

# Les logs du Poller

## Sommaire

- Démarrage du démon
- Gestion de la configuration
  - Premier chargement de la configuration
  - Mise à jour de la configuration
  - Mise à jour des liens vers d'autres démons
- Récupération des checks
  - Pour récupérer les checks à exécuter
  - Envoi des résultats de checks au Scheduler
- Surcharge serveur en activité disque, ralentissant l'écriture des logs
- Calcul des statistiques nécessaire au check Poller - \$KEY\$ - Performance du pack Shinken
- Suivi des lancement de sondes par les workers et de leur performances
  - Les Workers arrivent à lancer toutes leurs sondes ou se limitent
  - PROBLEME: Un worker n'arrive pas à lancer ses checks
  - Le total des sondes et leurs performances au sein du poller
- Échanges entre le processus principal et les Workers
  - Suivi des actions en cours
    - Dans le démon (boucle principale)
    - Dans un Worker
  - Problème temporaire de réception de résultats depuis les workers
- Log de performance de la boucle du Poller
  - Performance des boucles des Workers
    - Un Worker est lent
    - Un Worker est en retard

## Démarrage du démon

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

- ses limites systèmes en nombre de fichiers ouvrables, et nombre de threads/processus

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

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

## Gestion de la configuration

### Premier chargement de la configuration

Lorsque le Poller reçoit sa configuration pour la première fois deux logs **INFO** sont affichés.

- Le premier indiquant que nous rentrons dans la phase de chargement d'une nouvelle configuration.

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

- Le deuxième indiquant que nous avons reçu la configuration de l'Arbiter.

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ CONFIGURATION ] The arbiter send us a new
configuration: [configuration_uuid=configuration-uuid, arbiter=arbiter-name,
architecture=architecture-name, date=YYYY-MM-DD HH:MM:SS]
```

-  Dans le cas où le poller 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 : [ poller-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 poller n'est pas de la même version que l'arbitre et que l'option \*mismatch\_version\_error\* est désactivé sur l'arbitre:

```
[YYYY-MM-DD HH:MM:SS] WARNING : [ poller-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].
```

## Mise à jour de la configuration

Lorsque qu'il y a une mise à jour de la configuration, deux logs en **INFO** sont affichés.

- Le premier indiquant que nous rentrons dans la phase de chargement d'une nouvelle configuration.

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ CONFIGURATION ] [ UPDATE ] ----- Loading a configuration update from the arbiter
```

- Le deuxième indiquant que nous avons reçu la configuration de l'Arbitre.

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ CONFIGURATION ] [ UPDATE ] The arbiter send us a new configuration: [configuration_uuid=configuration-uuid, arbiter=arbiter-name, architecture=architecture-name, date=YYYY-MM-DD HH:MM:SS]
```

- ❌ Dans le cas où le poller n'est pas de la même version que l'arbitre et que l'option \*mismatch\_version\_error\* est activé sur l'arbitre:

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ poller-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 poller n'est pas de la même version que l'arbitre et que l'option \*mismatch\_version\_error\* est désactivé sur l'arbitre:

```
[YYYY-MM-DD HH:MM:SS] WARNING : [ poller-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].
```

## Mise à jour des liens vers d'autres démons

Lorsque que l'Arbitre détecte un changement de lien entre les démons quatre logs en **INFO** seront affichés.

- Les deux premiers logs affichent le(les) lien(s) du(des) démon(s) supprimé(s).

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ CONFIGURATION ] The arbiter asked us to remove daemons:
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ CONFIGURATION ] - REMOVED scheduler : [name=scheduler1-name]
[shard_id= XXX] [uri=http://scheduler_address:port/]
```

- Les deux premiers logs affichent le(les) lien(s) du(des) démon(s) ajouté(s).

