

TP Monitoring

Jonathan Schaeffer

21 janvier 2016

Table des matières

1	Contexte et prérequis	2
1.1	La machine virtuelles	2
1.2	Le matériel à monitorer	2
1.3	Validation des prérequis	2
2	Installation de Munin	3
2.1	Apperçu du fonctionnement	3
2.2	Installation du serveur	4
2.3	Agent local	4
2.4	Ajout des blades	6
3	Prise en main IPMI	7
4	Intégrer les métriques IPMI dans munin	7
5	Parler SNMP	8
5.1	Avec une prise intelligente	9
5.2	Avec une sonde de température	9
5.3	Optionnel : avec la sonde de température branchée au PDU	9
5.4	Intégration dans le serveur munin	9
	5.4.1 PDU	9
	5.4.2 Sonde externe	10
5.5	Retour au PDU	12
6	Munin "avancé"	12
6.1	Pour le cluster	12
6.2	composition de graphes (optionnel)	12

1 Contexte et prérequis

1.1 La machine virtuelles

La VM pour ce monitoring s'appelle `monitoring`, les comptes configurés sont :

Login	Mot de passe	Commentaire
<code>root</code>	<code>monitoring</code>	Administrateur
<code>tp</code>	<code>monitoring</code>	Utilisateur normal

C'est une Debian 8, avec une interface graphique simple (`XFCE`), une console (`terminator`) et un navigateur web (`iceweasel`).

Nous allons utiliser cette machine virtuelle comme serveur de monitoring. Elle collectera les données du matériel que nous souhaitons monitorer.

1.2 Le matériel à monitorer

Matériel	Interface	Adresse IP
PDU	SNMP	192.168.10.200
Lames IBM 6385-B2X	IPMI et SNMP	192.168.10.[67]
Sonde scangaule	SNMP	192.168.10.17

Les lames IBM ont des comptes accessible par SSH :

Login	Mot de passe	Commentaire
<code>root</code>	<code>admin</code>	Administrateur
<code>tpcluster</code>		

Le PDU est accessible par telnet (login/mot de passe : `apc/apc`)

1.3 Validation des prérequis

1. Vérifiez la connexion SSH aux différents serveurs
2. Ouvrez une session avec l'utilisateur `tp` sur la machine virtuelle
3. Lancez un terminal et testez la connexion `root` avec la commande suivante :

```
$ su -
```

4. Vérifiez la connectivité telnet vers le PDU.

Sur la machine virtuelle, il manque une étape pour que la suite du TP fonctionne correctement.

Suivez les étapes suivantes :

D'abord éditer le fichier `/etc/apt/sources.list`, ajoutez à la fin de la ligne 7 le mot `non-free`. La ligne 7 ressemblera alors à :

```
deb ftp://ftp.fr.debian.org/debian/ jessie main non-free
```

Mettez à jour la base des paquets et installez `snmp-mibs-downloader`

```
# apt-get update
# apt-get install snmp-mibs-downloader
```

Voilà, c'est tout. Merci !

2 Installation de Munin

Afin de profiter d'un historique dans les graphes, nous allons commencer par installer notre système de monitoring.

Il s'agit de Munin. Sur la machine virtuelle, nous allons installer à la fois le serveur qui collecte les données et génère les graphes et l'agent capable de remonter des données au serveur. Ainsi, nous aurons déjà le suivi de la machine locale.

2.1 Aperçu du fonctionnement

Munin fonctionne en mode Agents/Serveur.

Les agents écoutent sur le port 4949 afin de répondre à un certain nombre de requêtes :

- lister les capacités de collecte de l'agent (les "plugins")
- remonter les données d'un plugin
- remonter les configurations du graphe d'un plugin

Les sondes, ou plugins, sont des exécutables qui doivent être capables de

- retourner une métrique dans un format texte précis (et simple) ;
- retourner des instructions pour le dessin des graphes.

Le serveur munin collecte périodiquement (par cron) toutes les métriques de chacun de ses agents, et les stocke dans une base RRD avec les instructions de rendu graphique.

Voici un récapitulatif des fichiers de configuration :

munin.conf Configuration des nœuds ou du serveur

munin-conf.d/ Tous les fichiers de ce dossier seront ajoutés à la configuration générale

plugins Contient les plugins actifs (ce sont des liens symboliques vers les plugins disponibles)

plugin-conf.d/ Contient les éventuelles configurations pour les plugins

Un mode de fonctionnement CGI permet de ne générer les graphes que lors de requêtes passant par le serveur web, lors de la consultation du dashboard.

