

LE MONITORING AVEC NAGIOS



LAURENT.DESIMONE@epfl.ch, IC-IT
& PASCAL.JERMINI@epfl.ch, DOMAINE IT



INTRODUCTION

LE MONITORING, QU'EST-CE QUE C'EST ?

Le principe de base de la surveillance ou *monitoring* est d'avertir d'un problème le responsable d'une ressource avant même que les utilisateurs ne s'en aperçoivent. Ce responsable pourra ainsi intervenir dans les plus brefs délais sur un ordinateur ou un service. Ceci permet d'être *proactif* pour un service informatique qui gère de nombreux serveurs sur lesquels résident beaucoup de services. Par exemple, une alerte peut être envoyée dans le cas d'un serveur dont la charge dépasse un certain seuil, qu'un disque en miroir est défectueux ou que l'espace libre sur un disque devient insuffisant. La même chose peut être faite pour surveiller un service comme HTTP, NFS ou une base de données.

Un amalgame est souvent fait entre le *monitoring* et le suivi de l'utilisation de ressources. Ceci est probablement dû au fait que certains logiciels commerciaux essaient d'intégrer les deux choses dans le même logiciel. Il est important de bien différencier ces deux aspects, car autant le *monitoring* doit rester simple pour être efficace, autant le suivi de l'utilisation de ressources peut s'avérer beaucoup plus compliqué. Ce suivi est pratique pour le dimensionnement d'un service ou d'un serveur ou pour mettre en évidence un problème de configuration. Par exemple, sur un serveur Web Apache vous désirez suivre le nombre de requêtes par seconde. Si vous les mettez en relation avec la charge du serveur, vous pouvez vous rendre compte à partir de quel moment il devient sous-dimensionné. Cet aspect est aussi important, mais ce n'est pas purement du *monitoring*. Le logiciel **Nagios** [1] ne fait strictement que du *monitoring*, mais il le fait bien ! Pour le suivi de l'utilisation de ressources, un très bon complément, également dans le domaine public, est le duo **Cacti-RRD-Tool**. Mais ce n'est pas l'objet de cet article.

LES OUTILS EXISTANTS

Nagios n'est pas le seul outil existant qui permet de faire du *monitoring*. Il en existe plusieurs autres, comme **Big Sister** [2], qui est un produit *OpenSource*. Dans la catégorie des logiciels payants, on peut citer **Tivoli Netview** (ce dernier est plus orienté pour du *monitoring* réseau) ou alors **HP Open-View Operations**. Microsoft dispose aussi d'une solution plus axée sur la plate-forme Windows, appelée **Microsoft System Center Operations Manager (SCOM)**.

NAGIOS

INTRODUCTION

Le logiciel **Nagios** est le successeur de **NetSaint** dont la première version date de 1999. Ce logiciel apparaît sous le nom de Nagios le 10 mai 2002 aux conditions de la *GNU General Public License*.

Avec la mise à disposition de la première version finale en novembre 2002, nous avons commencé à le mettre en exploitation au service informatique de la faculté I&C. Jusqu'en 2005 ce service a fonctionné sur une vieille station Sun à 167MHz, c'est à dire si ce logiciel est peu gourmand en ressource.

Nagios fonctionne sous pratiquement toutes les déclinaisons de Unix. Mais il permet aussi de surveiller des serveurs sous d'autres systèmes d'exploitation, Windows par exemple. Son architecture décrite plus bas permet de comprendre l'indépendance qu'il y a entre le serveur Nagios et les ressources sous surveillance.

FONCTIONNALITÉS

Malgré sa simplicité, toutes les fonctionnalités dont nous avons besoin sont présentes. On peut mettre en évidence celles-ci:

- *monitoring* de services réseau (SMTP, POP3, HTTP, SSH, etc.);
- *monitoring* des ressources d'un serveur (charge processeur, utilisation disque et mémoire, fichiers logs, etc.)
- *monitoring* environnemental par exemple la température de la salle serveurs;
- architecture par *plugin* afin de pouvoir facilement développer nos propres outils;
- notification de problème via e-mail, SMS ou autre;
- interface Web ou par la ligne de commande;
- définition de contacts et de groupes de contacts à notifier;
- définition d'une hiérarchie de serveurs et/ou de services pour limiter le nombre de notifications.

