Protocoles et outils de base liés à la supervision

Stéphane Aicardi

CNRS / Centre de Mathématiques Laurent Schwartz

ANGD *Mathrice*, Novembre 2009

Plan

Que superviser ?

Des serveurs :

- hard : températures, alimentation électrique, ventilateurs, carte RAID, ...
- système : charge CPU, utilisation mémoire, occupation des disques, intégrité du système, logs, ...
- réseau : nombre de paquets, d'erreurs ou de collisions, ...
- pour chaque service : port accessible, réponse correcte au protocole, temps de réaction, ...

Que superviser ?

Des équipements réseaux :

- état de chaque interface : cable, vitesse, VLAN, . . .
- nombre de paquets, d'erreurs ou de collisions

Des périphériques :

- Imprimantes : nombre de pages imprimées, état des consommables
- Onduleurs
- Sondes températures
- . . .

Comment superviser ?

De nombreuses techniques sont possibles de la plus bas niveau à la plus intégrée et professionnelle.

Il faut un système de supervision adapté aux besoins :

- qui donne une vue synthétique du système d'information ;
- qui signale tous les problèmes ;
- mais qui ne submerge pas l'administrateur d'informations désorganisées et dispersées.

Architecture d'un système de supervision

Un système de supervision efficace sera basé sur une console centrale qui aura les fonctions suivantes :

- interroger un agent installé sur chaque équipement ($\Longrightarrow$ vue de l'intérieur),
- interroger chaque service ou équipement directement ($\Longrightarrow$ vue de l'extérieur),
- recevoir les alertes émises par les équipements,
- notifier les administrateurs,
- agir pour remettre en service ou protéger un équipement,
- archiver les données récoltées,
- produire des rapports statistiques ou graphiques,
- permettre de visualiser l'état du système d'information

Architecture d'un système de supervision

Un système de supervision efficace sera basé sur une console centrale qui aura les fonctions suivantes :

- interroger un agent installé sur chaque équipement ($\Longrightarrow$ vue de l'intérieur),
- interroger chaque service ou équipement directement ($\Longrightarrow$ vue de l'extérieur),
- recevoir les alertes émises par les équipements,
- notifier les administrateurs,
- agir pour remettre en service ou protéger un équipement,
- archiver les données récoltées,
- produire des rapports statistiques ou graphiques,
- permettre de visualiser l'état du système d'information

Architecture d'un système de supervision

Un système de supervision efficace sera basé sur une console centrale qui aura les fonctions suivantes :

- interroger un agent installé sur chaque équipement ($\Longrightarrow$ vue de l'intérieur),
- interroger chaque service ou équipement directement ($\Longrightarrow$ vue de l'extérieur),
- recevoir les alertes émises par les équipements,
- notifier les administrateurs,
- agir pour remettre en service ou protéger un équipement,
- archiver les données récoltées,
- produire des rapports statistiques ou graphiques,
- permettre de visualiser l'état du système d'information

Briques de base d'une système de supervision

Supervision bas niveau

Il est possible d'accéder à de nombreuses informations localement sur une machine en utilisant des outils "bas niveau". En quelques lignes de shell script, on peut construire un rapport d'état de la machine. Une entrée dans la crontab et la supervision locale peut être assurée.

Supervision bas niveau

Il est possible d'accéder à de nombreuses informations localement sur une machine en utilisant des outils "bas niveau". En quelques lignes de shell script, on peut construire un rapport d'état de la machine. Une entrée dans la crontab et la supervision locale peut être assurée.

Cette méthode est à éviter autant que possible :

- on réinvente la roue ;
- difficulté de maintenance de scripts maison à long terme ;
- de nombreux outils "clés en main" sont disponibles !

On va mentionner cependant quelques outils "bas niveau" qui peuvent être utiles pour effectuer des tests rapides, mais aussi pour configurer des outils plus conséquents.