Munin permet également de remonter des alertes lors de dépassement des valeurs (par NSCA notamment).

2.2 Installation du serveur

Pour installer munin, faire :

```
# apt-get install munin apache2
```

Le paquet apache2 permet d'installer un serveur web pour la consultation du dashboard. Les configurations par défaut amenées par le paquet nous conviendront pour ce TP.

Il faut indiquer à Munin quels hôtes il doit surveiller. Dans le fichier `/etc/munin/munin.conf`, Vous devriez avoir une configuration par défaut qui ressemble à ceci :

```
[localhost.localdomain]
  address 127.0.0.1
  use_node_name yes
```

Vérifiez que les graphes apparaissent. Munin met à jour sa page web par Cron toutes les 5 minutes. Il faut être patient : <http://localhost/munin>

2.3 Agent local

Pour mieux comprendre le fonctionnement de munin, nous allons contacter l'agent munin local à notre VM. Pour cela, on le contacte sur le port 4949 :

```
# nc localhost 4949

munin node at tpmonitoring

list
cpu cpu_rest df df_inode df_inode_rest df_rest ...

fetch cpu
user.value 21758967
nice.value 66796648
system.value 8623020
idle.value 447198917
iowait.value 2819685
irq.value 46
softirq.value 245534
steal.value 640481
guest.value 0
```

Il est possible de lancer localement chaque plugin, dans son environnement, avec la commande **munin-run** :

```
# munin-run memory
slab.value 335646720
swap_cache.value 6328320
page_tables.value 8306688
vmalloc_used.value 19480576
apps.value 538898432
free.value 963391488
buffers.value 8192
cached.value 2305785856
swap.value 39321600
committed.value 1234829312
mapped.value 21848064
active.value 2228047872
inactive.value 531476480
```

On peut obtenir la liste des plugins disponibles ainsi que leur possible activation avec la commande :

```
# munin-node-configure
Plugin                | Used | Extra information
-----              | ---- | -----
cps_                  | no   |
cpu                   | yes  | _rest
```

L'option `--shell` permet de sortir les commandes qui activeront un plugin installé qui a détecté qu'il peut tourner sur le système.

Les plugins sont de petits scripts qui doivent renvoyer une chaîne de caractères au format vu au dessus. Par ailleurs, s'il est appelé avec l'option "config", il doit renvoyer les paramètres nécessaires au rendu du graphe.

Par exemple :

```
# munin-run cpu config
```

2.4 Ajout des blades

Un agent est déjà installé dans les blades. Il répondra à toutes les machines sur son réseau.

Pour le contacter :

```
$ nc cluster1 4949
list
fetch cpu
```

Pour ajouter les machines à notre serveur de monitoring, il faut ajouter dans `/etc/munin/munin-conf.d/cluster.conf` :

```
[cluster1.localdomain]
address _____.____.____.____
use_node_name yes

[cluster2.localdomain]
address _____.____.____.____
use_node_name yes
```

Pendant que se mettent en place les graphes, nous allons manipuler l'IPMI.

3 Prise en main IPMI

Dans cette partie, nous allons interroger les serveurs afin d'obtenir des informations environnementales :

- température en face avant
- consommation électrique
- vitesse de rotation des ventilateurs

La BMC (Basecard Management Controller) des blades est accessible par le protocole IPMI.

Nous allons commencer par y accéder localement afin de valider le fonctionnement :

```
$ ssh cluster1
$ ipmitool -Iopen sensor
$ ipmitool -Iopen chassis power status
```

Ensuite, depuis notre machine virtuelle,

```
$ ipmitool -Ilanplus -H cluster1 -U monitor sensor
```

Le mot de passe de l'utilisateur `monitor` est `monitor`

La commande ci-dessus doit vous permettre d'obtenir des valeurs correspondant aux différents capteurs présents dans la machine.

4 Intégrer les métriques IPMI dans munin

La sonde IPMI pour munin est déjà installée sur les clusters.

Voici les éléments de configurations en place que vous pouvez explorer sur un des membres du cluster :

1. le paquet `munin-node` est installé ;
2. la configuration de munin permet d'interroger le port 4949 depuis tout le monde (il s'agit de l'instruction "allow" dans le fichier `munin-node.conf`) ;

```
allow ^127\.0\.0\.1$
allow ^.*$
allow ^:::1$
```

3. la sonde munin `ipmi_temp` est configurée :

- lien symbolique `/usr/share/munin/plugins/ipmi_ -> /etc/munin/plugins/ipmi_temp`
- configuration particulière dans `/usr/munin/plugin-conf.d/ipmi`

```
[ipmi_*]
user root
timeout 20

[ipmi_sensor_*]
user root
timeout 20

[freeipmi]
user root
timeout 20
```

Sur le cluster, vous pouvez interroger la sonde munin :

```
# munin-run cpu
# munin-run ipmi_temp
```

Normalement, les métriques sont déjà remontées par votre serveur munin.