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ CONFIGURATION ] The arbiter send us new daemons:
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ CONFIGURATION ] + ADDED scheduler : [name=scheduler2-name]
[shard_id= XXX] [uri=http://scheduler_address:port/]
```

## Récupération des checks

### Pour récupérer les checks à exécuter

- **Poller actif**
  - Le Poller va demander au Scheduler.
  - Le Poller indique un temps de travail qu'il a de disponible ( en temps cpu ).
  - Le Scheduler lui donne des checks pour le temps disponible ( suivant le temps d'exécution moyen pour les checks constatés sur ce poller ). Il lui donne pour un temps inférieur ou égal au temps disponible.
  - Un log permet d'avoir le nombre de checks récupérés. Ce log s'affiche même si aucun check n'a été récupéré :

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ CHECKS ] [ scheduler-master ] [ GET ] Requesting checks
todo from this scheduler for 2.000s cpu time [received=3 check(s) for 0.039s cpu time]
```

- **Poller passif**

- Le Scheduler demande au Poller le temps CPU disponible.
- Le Scheduler lui envoie des checks à traiter pour le temps disponible ( suivant le temps d'exécution moyen pour les checks constatés sur ce Poller ). Il lui donne pour un temps inférieur ou égal au temps disponible.
- Si des checks sont reçus, un log permet d'avoir le nombre de checks récupérés selon le temps de travail disponible sur le Poller :

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ CHECKS ] [ scheduler-master ] [ RECEIVED ] We received
checks todo from this scheduler for 2.000s cpu time [received=1 check(s) for 0.317s cpu time]
```

## Envoi des résultats de checks au Scheduler

- **Poller actif**

- Une fois les commandes exécutées, le Poller envoie les résultats au Scheduler.
- Un log permet d'avoir le nombre de résultats de checks envoyés au Scheduler et le temps mis pour être envoyé.

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ CHECKS RESULTS ] [scheduler-master] [ PUSHED ] 1 check's
result(s) sends to this scheduler in [0.043]s
```

- **Poller passif**

- Une fois les checks exécutés, le Poller stocke les résultats en attendant que le Scheduler vienne les récupérer.
- A chaque tour de boucle du Scheduler, ce dernier demande au Poller s'il a des résultats de checks disponibles.
- Si des résultats sont disponibles, un log permet d'avoir le nombre de résultats checks donnés au Scheduler.

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ CHECKS RESULTS ] [scheduler-master] [ GIVEN ] 1 check's
result(s) given to answer scheduler request
```

## 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 ]
```

## Calcul des statistiques nécessaire au check Poller - \$KEY\$ - Performance du pack Shinken

Quand un check de supervision du démon est fait, on va avoir plusieurs entrées dans les logs qui concernent des données que le démon garde sur diverses statistiques.

Un log permet d'avoir le temps pris sur le calcul des dernières commandes en timeout:

```
[YYYY-MM-DD HH:MM:SS] DEBUG : [ poller-name ] [ STATS ] Compute "Checks in timeouts" stats : 0.000s
in a total of 2048 commands in timeouts
```

Un log permet d'avoir le temps de calcul concernant les ranges d'exécution des checks/notifications en fonction du temps (<100ms, <400ms, etc):

```
[YYYY-MM-DD HH:MM:SS] DEBUG : [ poller-name ] [ STATS ] Compute "Checks per CPU running time" : 0.000s (on a total of 2048 checks)
```

Un log permet d'avoir le temps de calcul pour avoir les 5 commandes les plus longues en temps CPU:

```
[YYYY-MM-DD HH:MM:SS] DEBUG : [ poller-name ] [ STATS ] top5 execution time 0.003s (loop over 1 ranges and 343 elements)
```

Un dernier log permet d'avoir le temps complet du calcul des statistiques du démon:

- sur une partie commune à tous les démons (version, chemins, etc)
- sur la partie qui concerne uniquement ce qui concerne ce démon, donc sont inclus ici les temps précédents
- il est affiché en :
  - **DEBUG**: si le temps de calcul est inférieur à la valeur du paramètre `display_statistics_compute_time_if_higher` du démon.
  - **INFO**: si le temps de calcul est supérieur ou égal au paramètre `display_statistics_compute_time_if_higher` du démon.

```
[YYYY-MM-DD HH:MM:SS] DEBUG : [ poller-name ] [ STATS ] Daemon stats were computed in 0.020s (0.001 for daemon common part, 0.020 for poller part)
```