Supervision bas niveau : état d'un serveur

Tester la charge :

```
# uptime
 17:54:35  up 17 days,  3:19, 50 users,  load average: 0.21, 0.78, 1.05
```

Tester la mémoire :

```
# free
             total       used       free     shared    buffers     cached
Mem:       2054912    1994012      60900          0      75452     705720
-/+ buffers/cache:    1212840     842072
Swap:      8193064     789148    7403916
```

Tester le disque :

```
# df -m /usr/local
Filesystem           1M-blocks      Used Available Use% Mounted on
/dev/sda5                11814     10265      1538  87% /usr/local
```

Supervision bas niveau : ethtool

Tester l'état d'une connexion ethernet :

```
# ethtool eth0
Settings for eth0:
  Supported ports: [ TP ]
  Supported link modes:   10baseT/Half 10baseT/Full
                          100baseT/Half 100baseT/Full
                          1000baseT/Half 1000baseT/Full
  Supports auto-negotiation: Yes
  Advertised link modes:  10baseT/Half 10baseT/Full
                          100baseT/Half 100baseT/Full
                          1000baseT/Half 1000baseT/Full
  Advertised auto-negotiation: Yes
  Speed: 100Mb/s
  Duplex: Full
  Port: Twisted Pair
  PHYAD: 1
  Transceiver: internal
  Auto-negotiation: on
  Supports Wake-on: g
  Wake-on: g
  Current message level: 0x000000ff (255)
  Link detected: yes
```

Superviser "à la main" : netstat

Récupérer les statistiques d'une connexion ethernet :

```
# netstat -ieth0
Kernel Interface table
Iface       MTU Met    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0       1500   0 901984416      0      0      0 1056566005      0      0      0 ABMRU
# netstat -s
Ip:
    1034331822 total packets received
    75 with invalid headers
    0 forwarded
    6 incoming packets discarded
    1032706569 incoming packets delivered
    1268702374 requests sent out
    3276 reassemblies required
    1638 packets reassembled ok
Icmp:
    128809 ICMP messages received
    11538 input ICMP message failed.
    ICMP input histogram:
        destination unreachable: 116137
        timeout in transit: 14
        echo requests: 11420
        echo replies: 1238
[...]
```

Supervision bas niveau : ping, nmap

Tester qu'une machine est connectée au réseau :

```
#  ping -c1 daphne.math.polytechnique.fr
PING daphne.math.polytechnique.fr (129.104.3.2) 56(84) bytes of data.
64 bytes from daphne.math.polytechnique.fr (129.104.3.2): icmp_seq=0 ttl=64 time=0.077 ms

--- daphne.math.polytechnique.fr ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.077/0.077/0.077/0.000 ms, pipe 2
# nmap -sP daphne.math.polytechnique.fr

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Host daphne.math.polytechnique.fr (129.104.3.2) appears to be up.
Nmap run completed -- 1 IP address (1 host up) scanned in 0 seconds
```

Tester qu'un port est ouvert :

```
# nmap -p 22 daphne.math.polytechnique.fr

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on daphne.math.polytechnique.fr (129.104.3.2):
Port       State       Service
22/tcp     open        ssh

Nmap run completed -- 1 IP address (1 host up) scanned in 0 seconds
```

Supervision bas niveau : test d'un service réseau

Tester qu'un service répond :

```
# telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.localdomain (127.0.0.1).
Escape character is '^]'.
220 daphne.math.polytechnique.fr ESMTP Postfix
QUIT
221 Bye
Connection closed by foreign host.
```

Automatiser ce test avec `expect` :

```
#!/usr/bin/expect
set timeout 1000
spawn telnet localhost 25
expect {
        timeout { puts stderr "TIMEOUT\n"; exit 1 }
        "220 daphne.math.polytechnique.fr ESMTP Postfix"
}
send "QUIT\n"
expect {
        timeout { puts stderr "TIMEOUT\n"; exit 1 }
        "221 Bye"
}
exit
```