Depuis la machine virtuelle, contactez un des cluster :

```
# nc cluster1 4949
> list
> fetch ipmi_temp
```

Vous devriez obtenir la liste des plugins disponibles sur cluster1, puis le retour du plugin `ipmi_temp`.

C'est ce que fait périodiquement notre machine virtuelle, afin de constituer les graphes.

5 Parler SNMP

Dans cette partie, nous allons interroger le PDU monitoré afin d'obtenir des informations sur les consommations électriques. Il est aussi possible d'obtenir des informations de température grâce à des sondes spécifiques.

5.1 Avec une prise intelligente

Récupérer les informations spécifiques au matériel :

```
$ snmpwalk -c public -v1 _____.____.____.____ enterprises.318
```

Le paramètre `enterprises.318` est la représentation logique de l’OID correspondant aux informations spécifiques de ce matériel.

5.2 Avec une sonde de température

Le TP se fera sur le modèle de sonde de marque ”ScanGaulle”.

Le modèle de sonde de température pour ce TP répond aux requêtes SNMP. La mesure de la température est dans l’OID :

```
.1.3.6.1.4.1.3854.1.2.2.1.16.1.3.0
```

On peut donc récupérer cette valeur avec la commande :

```
$ snmpget -v1 -c public _____.____.____.____ \
.1.3.6.1.4.1.3854.1.2.2.1.16.1.3.0
SNMPv2-SMI::enterprises.3854.1.2.2.1.16.1.3.0 = INTEGER: 22
```

5.3 Optionnel : avec la sonde de température branchée au PDU

Si le matériel est disponible, nous allons collecter des données d’une sonde externe branchée sur le PDU.

La température se récupère sur cet OID :

```
1.3.6.1.4.1.318.1.1.26.10.2.2.1.8.1
```

L’humidité relative sur cet OID :

```
1.3.6.1.4.1.318.1.1.26.10.2.2.1.10.1
```

5.4 Intégration dans le serveur munin

5.4.1 PDU

Nous allons installer un plugin munin pour grapher les données renvoyées par le PDU. Récupérez sur votre machine virtuelle ce script :

```
https://github.com/munin-monitoring/contrib/blob/master/plugins/snmp/snmp\_\_apc\_pdu
```

Copier ce fichier dans `/usr/share/munin/plugins/`, avec tous nos plugins.

Afin d'activer le plugin, trois étapes sont nécessaires :

1. Déclarer la sonde
2. Activer le plugin
3. Configurer le plugin

La déclaration de la sonde se fait comme si on déclarait un nouveau nœud au serveur munin. Pour ce faire, il suffit d'ajouter un fichier `/etc/munin/munin-conf.d/sondes_snmp.c` qui contient la déclaration de la ou des sondes (évitez les signes `-` dans le nom de la sonde) :

```
[sondeapc.localdomain]
address 127.0.0.1
use_node_name no
```

L'activation du plugin se fait par un lien symbolique. Le nom du lien contient un des paramètres pour le plugin, le nom du nœud (virtuel) à contacter. Ce paramètre est placé entre deux soulignés `"_"`.

Et le lien symbolique sera donc : `/usr/share/munin/plugins/snmp__apc_pdu`
-> `/etc/munin/plugins/snmp_sondeapc_apc_pdu`

La configuration consiste à la mise en place d'un fichier dans :
`/etc/munin/plugin-conf.d/sondeapc` avec le contenu :

```
[snmp*_apc_pdu]
env.community public
env.version 1
```

Il faut redémarrer l'agent munin afin qu'il tienne compte de son nouveau plugin :

```
# systemctl restart munin-node
# munin-run snmp_sondeapc_apc_pdu
```

Si la dernière commande renvoie bien une valeur, alors il nous suffit de patienter pour que le tableau de bord nous dessine les valeurs récupérées.

5.4.2 Sonde externe

Ensuite pour mettre en place le tracé d'une sonde SNMP, nous allons intégrer un script déjà prêt.

