

Machine VMWare avec un fort taux de CPU Stolen (%ready + %costop)

Sommaire

Signification du CPU Stolen
Composantes du CPU Stolen
CPU Stolen: le temps perdu par la VM par rapport au cas où la machine aurait été physique sans ESXi
Voir les indicateurs de perte de temps dans VSphere
Calcul des ready et costop en pourcentage à partir des données dans VSphere
Valeurs d'avertissement et critiques concernant le CPU Stolen
Pistes de résolution d'un taux de CPU Stolen élevé
Cas important du sizing de l'ESXi avec de l'hyperthreading : gain de 20% de performance seulement (pas 100%)

Rôle

Le Scheduler gère la distribution des checks et actions vers les Pollers et Schedulers respectifs. Le Scheduler est également responsable du traitement et de l'analyse des résultats de checks, faisant la corrélation et prenant les actions adéquates (si un check est critique, il va demander une vérification de son hôte).

Il ne lance pas de checks ou de notification, il garde juste la liste d'attente pour les autres démons de l'architecture. Cela permet de gérer la charge entre plusieurs Pollers. Il peut également y avoir plusieurs Schedulers pour des questions de load-balancing ou de rôle de standby à chaud. La persistance des statuts est faite grâce au module de rétention (Voir la page [La rétention des données des Schedulers](#)).

Connexions aux autres démons

Le Scheduler ouvre le port 7768. Il recevra sa configuration de l'Arbiter à chaque changement par ce port.

Il recevra également sur ce port les tâches venant des autres démons (Poller, Scheduler et Broker). Les autres démons consomment les données et actions depuis le Scheduler.

Les connexions du Scheduler

Dans un cas spécifique, le Scheduler ouvre des connexions vers les autres démons. Cela arrive lorsqu'il y a un Poller passif (Voir la page [Le Poller](#)).

? Unknown Attachment

Données

Ce démon stocke la totalité de la configuration. Il a accès à la liste de tous les contacts, mais seulement à une partie des hôtes du royaume.

Résumé des connexions

Source	Destination	Port	Protocole	Note
Poller	Scheduler	7768	HTTP/HTTPS	
Scheduler	Poller	7771	HTTP/HTTPS	Only if there is a passive poller.

Description des variables

Propriété	Défaut	Description
scheduler_name	N/A	Cette variable est utilisée pour définir le nom raccourci du Scheduler auquel les données sont attachées.
address	N/A	Cette directive est utilisée pour définir l'adresse permettant à l'Arbiter de joindre ce Scheduler. Par défaut "localhost", changez-le par un nom DNS ou une adresse IP.
port	7768	Cette directive est utilisée pour définir le port TCP utilisé par le démon.
use_ssl	0	Cette variable est utilisée pour définir si le Scheduler doit être contacté en HTTPS (*1*) ou HTTP (*0*). La valeur par défaut est *0* (HTTP).
spare	0	Cette variable est utilisée pour définir si le Scheduler peut être géré comme un spare (prendra uniquement la configuration si le maître échoue). La valeur par défaut est *0* (maître).

timeout	3	Cette variable est utilisée pour définir le temps en secondes avant que les autres démons ne considèrent ce démon comme ayant raté un Ping. Si ce démon est joignable en HTTPS (use_ssl à 1) avec une latence élevée, nous vous conseillons alors d'augmenter cette valeur de timeout (l'Arbiter aura besoin de plus d'allers/retours pour le contacter).
data_timeout	120	Cette variable est utilisée pour définir le temps en secondes avant de considérer un transfert de configuration ou de données comme échoué.
max_check_attempts	3	Si le Ping permettant de détecter la disponibilité réseau du nœud est en échec N fois ou plus, alors le nœud est considéré comme mort. (par défaut, 3 tentatives)
check_interval	60	Intervalle de Ping toutes les N secondes.
satellitemap	N/A	Cette variable est utilisée dans le cas de royaume situé derrière un réseau NATé. Elle est de la forme d'une liste séparée par des "," de valeur nom-démon=address:port Les démons ainsi listés seront contactés avec le couple address:port du paramètre au lieu de leur adresse dans leur .cfg.* Ceci permet ainsi à des démons derrière un réseau NAT d'échanger sur leur adresse locale au lieu de devoir ressortir sur leur adresse publique. <i>Exemple: daemon1=192.168.0.1:7768,daemon2=192.168.0.1:7771</i>
realm	N/A	Cette variable est utilisée pour définir le royaume où le Scheduler doit être. Si aucun n'est sélectionné, celui par défaut lui sera assigné.
modules	N/A	Cette variable est utilisée pour définir les modules que le Scheduler chargera.
scheduler_export_data_enabled	1	S'il est désactivé, aucune extraction d'ordonnancement des checks ne sera autorisée, même en anonyme. La commande pour l'extraction de données sera fournie dans une prochaine version. Pour le moment, les appels d'extraction de donnée ne sont utilisés qu'à des fins de support.
enabled	N/A	Cette variable est utilisée pour définir si le Scheduler est activé ou non.

Exemple de définition

⚠ Il est conseillé d'éditer les fichiers .cfg avec l'encodage utf-8

```

=====
# SCHEDULER
=====
# The scheduler is a "Host manager". It gets the hosts and their services,
# schedules the checks and transmit them to the pollers.
# Description: The scheduler is responsible for:
# - Scheduling checks
# - Computing states
# - Enqueing jobs
=====

define scheduler {

    #===== Daemon name and address =====
    # Daemon name. Must be unique
    scheduler_name          scheduler-master

    # IP/fqdn of this daemon (note: you MUST change it by the real ip/fqdn of this server)
    address                 localhost

    # Port (HTTP/HTTPS) exposed by this daemon
    port                   7768

    # 0 = use HTTP, 1 = use HTTPS
    use_ssl                 0

    #===== Master or spare selection =====
    # 1 = is a spare, 0 = is not a spare
    spare                   0

    #===== Daemon connection timeout and down state limit =====
    # timeout: how many seconds to consider a node don't answer

```

```
timeout                3

# data_timeout: how many second to consider a configuration transfert to be failed
# because the network brandwith is too small.
data_timeout          120

# max_check_attempts: how many fail check to consider this daemon as DEAD
max_check_attempts    3

# Check this daemon every X seconds
check_interval        60

##### Modules to enable for this daemon #####
# Exactly one retention module can be chosen. No modules or more than one retention will result in a
configuration error displayed in the output of the shinken-healthcheck command.
# Available:
# - PickleRetentionFile : (if you have only one scheduler into a realm) save retention data (element
state and scheduling) into a file
# - MongodbRetention   : (if you have more than one scheduler into a realm) save retention data
(element state and scheduling) into a mongodb database
modules                PickleRetentionFile

##### Realm and architecture settings #####
# Realm to set this daemon into
realm                  All

# In NATted environments, you declare each satellite ip[:port] as seen by
# *this* scheduler (if port not set, the port declared by satellite itself
# is used)
#satellitemap          poller-1=1.2.3.4:1772, reactionner-1=1.2.3.5:1773

# ===== Scheduler data export =====
# scheduler__export_data__enabled: (default 1) If disabled, no data dump will be allowed, even anonymous
ones
# scheduler__export_data__enabled      1

##### Enable or not this daemon #####
# 1 = is enabled, 0 = is disabled
enabled                    1

}
```