Supervision bas niveau : iptables

Connaître le nombre de paquets concernés par les règles d'`iptables` :

```
# iptables -nvL OUTPUT
Chain OUTPUT (policy DROP 236K packets, 14M bytes)
 pkts bytes target     prot opt in     out     source               destination
 123M  107G ACCEPT     0    --  *      *       0.0.0.0/0            0.0.0.0/0
       state RELATED,ESTABLISHED
 9595  847K ACCEPT     0    --  *      lo      0.0.0.0/0            0.0.0.0/0
2181K  131M ACCEPT     tcp  --  *      eth0    0.0.0.0/0            0.0.0.0/0
      multiport dports 80,443,21,22 state NEW
[...]
```

Pour garder traces de certains paquets pour les compter ensuite :

```
# iptables ... -j LOG --log-prefix ``pourquoi on jette''
```

iperf

iperf sert à mesurer la bande passante réseau entre deux machines.

Pour cela, on lance iperf en mode serveur sur l'une des machines

```
# iperf -s
```

et on effectue la mesure avec iperf en mode client sur l'autre :

```
# iperf -c 129.104.3.230
------------------------------------------------------------
Client connecting to 129.104.3.230, TCP port 5001
TCP window size: 16.0 KByte (default)
------------------------------------------------------------
[  3] local 129.104.3.225 port 36510 connected with 129.104.3.230 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec    113 MBytes  94.8 Mbits/sec
```

Contrôle d'intégrité

On peut vérifier l'intégrité d'un fichier en combinant les sorties des commandes `ls -li`, `md5sum` et `sha1sum` :

```
# ls -li /usr/sbin/cron
1192362 -rwxr-xr-x 1 root root 41192 2009-09-16 00:32 /usr/sbin/cron
# md5sum /usr/sbin/cron
f060f692ed7ce413e85295d785ffa904  /usr/sbin/cron
# sha1sum /usr/sbin/cron
6aabd34f4c4426b1c84932c06fa2562304b9b20b  /usr/sbin/cron
```

Plusieurs outils permettent d'automatiser le contrôle d'intégrité : Tripwire et aide qui fonctionnent serveur par serveur et osiris qui fournit une protection centralisée.

Un certain nombre des logiciels de supervision cités dans la suite permettent de faire du contrôle d'intégrité.

Intelligent Platform Management Interface

IPMI est une spécification commune à la plupart des constructeurs, consistant en un ensemble d'interfaces permettant de superviser une machine, indépendamment de son système d'exploitation, y compris si elle est éteinte mais connectée à une prise électrique.

Les informations peuvent être accessibles via une interface système directe, un lien série ou une connexion ethernet.

La norme IPMI est propriétaire. On peut trouver des informations détaillées sur le site d'Intel.

IPMI : Architecture et Vocabulaire

Le cœur d'IPMI est un contrôleur appelé *BMC* (Baseboard Management Controller). Il surveille les différents capteurs intégrés à la carte mère (température, vitesse de rotation des ventilateurs, état du système d'exploitation...) et permet certaines actions sur la machine comme l'extinction ou le démarrage.

Les valeurs des capteurs sont appelées *SDR* (Sensor Data Record).

Les divers éléments de la machine appelés *FRU* (Field Remplaceable Unit).

IPMI sous Linux

OpenIPMI fournit les pilotes de périphériques IPMI pour Linux.

ipmitool permet d'interagir avec un serveur équipé d'IPMI localement ou à distance.

ipmitool en action

Localement :

```
# ipmitool sdr
Fan 1 Tach       | 3150 RPM          | ok
Fan 2 Tach       | 2850 RPM          | ok
Fan 3 Tach       | 2850 RPM          | ok
Ambient Temp     | 19 degrees C      | ok
CPU 1 Temp       | 17 degrees C      | ok
CPU 2 Temp       | 20 degrees C      | ok
[...]

# ipmitool fru
FRU Device Description : Builtin FRU Device (ID 0)
 Chassis Type    : Pizza Box
 Chassis Part Number   : 7941Z0A
 Chassis Serial    : 99B0059
 Board Mfg Date        : Mon Apr 28 18:49:00 2008
 Board Mfg             : FOXC
 Board Product         : x3 3250 Planar
 Board Serial          : 87Y053K1090
 Board Part Number     : 46M2288
[...]
```

