Architecture distribuée de Shinken Enterprise pour le load balancing

Le load balancing est très facile à mettre ne place avec Shinken.Enterprise

Si vous utilisez le load balancing, la charge est principalement sur 2 process : :

- pollers: ils lancent les checks, et utilisent beaucoup de ressource CPU
- schedulers: ils planifient, potentiellement beaucoup de checks

Pour les 2, une limite de 15000 checks/5min est une moyenne raisonnable sur un serveur standard (4 cores\@3Ghz). On peut scaler horizontalement en ajoutant simplement plus de serveurs et en les déclarant comme pollers ou scheduleers. .

Attention, le scheduler N'EST PAS un process multi-threaded process, donc si vous rajouter des coeurs, ça ne changera rien à ses performances.

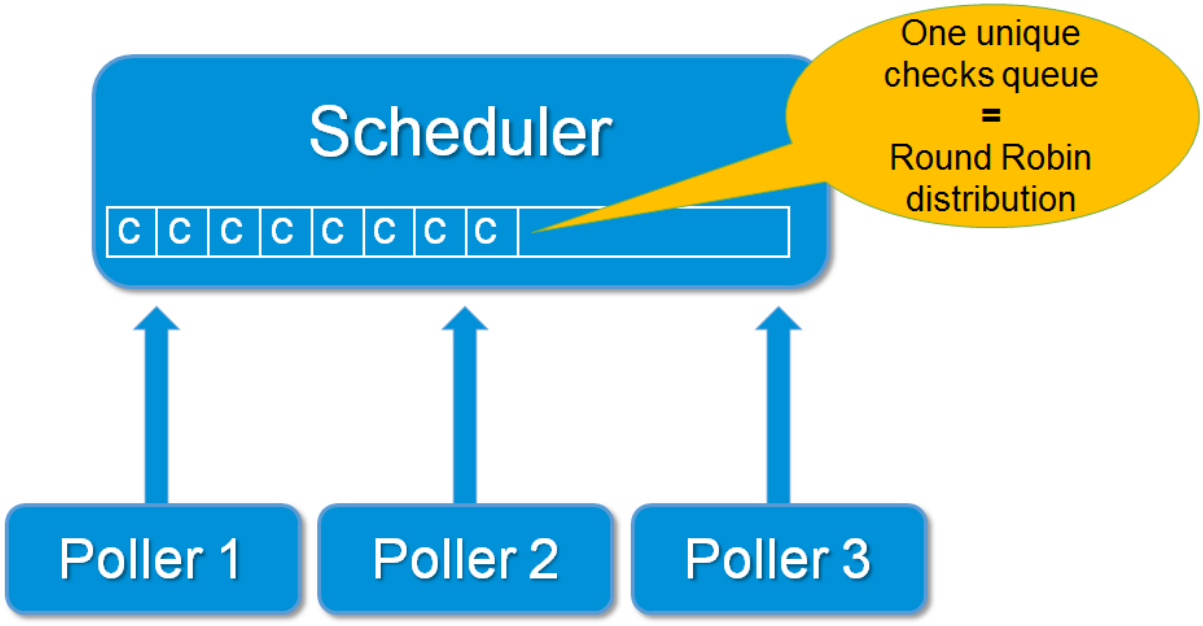
Il y a principalement 2 cas où la charge pose problème :

- utiliser des plugins qui nécessitent beaucoup de processing (check_esx3.pl est un bon exemple)
- planifier un très grand nombre de checks

Dans la 1er cas, vous devez ajouter plus de pollers. Dans le second, vous devez ajouter plus de schedulers.

Pour l'instant, restons sur le 1er cas, typiquement une installation avec moins de 150K checks en 5 minutes, et nous devons seulement calibrer les pollers, pas les schedulers.

Comment plusieurs pollers se connectent à un scheduler commun



Mise en oeuvre d'une architecdture en load balancing avec plusieurs pollers

Commencez par installer le package Shinken Enterprise comme d'ordinaire, mais juste en mode pollernode :

$ ./install.sh --pollernode

Declarez le nouveau poller dans le fichier principal de configuration

Maintenant, vous avez un nouveau pollller déclaré, server2. Mais l'arbiter server1 a besoin de savoir qu'il a des tâches à lui donner. Cela se fait en déclarant le nouveau poller dans le fichier etc/shinken/pollers/poller-master.cfg file.

Editez le fichier /etc/shinken/pollers/poller-master.cfg file et définissez votre nouveau poller sous la définirion du poller-1 existant (sur le server1):

```
#Pollers launch checks
define poller{
  poller_name poller-2
  address server2
  port 7771
}
```

Vérifiez que vous avez bien ces lignes:

```
define scheduler{
  scheduler_name scheduler-1 ; just the name
  address 192.168.0.1 ; ip or dns address of the daemon
  port 7768 ; tcp port of the daemon
}
```

Note: l'adresse doit être 192.168.0.1 ou server1 mais pas localhost!

Une fois fait, redémarrez Shinken Enterprise.

$ /etc/init.d/shinken restart

Vérification de la connexion

Vous pouvez regarder dans le fichier global shinken.log file que le nouveau poller est démarré et qu'il peut joindre le scheduler-1.
Recherchez les lignes :

```
[All] poller satellite order: poller-2 (spare:False), poller-1 (spare:False),
[All] Trying to send configuration to poller poller-2
[All] Dispatch OK of for configuration 0 to poller poller-2
```

Vous pouvez également regarder les logs du poller sur le server2.

Vous aurez alors les lignes :

```
Waiting for initial configuration
[poller-2] Init de connection with scheduler-1 atHTTP://192.168.0.1:7768
[poller-2] Connexion OK with scheduler scheduler-1
I correctly loaded the modules: []
[poller-2] Allocating new fork Worker: 0
```