Récupérer le script `snmp__scangauletemperature` en suivant ce lien : <https://box.univ-brest.fr/f/12b6fa65c7/?dl=1> et installez-le, sur votre VM, dans le dossier des plugins : `/usr/share/munin/plugins/`

Afin d'activer le plugins, trois étapes sont nécessaires :

1. Déclarer la sonde
2. Activer le plugin
3. Configurer le plugin

La déclaration de la sonde se fait comme si on déclarait un nouveau nœud au serveur munin. Pour ce faire, il suffit d'ajouter un fichier `/etc/munin/munin-conf.d/sondes_snmp.c` qui contient la déclaration de la ou des sondes :

```
[sondeexterne.localdomain]
address 127.0.0.1
use_node_name no
```

L'activation du plugin se fait par un lien symbolique. Le nom du lien contient un des paramètres pour le plugin, le nom du nœud (virtuel) à contacter.

Et le lien symbolique sera donc :

```
/usr/share/munin/plugins/snmp__scangauletemperature ->
/etc/munin/plugins/snmp_sondeexterne_scangauletemperature
```

La configuration consiste à la mise en place d'un fichier dans `/etc/muin/plugin-conf.d/scangaule` avec le contenu :

```
[snmp*_scangauletemperature]
env.community public
env.version 1
```

Il faut redémarrer l'agent munin afin qu'il tienne compte de son nouveau plugin :

```
# systemctl restart munin-node
# munin-run snmp_sondeexterne_scangauletemperature
```

Si la dernière commande renvoie bien une valeur, alors il nous suffit de patienter pour que le tableau de bord nous dessine les valeurs récupérées.

En attendant :

5.5 Retour au PDU

Nous allons chercher à allumer et éteindre les prises de courant du PDU, via le protocole SNMP.

L'OID correspondants aux prises de courant (outlets) est :

1.3.6.1.4.1.318.1.1.12.3.1.1

Il est possible de parcourir les OIDs disponibles, par exemple avec ce site : <http://oid-info.com/cgi-bin/display?tree=1.3.6.1.4.1.318.1.1.12.3.1.1&see=all>

On peut donc allumer ou éteindre une prise en donnant une valeur au bon champs : 1 pour allumé, 2 pour éteint.

Par exemple cette commande coupe l'alimentation de la prise numéro 4 :

```
# snmpset -v 1 -c public 192.168.0.2 \  
1.3.6.1.4.1.318.1.1.12.3.3.1.1.4 i 2
```

La commande `snmpset` prend 3 paramètres : l'OID, le type de valeur (ici, "i" est pour "integer") et la valeur à poser.

6 Munin "avancé"

6.1 Pour le cluster

Nous allons utiliser munin pour grapher les queues des schedulers.

Les plugins Grid Engine et Slurm sont déjà installés sur les cluster. En activant le plugin, il sera automatiquement récupéré par votre serveur munin qui générera les graphes.

Lancez des jobs sur le cluster et regardez le résultat sur le serveur Munin.

6.2 composition de graphes (optionnel)

Nous allons créer un nouveau graphe qui compose plusieurs métriques remontées pas Munin.

Par exemple, toutes les données de température.

Pour cela nous allons créer une nouvelle configuration de munin permettant d'agréger plusieurs métriques. L'idée est de créer un nœud "virtuel" pour lequel il n'y aura aucune collecte de données, seulement des grahes.

Créez un fichier `/etc/munin/munin-conf.d/agregate_temp.conf`

```
[localdomain;temperatures]
  update no
  temperatures.update no
  temperatures.graph_title Temperature agregation
  temperatures.graph_args --base 1000
  temperatures.graph_vlabel Temperature
  temperatures.graph_info Montre la temperature du datacentre
  temperatures.graph_scale no
  temperatures.graph_order \
    scangaule=localdomain;sondetemperature \
    cluster1=localdomain;cluster1;ipmi_temp.SONDE \
    cluster2=localdomain;cluster2;ipmi_temp.SONDE
```

Ici **SONDE** doit être remplacé par le nom de la métrique récupérée par `ipmi_temp`.

Explications :

update no Indique à Munin de ne pas récupérer de données pour ce nœud

temperatures.update no déclare la valeur `temperatures` sans aucune collecte

temperatures.graph* différentes configurations pour l'aspect du graph.

temperatures.graph_order déclare l'ensemble des valeurs à afficher dans ce graphe, avec un nom logique et la métrique à intégrer.

S'il vous reste du temps, mettez en place le système d'agrégation en cumulant les valeurs de consommation électriques remontées par le PDU monitoré. Vous pouvez vous aidez de cette page de documentation :

<http://guide.munin-monitoring.org/en/latest/example/graph/aggregate.html>