ipmitool en action

À distance :

```
# ipmitool -I lan -H 192.168.200.6 -U USER -P PASS sdr
Fan 1 Tach       | 3150 RPM          | ok
Fan 2 Tach       | 2850 RPM          | ok
Fan 3 Tach       | 2850 RPM          | ok
Ambient Temp     | 19 degrees C      | ok
CPU 1 Temp       | 17 degrees C      | ok
CPU 2 Temp       | 20 degrees C      | ok
[...]

# ipmitool -I lan -H 192.168.200.6 -U USER -P PASS fru
FRU Device Description : Builtin FRU Device (ID 0)
 Chassis Type    : Pizza Box
 Chassis Part Number   : 7941Z0A
 Chassis Serial    : 99B0059
 Board Mfg Date        : Mon Apr 28 18:49:00 2008
 Board Mfg             : FOXC
 Board Product         : x3 3250 Planar
 Board Serial          : 87Y053K1090
 Board Part Number     : 46M2288
[...]
```

ipmitool en action

Redémarrer une machine à distance :

```
# ipmitool -I lan -H 192.168.200.6 -U USER -P PASS power cycle
```

Se connecter à une console à distance :

```
# ipmitool -I lanplus -H 192.168.200.6 -U USER -P PASS sol activate
```

NB : ipmitool utilise les mêmes séquences d'échappement que SSH.

Impossible de sortir d'une console sans tuer la session SSH sous-jacente. . .

Simple Network Management Protocol

SNMP est un protocole principalement utilisé pour superviser des équipements réseaux (routeurs, switchs. . .), des serveurs ou même des périphériques tels que baies de disques, sondes météorologique, onduleurs...

SNMP : principe de fonctionnement

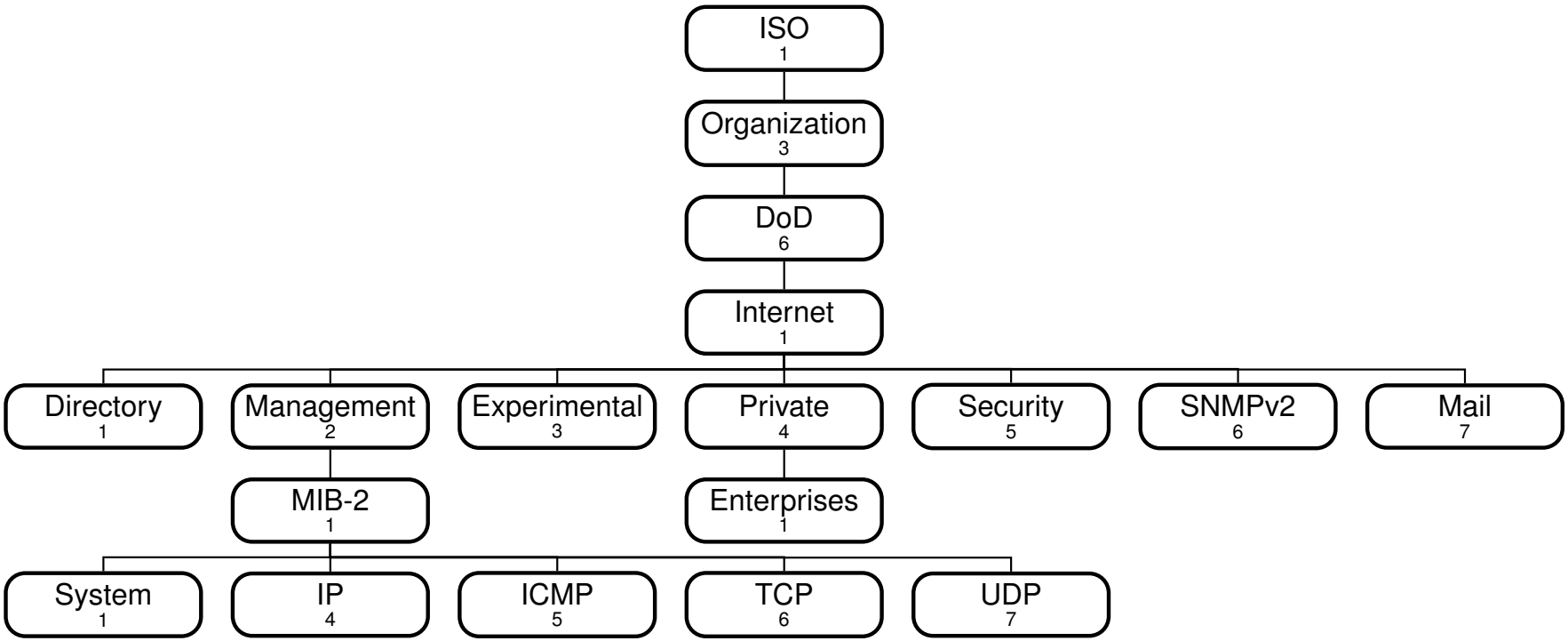
SNMP est basé sur trois éléments :

