

Le monitoring au Cr@ns

Pauline POMMERET

Encadrée par Sylvain BOILARD

26 février 2013

Qu'est ce que le *monitoring* ?

monitoring : surveillance et mesure d'une activité informatique.

Les raisons peuvent être :

- mesure de performance (temps de réponse)
- mesure de disponibilité (charge)
- mesure d'intégrité (état des processus Unix, modifications)
- mesure de changement (édition pages web)

Le *monitoring* au Cr@ns

3 outils de *monitoring* :

- 1 monit *et autostatus*
- 2 munin
- 3 nagios

Plan

- 1 Monit
 - À quoi sert `monit` ?
 - Où est Monit au Cr@ns ?
 - `/etc/monit/monitrc`
 - Configuration du monitoring

- 2 Autostatus
 - Etat de la connexion

Principe

- `Monit` est un logiciel de surveillance des services et de système de fichiers ;
- prévient lors de dysfonctionnements (service injoignable, espace disque disponible trop faible) ;
- relancer des services ;
- peut vérifier la disponibilité d'un daemon et les ressources consommées (selon résultat : laisser tranquille, redémarrer, stopper) ;
- envoi de mail.

Sur quels serveurs ?

Monit est présent sur tous les serveurs sauf `fy` et `fz`. En revanche leurs `domU` le sont.

Monit est configuré grâce à :

- `monitrc` pour la configuration de base
- `bcfg2` pour le reste

daemon, serveur de mail et alertes

`set daemon 60` lance monit en tant que daemon toutes les 60 secondes

`set mailserver localhost, smtp.adm.crans.org` permet l'envoi de mails par localhost par postfix ou par smtp.adm.crans.org (monit supporte authentification smtp)

`set alert roots@crans.org {events}` permet l'envoi d'une alerte globale à roots@crans.org en cas d'apparition d'un event (nonexist, timeout, permission,...)

interface web

`set httpd port 2812` permet l'utilisation de l'interface web sur le port 2812. Jusqu'à la nuit dernière, tous les adhérents pouvaient tuer `ssh` sur `zamok` et tous les apprentis pouvaient tuer tous les services sur tous les serveurs.

format des mails

`set mail-format {<en-têtes>}` les en-têtes sont `from`,
`subject` et `message`. On peut utiliser des
`$HOST`, `$ACTION`, `$EVENT`,...

La configuration par services est gérée par :

```
include /etc/monit/conf.d/*
```

Mais au Cr@ns, on fait différemment

Les fichiers individuels de configurations des services ne sont pas dans `/etc/monit/conf.d/` au Cr@ns.

La configuration de tous les services est fait dans un seul fichier `/etc/monit/services`, ce qui permet la gestion par `bcfg2`.

C'est pourquoi on ajoute à la fin de `monitrc` :

```
include /etc/monit/services
```

/etc/monit/services

```
# -*- mode: conf-unix; coding: utf-8 -*-  
#  
# Fichier gere par BCfg2 (avec le plugin Python)  
#
```

FIGURE: Début du fichier services

C'est assez explicite... (voir séminaire `bcfg2`)

/etc/monit/services/

Exemple de monitoring de service :

```
# apache2
check process apache2 with pidfile /var/run/apache2.pid
  start program = "/etc/init.d/apache2 start"
  stop program = "/etc/init.d/apache2 stop"
  if failed host localhost port 80 protocol http timeout 30 seconds then restart
  if failed host intranet.crans.org port 443 type tcpssl protocol http timeout 30 seconds then restart
  if cpu is greater than 60% for 2 cycles then alert
  if cpu > 80% for 5 cycles then restart
  if totalmem > 500.0 MB for 5 cycles then restart
  if children > 250 then restart
  if loadavg(5min) greater than 10 for 8 cycles then restart
  if 3 restarts within 5 cycles then timeout
```

FIGURE: Exemple de configuration pour apache2

La syntaxe est assez transparente, les choix sont ceux des nounous...

/etc/monit/services/

Exemple de monitoring de système de fichiers :

```
# partition /  
check filesystem fs/ with path /  
  if space usage > 80% then alert  
  if inode usage > 80% then alert  
mode passive
```

FIGURE: Exemple de configuration pour / sur zamok

inode (pour `ext4`) : ensemble de métadonnées (taille, date de dernière modifications, droits, propriétaires, . . .) concernant un fichier ou dossier. Le nombre d'inode est limité.

AutoStatus

Pour « voir » ce qu'il se passe sur les serveurs, plutôt que d'aller sur l'interface web de chaque serveur pour regarder l'état des services, on va sur `CransNounous/AutoStatus`.

La page comporte les informations suivantes :

- état de la connexion
- autostatus des serveurs, géré par la macro «HostStatus»
- autostatus des services, géré par la macro «MonitStatus»

Système de secours