Pour des raisons évidentes de sécurité, nous ne pouvons pas vous donner accès à l'interface Web de nos serveurs Nagios. Néanmoins, des captures d'écran sont disponibles sur Internet [3].

ARCHITECTURE

L'architecture de Nagios se base sur le paradigme **serveur-agent**. D'une manière générale, un serveur fait office de point central de collecte des informations. Les autres machines du réseau exécutent un agent chargé de renvoyer les informations au serveur.

De manière plus fine, Nagios peut être décomposé en trois parties:

- un ordonnanceur, chargé de contrôler quand et dans quel ordre les contrôles des services sont effectués. Nagios essaie toujours de répartir les contrôles au mieux dans le temps, afin de ne pas surcharger le serveur et les machines à surveiller;
- une interface graphique qui affiche de manière claire et concise l'état des services surveillés. Il est ainsi possible de voir d'un seul coup d'œil quels sont les services ayant besoin d'une intervention de leur administrateur;

- des *plugins* qui exécutent les contrôles proprement dits et qui renvoient les résultats de leurs contrôles au serveur Nagios.

On peut distinguer deux catégories de contrôles de services:

- ceux qui peuvent être exécutés à distance, pour les services ouverts sur le réseau (par exemple un serveur HTTP ou SSH);
- ceux qui nécessitent l'exécution du contrôle sur la machine elle-même, pour des services ou des informations qui ne sont pas accessibles depuis le réseau (par exemple l'espace disque restant ou la charge CPU de la machine).

La première catégorie de contrôles est la plus simple: le serveur Nagios lui-même peut contrôler l'état du service en exécutant une requête ad hoc vers le service concerné au travers du réseau. Par exemple, une simple connexion sur le port TCP 22 permet immédiatement de voir si le service SSH est toujours disponible.

La deuxième catégorie est plus laborieuse. Il faut installer un agent sur la machine à surveiller, qui se fera interroger par le serveur Nagios au travers du réseau. Cet agent est chargé d'exécuter des contrôles localement sur la machine à surveiller, et de renvoyer le résultat à Nagios. L'agent le plus utilisé est NRPE. Il est composé de deux parties:

- l'agent avec les *plugins* de contrôle est installé sur la machine à surveiller;
- le *plugin* qui interroge l'agent est installé sur le serveur.

Une distinction supplémentaire entre les types de contrôles se fait au niveau de la manière dont le *monitoring* est effectué. Il y a d'une part les contrôles actifs, qui sont déclenchés par le serveur Nagios pour s'enquérir de l'état d'un service, et d'autre part les contrôles passifs. Pour ces derniers, c'est le service lui-même qui communique au serveur Nagios ses changements d'état.

LES SYSTÈMES D'EXPLOITATION SUPPORTÉS

Nagios est un outil qui permet de faire du *monitoring* sur d'autres plates-formes qu'Unix. En effet, il est possible de surveiller des machines Windows, notamment grâce au *plugin check_nt*, qui fonctionne selon le même principe que NRPE. L'agent qui est exécuté sur la machine à surveiller s'appelle NSClient++ et peut être couplé à des scripts ou exécutables qui font des contrôles spécifiques sous Windows.

Grâce à ses fondations UNIX, la plate-forme MacOS X est aussi supportée. Les machines MacOS X peuvent aussi héberger un serveur Nagios, en plus d'être surveillées à distance comme tout autre client UNIX.

LES OBJETS ET DIRECTIVES À DÉFINIR

Sans vouloir reprendre ici le contenu de la documentation, nous vous faisons un résumé des objets qu'il est possible de personnaliser:

- **Les contacts et groupes de contacts.** Une personne peut par exemple apparaître plusieurs fois. Une fois avec son adresse e-mail, une fois son numéro de portable pour les SMS, etc. Un groupe peut par exemple être composé uniquement des adresses e-mail pour des problèmes de basse priorité et des numéros de portable pour les problèmes urgents.