- un *équipement* à superviser qui contient des *objets* à gérer : informations de configuration, sur le matériel, statistiques. . .
- il exécute un *agent*, c'est-à-dire un logiciel qui agrège les données locales,
- une *console* de supervision qui permet d'interroger les agents accessibles par le réseau ou de recevoir des alertes émises par les agents.

L'interrogation d'un agent se fait en lui envoyant des messages sur le port UDP 161. L'agent envoie des alertes à la console sur le port UDP 162.

SNMP : organisation des données

Les données sont organisées de manière hiérarchique dans une MIB (Management Information Base).



Chaque objet est représenté par un OID (Object IDentifier).
Ex : 1.3.6.1.2.1.1 $\Longleftrightarrow$ iso.org.dod.internet.mgmt.mib-2.system.

SNMPv1 : le protocole

La console interroge l'agent par un datagramme UDP sur le port 161 contenant :

- la version du protocole (0 pour SNMPv1) ;
- la *communauté*, c'est-à-dire une chaîne de caractère déterminant les droits d'accès ;
- une requête parmi : `get-request`, `get-next-request`, `set-request` ;
- des OIDs et éventuellement des valeurs.

L'agent répond par un datagramme contenant la requête `get-response`, avec pour chaque OID requise, la valeur demandée et un code d'erreur.

Il peut envoyer des alertes à la console par un datagramme UDP sur le port 162 contenant la requête `trap`.

SNMP : Commandes en ligne

net-snmp fournit un agent SNMP et des commandes en lignes pour consulter ou administrer des agents SNMP.

`snmpget` permet d'obtenir une ou plusieurs données :

```
# snmpget -v2c -c public localhost hrSystemNumUsers.0 laLoadInt.1 system.sysContact.0
HOST-RESOURCES-MIB::hrSystemNumUsers.0 = Gauge32: 5
UCD-SNMP-MIB::laLoadInt.1 = INTEGER: 0
SNMPv2-MIB::sysContact.0 = STRING: root@localhost
```

`snmpset` permet de définir une ou plusieurs données :

```
# snmpset -v1 -c private localhost system.sysContact.0 s informatique@math.polytechnique.fr
SNMPv2-MIB::sysContact.0 = STRING: informatique@math.polytechnique.fr

# snmpget -v1 -c public localhost system.sysContact.0
SNMPv2-MIB::sysContact.0 = STRING: informatique@math.polytechnique.fr
```