En cas d'affichage INFO on met un petit morceau en plus sur comment gérer le niveau de log:

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ STATS ] Daemon stats were computed in 0.004s (0.000 for daemon common part, 0.004 for poller part) (NOTE: log is displayed in INFO because 0.004 is higher than display_statistics_compute_time_if_higher=1ms in the daemon cfg)
```

## Suivi des lancement de sondes par les workers et de leur performances

### Les Workers arrivent à lancer toutes leurs sondes ou se limitent

On peut savoir si les workers manquent de disponibilité CPU/RAM/Load average en suivant les logs.

Si on a manqué de performance ce tour (CPU, RAM ou load average trop élevé, indiqué), alors on aura le log suivant:

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ ALL WORKERS ] [ MISSING RESSOURCE ] Must launch: 2 checks ( Expected CPU Time: 1.318s ) => Launched: 1/2 checks => Wait 0.530s CPU/Memory availability
```

Sinon ils ont pu tout lancer, on aura le log suivant:

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ ALL WORKERS ] Launched all 2/2 checks ( Expected CPU Time: 1.318s )
```

### PROBLEME: Un worker n'arrive pas à lancer ses checks

Si le worker n'a pas réussi à lancer de check pendant plus de 5 seconds on aura le log suivant:

```
[YYYY-MM-DD HH:MM:SS] WARNING : [ poller-name ] [worker-fork] [WORKER_ID] is full since X. It has Y checks pending.
```

ou :

- `WORKER_ID` : est l'id du worker

- X : est la durée depuis lequel le worker n'a pas lancer de checks
- Y : est le nombre de checks à lancer

## Le total des sondes et leurs performances au sein du poller

On peut suivre les sondes au sein du Poller via le log suivant, pour les checks provenant des Schedulers & Synchronizers ( cumulés, pour l'instant on n'a pas l'information sur toute la chaine du Poller ):

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-master ] [ PERFS ] [ FOR SCHEDULERS/SYNCHRONIZERS ] [ THIS TURN ] [ Daemon ] [ Waiting to be push to workers ] 8 checks, Expected CPU Time: 1.06s
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-master ] [ PERFS ] [ THIS TURN ] => [ Workers ] [ New launch = 9 checks, Expected CPU Time: 1.10s ] [ Executing = 6 checks, Expected CPU Time: 1.601s ] [ Just finished = 7 checks, Consumed CPU Time: 0.62s, +0.03s from Expected CPU Time ]
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-master ] [ PERFS ] [ THIS TURN ] [ Daemon ] [ Waiting to be returned ] 6 checks, Consumed CPU Time: 1.06s
```

Avec :

Log	Description
Expected CPU Time	Temps CPU pur ( sans les appels réseaux et autres sleep ) qu'on a mesuré en moyenne sur ce check.
Consumed CPU Time	Vrai temps CPU mesuré sur CE dernier lancement.
+0.03s from Expected CPU Time	Différence de temps CPU ( en + ou en - ) qu'on a eu par rapport à leurs exécutions moyennes passées.
Waiting to be push to workers	Dans le processus maitre, avant d'être envoyé aux workers.
New launch	Dans le worker, en attente d'être lancé.
Executing	Dans le worker, en cours d'exécution ( lancé ce tour, ou un tout d'avant ).
Just finished	Fini dans ce tour-ci.
Waiting to be returned	Dans le processus maitre, en attente d'être renvoyés vers les Schedulers/Synchronizers.

