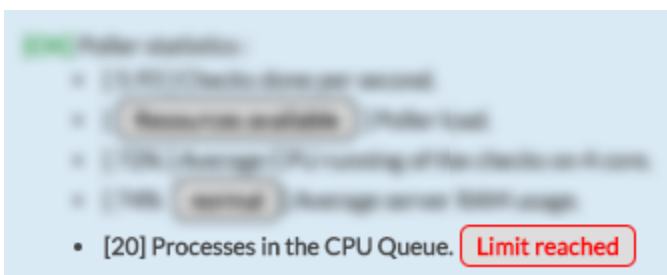
En fonction de la limite de processus dans la file d'attente du processeur, le Poller décide s'il faut exécuter des checks supplémentaires.

Le nombre de processus dans la file d'attente du processeur est affiché dans le résultat du check "Poller - Performance".



Le nombre de processus dans la file d'attente du processeur est aussi accompagné d'une pastille indiquant le statut de cette mesure. Lorsque le nombre de processus est supérieur à la limite définie dans la configuration, cette pastille indique ce dépassement.

Dans ce cas, le Poller n'exécutera plus de checks supplémentaires tant que le nombre de processus dans la file d'attente du processeur sera supérieur au seuil choisi.



Le seuil mentionné précédemment se définit dans le fichier de configuration du Poller concerné. On note que ce seuil se définit par cœur du processeur, et pas globalement pour le processeur.

Ainsi, avec une limite définie à 3, et un processeur possédant 4 cœurs, le Poller ne se limitera que lorsqu'il y a plus de 12 processus dans la file d'attente du processeur.

/etc/shinken/pollers/mon_poller.cfg

```
define poller {
...
# Number of maximum runnings processes in the CPU Queue per CPU. Default value is 4.
max_cpu_queue_per_cpu    4
...
}
```

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
Enterprise DEV 1 - MASTER 1	Cpu	809ms
Enterprise DEV 1 - MASTER 1	Net	789ms
Enterprise DEV 1 - MASTER 1	Mem	769ms
Enterprise DEV 1 - MASTER 1	Io	769ms
SUPDESUP - V02.03.03-U01	VM Tools	409ms

Nb Checks per CPU running time :

0 - 50ms	50 - 100ms	100 - 200ms	200 - 300ms	300 - 400ms	+ 400ms
10473	48	1	18	98	7
(98%)	(0%)	(0%)	(0%)	(0%)	(0%)

- 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.

Top 5 checks :

Host	Check	Cpu time
Enterprise DEV 1 - MASTER 1	Cpu	809ms
Enterprise DEV 1 - MASTER 1	Net	789ms
Enterprise DEV 1 - MASTER 1	Mem	769ms
Enterprise DEV 1 - MASTER 1	Io	769ms
SUPDESUP - V02.03.03-U01	VM Tools	409ms

Nb Checks per CPU running time :

0 - 50ms	50 - 100ms	100 - 200ms	200 - 300ms	300 - 400ms	+ 400ms
10473	48	1	18	98	7
(98%)	(0%)	(0%)	(0%)	(0%)	(0%)

- Si le temps moyen d'exécution des checks est très long, ou très court, il est possible de modifier les réglages des périodes de temps du tableau dans le fichier de configuration du Poller concerné. Dans le fichier de configuration du Poller en question, l'option "exec_stat_range" doit être modifiée.

/etc/shinken/pollers/mon_poller.cfg

```
define poller {  
    ...  
    # Ranges for the check : poller statistics  
    #exec_stat_range    50, 100, 200, 300, 400, 1000, 5000, 15000  
    ...  
}
```

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 [194.168.1.107]
 - arbiter-master : last connection 11s ago. Defined on the server with uuid d2a358b0-cae3-4ce8-bc3f-edd241627bac (/var/lib/shinken/server.uuid)
2. Testing [194.168.1.107]
 - 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 [194.168.1.107]
 - arbiter-master : last connection 33s ago. Defined on the server with uuid d2a358b0-cae3-4ce8-bc3f-edd241627bac (/var/lib/shinken/server.uuid)
2. Production [194.168.1.107]
 - 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
used_ram_percent	---	Pourcentage de mémoire utilisée sur le serveur

cpu_usage_percent	---	Pourcentage de temps processeur utilisé par le Poller
poller_load_state	---	Booléen (<i>0 ou 100</i>) qui indique si le Poller a atteint sa charge maximale : • 0 : Le Poller n'a pas atteint sa charge maximale, il peut encore exécuter des checks supplémentaires • 100 : Le Poller a atteint sa charge maximale, il ne peut pas exécuter plus de checks.
nb_action_done_per_sec	---	Nombre moyen de checks exécutés par le Poller (<i>moyenne glissante calculée sur 1mn</i>)
cpu_running_queue	---	Queue moyenne de checks exécutés par le Poller (<i>moyenne glissante calculée sur 1mn</i>)
checks_per_cpu_running_time_X_Y_ms	---	Ces métriques représentent la répartition du temps d'exécution des checks, le nombre de métriques générées va dépendre de la configuration du Poller, comme expliqué dans l'onglet Résultat Long