SNMP : Commandes en ligne

`snmpwalk` permet de parcourir les données disponibles :

```
# snmpwalk -v1 -c public localhost memory
UCD-SNMP-MIB::memIndex.0 = INTEGER: 0
UCD-SNMP-MIB::memErrorName.0 = STRING: swap
UCD-SNMP-MIB::memTotalSwap.0 = INTEGER: 2000052 kB
UCD-SNMP-MIB::memAvailSwap.0 = INTEGER: 2000052 kB
UCD-SNMP-MIB::memTotalReal.0 = INTEGER: 2057844 kB
UCD-SNMP-MIB::memAvailReal.0 = INTEGER: 81324 kB
UCD-SNMP-MIB::memTotalFree.0 = INTEGER: 2081376 kB
UCD-SNMP-MIB::memMinimumSwap.0 = INTEGER: 16000 kB
UCD-SNMP-MIB::memShared.0 = INTEGER: 0 kB
UCD-SNMP-MIB::memBuffer.0 = INTEGER: 128728 kB
UCD-SNMP-MIB::memCached.0 = INTEGER: 951812 kB
```

Syslog

Syslog est un protocole de transmission d'événements systèmes. On peut en trouver une spécification exhaustive dans la RFC 3164, améliorée par la RFC 5424.

Il permet de centraliser les événements systèmes de chaque serveur ou équipement réseau sur une seule machine pour des fins d'analyse statistique, d'archivage ou production d'alertes.

Attention ! Pour centraliser efficacement des journaux systèmes, il faut une source de temps (NTP) commune.

Syslog : protocole

Le protocole syslog utilise des datagrammes UDP à destination du port 514.

Chaque événement système est accompagné :

- son type de service (*facility*),
- sa gravité (*severity*),
- la date et l'heure (*timestamp*),
- la machine sur laquelle l'événement s'est produit (*host*).

Serveurs syslog

BSD syslogd est l'implémentation histoire du protocole.

Deux projets concurrents apportent des améliorations substancielles au protocole initial :

- syslog-ng (support de TCP, TLS, IPv6, filtrage par contenu, stockage dans une base de données,...)
- rsyslog (idem plus support de RELP et BEEP, SNMP traps,...)

Pour une comparaison partiale entre les deux projets, on pourra se référer ici.

Analyseurs de journaux systèmes

Les journaux systèmes sont extrêmement nombreux et verbeux. Outre le filtrage introduit dans syslog-ng et rsyslog, il est nécessaire d'utiliser une application d'analyse.

logwatch est un analyseur écrit en Perl. Il est facilement configurable, permet de produire des rapports par service, par jour, par fichier de log et d'envoyer le rapport à une adresse donnée.

swatch est une application qui suit en direct un journal système, y recherche des expressions régulières et lance les actions appropriées.

logcheck.

Netflow/IPFIX

C'est un protocole de suivi de flux réseau développé initialement par Cisco, puis standardisé par la RFC 5101.

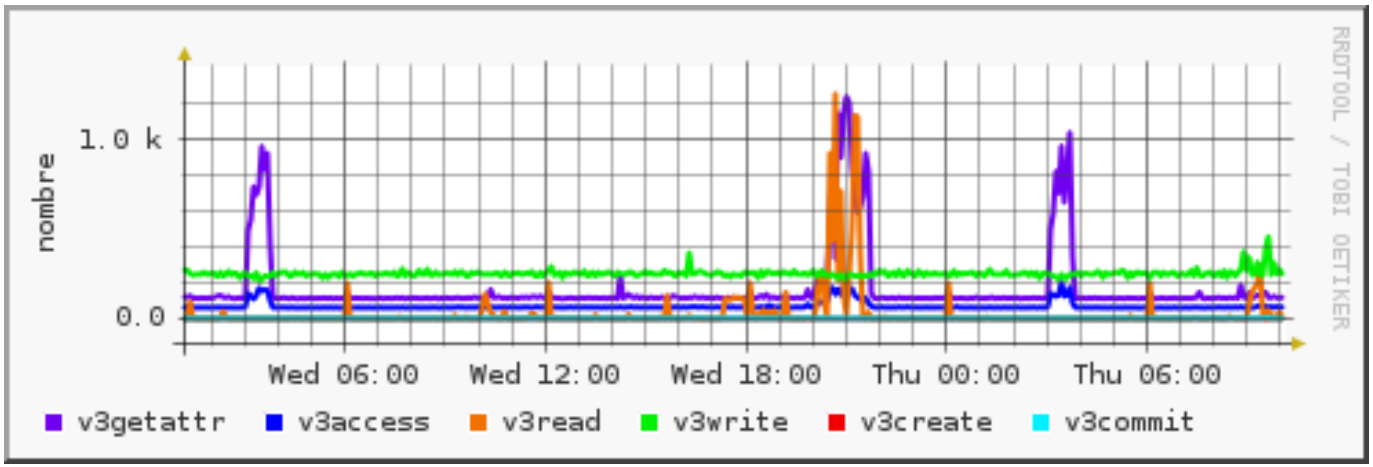
Un *point d'observation* (en général un routeur) collecte les informations sur les paquets qui transitent (source, destination, protocole, ports), agrège ces informations et envoie à intervalles réguliers ces données agrégées à un *collecteur*.