- **Les tranches horaires.** Elles seront utilisées pour les services et les notifications. Un administrateur système n'a peut-être pas envie de recevoir des SMS à 4h du matin; à cette heure un e-mail sera moins intrusif dans sa vie privée.

- **Les méthodes de notification.** Suivant le système en place, il est possible d'envoyer des commandes externes pour notifier d'un problème (vers un système de *pager* par exemple). Le système de SMS par e-mail disponible à l'EPFL nous simplifie grandement la tâche.

- **Les services et groupes de services.** Il s'agit bien là des services qui tournent sur les serveurs. Donc par exemple une base de données Oracle, un service Web, un serveur de fichiers, etc.

- **Les serveurs et groupes de serveurs.** Dans ce cas, c'est le serveur et ses composants, par exemple ses disques, sa mémoire, etc.

- **Les commandes.** Ces commandes sont celles qui vont être exécutées pour contrôler l'état d'un service ou d'un serveur. Très simplement, si le résultat d'une commande est **0**, *tout va bien*; si c'est **1** l'état est *attention à surveiller*; et **2** l'état est *critique*.

NAGIOS AU QUOTIDIEN

FINE-TUNING

Lors de la mise en exploitation d'un système Nagios il est inévitable que les seuils de certains *plugins* ne soient pas corrects. En général, on a tendance à paramétrer des seuils trop bas, ce qui déclenche trop régulièrement des alarmes. En étant inondé de messages d'alarmes, le risque le plus grand est de passer à côté de vraies situations critiques. Il convient donc de les ajuster au fur et à mesure afin qu'uniquement lors d'un problème sérieux, une alarme soit levée.

NOTIFICATIONS

Lorsque Nagios détecte un problème, les administrateurs du service concerné sont avertis par un message. La situation sera examinée, et si les administrateurs peuvent régler le problème immédiatement (libérer de l'espace disque par exemple), un message de rétablissement sera envoyé. Mais il peut arriver qu'un problème ne puisse pas être corrigé immédiatement, par exemple si une pièce de remplacement doit être commandée chez le fournisseur. Dans ce cas, si l'alerte n'est pas quittancée, Nagios enverra périodiquement des messages pour ce problème. Cela n'est pas toujours souhaitable, car on court à nouveau le risque de passer à côté d'autres alertes importantes. Afin d'éviter cela, il est possible dans l'interface Web de Nagios de lui dire que l'alerte est quittancée. Le service en question restera dans l'état d'alerte, mais plus aucun message ne sera envoyé, du moins jusqu'à ce que le problème soit corrigé.

PRÉPARER UNE INTERVENTION

Dans le cycle de vie d'un serveur, il arrivera le moment où le service devra être temporairement interrompu, par exemple pour de la maintenance préventive ou des besoins de reconfiguration. Normalement, dès que le service sera coupé pour le début de la maintenance, Nagios devrait remarquer le problème et immédiatement alerter les personnes de contact.

Ceci n'est pas forcément idéal, car les personnes de contact ne seront pas toutes au courant de cette maintenance et pourraient s'alarmer pour rien. Dans ce cas, Nagios permet de mettre un service en mode *maintenance* pour une durée déterminée. De cette manière, aucun message d'alerte ne sera envoyé.

ÉCRIRE SES OUTILS DE CONTRÔLE

LA THÉORIE

Le gros avantage de Nagios réside dans la possibilité d'écrire ses propres commandes de contrôle (*plugins*). Ceci nous permet de surveiller pratiquement n'importe quel équipement ou n'importe quel service. Vous avez une imprimante et vous souhaitez avertir par e-mail la personne chargée de son entretien ? Vous pouvez simplement récupérer dans la documentation les codes SNMP qui indiquent le niveau du toner et du papier. Vous utilisez ensuite la commande `check_snmp` pour récupérer son état avec ses codes.

Mais il est possible d'aller encore plus loin. Vous pouvez écrire votre propre commande en n'importe quel langage. Du Visual Basic sous Windows, un script Perl ou Bash sous Unix. Peu importe, du moment que votre script retourne une des valeurs suivantes:

- 0 *pas de problème*, le service ou le serveur est OK
- 1 *attention*, il y a un problème
- 2 *critique*, il y a un problème grave
- 3 il y a eu une erreur non définie.