Et les mêmes informations, mais sur une moyenne d'une minute glissantes :

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-master ] [ PERFS ] [ FOR SCHEDULERS/SYNCHRONIZERS ] [ 1min AVERAGE ] [ Daemon ] [ Waiting to be push to workers ] 14.42 checks/s, Expected CPU Time: 2.11s
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-master ] [ PERFS ] [ 1min AVERAGE ] => [ Workers ] [ New launch = 5.98 checks/s, Expected CPU Time: 0.91s ] [ Executing = 5.10 checks/s, Expected CPU Time: 0.933s ] [ Just finished = 5.78 checks/s, Consumed CPU Time (total): 1.35s, +0.00s from Expected CPU Time ]
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-master ] [ PERFS ] [ 1min AVERAGE ] [ Daemon ] [ Waiting to be returned ] 5.80 checks/s, Consumed CPU Time: 0.88s
```

Note : on doit avoir des valeurs sensiblement équivalentes entre les valeurs suivantes:

- New launch
- Just finished
- Waiting to be returned

Et ils représentent le débit du Poller.

## Échanges entre le processus principal et les Workers

### Suivi des actions en cours

#### Dans le démon (boucle principale)

Le log suivant indique

- le nombre d'actions présentes dans le démon principal
- le nombre d'actions traitées sur ce tour de boucle ainsi que le nombre total d'actions présentes par Worker
- le nombre total d'actions en cours de gestion (démon + Workers)

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-master ] [ STATS ] action in main daemon to be dispatched to workers:
[ XX ], distribution by worker [ WORKER XX: XX done this turn / XX total pending ] [ WORKER XX: XX done this
turn / XX total pending ] total: [ XX ]
```

## Dans un Worker

Le log suivant indique le nombre d'actions présentes dans le Worker, ainsi qu'une estimation du temps CPU nécessaire à leur exécution

```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-master ] [WORKER XX][COMMAND TO PROCESS] Worker load_todo_actions : X.
XXX nb_action : XX
```

## Problème temporaire de réception de résultats depuis les workers

Il se peut que si un worker est chargé, il mette trop de temps à nous renvoyer un résultat, dans ce cas on aura le log WARNING suivant:

```
[YYYY-MM-DD HH:MM:SS] WARNING : [ poller-name ] The worker 2 reception did fail this turn (IOError), skip to
the next turn to receive more.
```

La vérification du bon fonctionnement des workers est faite lors de l'envoi des checks, donc en cas de problème worker mort il sera détecté.

- Si ces logs WARNING sont sporadiques, ils ne sont pas un problème, les résultats sont juste récupérés à la seconde suivante
- Si ces logs WARNING sont constants, ceci signifie que le worker est surchargé, et alors le mécanisme de surcharge doit être activé également

## Log de performance de la boucle du Poller

Les logs suivants permettent de suivre le temps d'exécution de la boucle principale du Poller


```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ POLLER TIME ] [ === Loop start === ] [ Loop number=XXX ]
=====
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ POLLER TIME ] [ === Loop stop === ] [ Loop number=XXX ]
[PERF] [ X.XXX ]s
```

## Performance des boucles des Workers

Le log suivant permet de suivre l'activité de chaque Worker et de s'assurer que chacun continue de tourner


```
[YYYY-MM-DD HH:MM:SS] INFO : [ poller-name ] [ POLLER TIME ] [ WORKERS ] Last activity [ WORKER X: X.XXXs
ago, WORKER Y: Y.YYYs ago]
```

## Un Worker est lent

 Quand le tour de boucle d'un Worker n'a pas fini avant un certain délai, le log suivant signale la lenteur observée

```
[YYYY-MM-DD HH:MM:SS] WARNING : [ poller-name ] [ POLLER TIME ] [ WORKER X ] is slow, last tick was X.XXXs
ago, over limit of X.XXXs
```

## Un Worker est en retard

 Quand le tour de boucle d'un Worker prend beaucoup trop de temps, le log suivant signale le retard observé

```
[YYYY-MM-DD HH:MM:SS] ERROR : [ poller-name ] [ POLLER TIME ] [ WORKER X ] is late, last tick was X.XXXs
ago, over limit of X.XXXs
```