La chaîne de caractères en retour sera utilisée pour l'affichage d'un message sur l'interface utilisateur.

Pour définir les seuils de ce qui est *Warning* et *Critical*, il faut les passer comme arguments au moment d'invoquer le script. Habituellement, les arguments sont donnés de la manière suivante:

```
check_users -w 100 -c 200
```

Dans cet exemple, le seuil de 100 utilisateurs connectés sur la machine déclenchera une notification de type *Warning* et à 200, ce sera *Critical*. L'invocation de ce script avec ses arguments nous permet de définir la directive **commande** d'un service et d'agir différemment en fonction de son état.

UN PETIT EXEMPLE PRATIQUE

Certains services ne peuvent avoir que des états binaires et passeront directement de OK à *Critical* ou vice versa. L'état de deux disques en miroir ne peut, par exemple, pas être *presque bon*. Les deux disques sont, ou ne sont pas synchronisés.

Dans l'exemple simple en *Bourne shell* ci-après, nous voulons tester l'état d'un serveur NFS. L'argument donné est le nom du serveur à tester. La commande qui est évaluée est `showmount -e nom_du_serveur`.

```
mountdata=`showmount -e $1 2>&1`
status=$?
if test ${status} -ne 0; then
    echo «CRITICAL - NFS server problem!»
    exit 2
else
    dataline=`showmount -e $1 |wc -l`
    mountnbr=`expr $dataline - 1`
    echo «NFS server OK - $mountnbr exports»
    exit 0
fi
```

Le code de sortie indique à Nagios l'état du service qui vient d'être testé (`exit 2` ou `exit 0`). Le texte défini avant la sortie du script sera utilisé pour l'affichage dans l'interface utilisateur (`echo "CRITICAL - NFS server problem!"` ou `echo "NFS server OK - $mountnbr exports"`).

NAGIOS À L'EPFL

SERVICE INFORMATIQUE DE LA FACULTÉ I&C

En 2002, nous avons commencé à inventorier tous les services susceptibles d'être surveillés par Nagios. Environ 90% des serveurs ou services le sont et leur état actuel est visible à cette adresse: ic-it.epfl.ch/status.

Sur cette page, vous pouvez également avoir une vue par salle, par armoire ou par serveur. Au moment d'écrire ces lignes, pas moins de 46 serveurs (physiques ou virtuels) et 178 services sont surveillés en permanence. Les 10% restants restent encore à faire ou ont leur propre système de notification, c'est le cas par exemple du NetApp de la faculté.

A noter les senseurs environnementaux [4] qui surveillent la température et l'humidité des 5 salles sous notre responsabilité. Ils apparaissent deux fois:

- comme services surveillés avec des seuils qui déclencheront une notification (vue de *Tous les services*);
- comme suivi de ressource avec graphe de l'évolution dans le temps de la température et de l'humidité (vue de *Sensors*).

Ceci illustre bien la différence qu'il faut faire entre le *monitoring* et le suivi de l'utilisation de ressources.

Pour surveiller le serveur de *monitoring* lui-même, un deuxième serveur Nagios est installé sur une autre machine. Mais il est extrêmement simple, il ne fait que surveiller le serveur principal. A noter que le logiciel Nagios est tellement peu gourmand en ressource qu'il s'accommode très bien de cohabiter avec d'autres services sur un seul et même serveur.

BLUEGENE/L ET GREEDY AU DIT-EX

Les serveurs utilisés par les services BlueGene/L et Greedy du DIT sont tous surveillés par un serveur Nagios chacun. Le service de *monitoring* n'est pas consolidé entre les deux pour des raisons techniques, notamment l'utilisation de réseaux privés non-routables dans le cas de BlueGene/L.

Pour ces deux services, il s'agit essentiellement de surveiller l'état de santé des disques en RAID, ainsi que les agrégats réseau. Dans le cas de Greedy, la flexibilité apportée par la possibilité d'écrire des *plugins* fut d'une grande aide, car à notre connaissance aucun module de contrôle spécifique aux logiciels utilisés sur ce serveur n'existait. Nous avons donc pu créer nos propres outils afin de surveiller uniquement les parties qui nous intéressaient.

COMMUNAUTÉ NAGIOS

Le projet Nagios existe depuis plusieurs années, ce qui a permis le développement d'une grande communauté d'utilisateurs. Parmi les plus influentes, on peut mentionner le site Web Nagios Exchange [5], qui répertorie un grand nombre de *plugins* qui ne sont pas dans la distribution de base, ou alors

le site un peu plus généraliste Nagios Community [6], qui est un point de rencontre virtuel pour les utilisateurs de Nagios. Il existe en outre plusieurs sociétés qui vendent leurs services de consultants pour la mise en place et la configuration de serveurs Nagios.

Témoins du succès de Nagios, certains boîtiers de contrôle des valeurs environnementales ont vu le jour, par exemple le *Websensor* EM01B [7]. Ils sont certifiés pour s'intégrer facilement dans une infrastructure Nagios, avec *plugin* de surveillance spécifique à l'appui. Il est en effet intéressant d'intégrer les informations de température et d'humidité des salles machine dans Nagios.

Outre les *plugins*, il existe d'autres outils et produits basés sur Nagios, tous issus de la communauté. Parmi les extensions logicielles, on peut citer Nagios Web Config. Cette extension permet de configurer un serveur Nagios directement depuis une interface Web: il n'est donc plus nécessaire d'éditer les fichiers de configuration.

Une approche intéressante a été prise par quelques sociétés qui vendent des *boîtiers* de *monitoring* basés sur Nagios [8]. Ces boîtes à brancher sur le réseau contiennent un système Nagios embarqué: il ne reste plus que la configuration à faire (en général au travers d'une interface Web simplifiée) pour avoir un système de *monitoring* complet et fonctionnel très rapidement.

CONCLUSIONS

La mise en place physique et logicielle d'un serveur et de ses services associés n'est que la première étape de son cycle de vie, mais c'est aussi la plus courte. Lors de sa mise en exploitation, il est indispensable de se préoccuper de son état de santé et ceci de manière continue, afin de garantir un service avec le moins d'interruptions possible. S'il est relativement aisé de contrôler manuellement l'état de santé de quelques serveurs, la tâche devient plus ardue au fur et à mesure que l'on augmente le nombre de machines à surveiller. Le *monitoring* doit aussi être utilisé comme outil préventif

qui alerte l'administrateur avant que la situation ne soit irréparable sans un arrêt de service prolongé.

Le choix du logiciel de surveillance n'est pas le point de départ. Il faut commencer par bien réfléchir à ses propres besoins et attentes. Il faut également faire une bonne analyse de son architecture réseau, de serveurs et de services. Cette phase est importante, car comme toujours, un logiciel ne fera que ce qu'on lui demande de faire. Dans le cas du *monitoring*, c'est d'autant plus important que s'il ne fonctionne pas bien, il y aura très vite une perte de confiance dans la solution. Ce sera d'autant plus marqué si le nombre de personnes notifiées est important.

Le dicton *trop d'info tue l'info* se vérifie aussi dans le *monitoring* avec la quantité de notifications pour un seul et même problème. La seule manière d'éviter cela est de bien définir les dépendances entre les services et les serveurs.

Nagios est un bon outil, simple et fonctionnel, complètement dédié à la tâche du *monitoring*. Sa grande communauté d'utilisateurs et de développeurs est un gage de pérennité, qui montre aussi que le sujet du *monitoring* est d'actualité, en particulier à une époque où la disponibilité est un facteur critique. Nos *indics* nous ont même signalé l'avoir déployé dans certaines banques privées ayant pignon sur rue, mais secret bancaire oblige, nous ne citerons aucun nom...

RÉFÉRENCES

- [1] www.nagios.org
- [2] www.bigsister.ch
- [3] www.nagios.org/about/screenshots.php
- [4] www.nagios.org/products/environmental/esensors/em01b.php
- [5] www.nagiosexchange.com
- [6] www.nagioscommunity.org
- [7] www.nagios.org/products/environmental/esensors/em01b.php
- [8] www.op5.com/op5/products/monitor/nagios ■