Le protocole de transport utilisé peut-être SCTP, TCP ou UDP.

RRDtool

RRDtool est une suite d'utilitaires de création de graphes basés sur des statistiques temporelles.

RRDtool est devenu incontournable pour tracer des graphiques, tous les outils graphiques cités dans la suite l'utilisent.



Logiciels de suivi graphique

Logiciels de suivi graphique

Ces logiciels fonctionnent tous sur le même principe :

- collecter régulièrement les données de supervision sur les différents équipements en appelant un agent pré-installé (SNMP ou spécifique),
- stocker ces données dans une base de données,
- produire des représentations graphiques temporelles,
- rassembler l'ensemble de ces informations sur un site web.

MRTG

MRTG (Multi Router Traffic Grapher) est un système de collecte de données et de production de graphiques ; il a été écrit par l'auteur de RRDtool.

Initialement écrit pour représenter le trafic réseau d'un switch, MRTG peut être adapté à une multitude de situations : il peut produire des graphes de n'importe quelle donnée accessible via SNMP ou via un script exécuté sur la console.

MRTG : Fonctionnement

MRTG est fait pour représenter graphiquement des groupes de deux données (typiquement le trafic entrant et sortant d'une interface d'un switch). Les données sont récupérées toutes les cinq minutes par une crontab et stockées dans un fichier de taille fixe. Une page web contenant les graphes est créée à chaque itération.

Pour plus de flexibilité et de rapidité, il est possible de configurer MRTG pour utiliser une base RRD au lieu du format natif afin de créer les graphes à la demande.

MRTG : Configuration

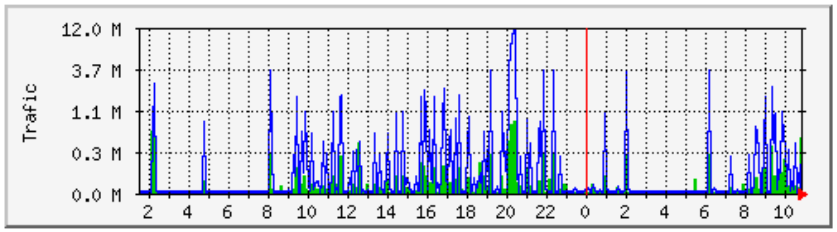
```
WorkDir: /web/cmls/monitoring/hermes
LoadMIBs: /usr/local/share/snmp/mibs/UCD-SNMP-MIB.txt

Title[em1]:  Trafic
PageTop[em1]: <H1>Trafic reseau sur hermes (reseau 129.104.10.x)</H1>
        <P>

MaxBytes[em1]: 12500000
Options[em1]: growright, nopercent, logscale
Target[em1]:  4:public@localhost
YLegend[em1]: Trafic
ShortLegend[em1]: b
Legend1[em1]: Trafic entrant en b/s
Legend2[em1]: Trafic sortant en b/s
Legend3[em1]: Maximal 5 Minute
Legend4[em1]: Maximal 5 Minute
LegendI[em1]:  In:
LegendO[em1]:  Out:
WithPeak[em1]: ymwd
```

MRTG : Exemple

Trafic réseau sur hermes (réseau 129.104.10.x)

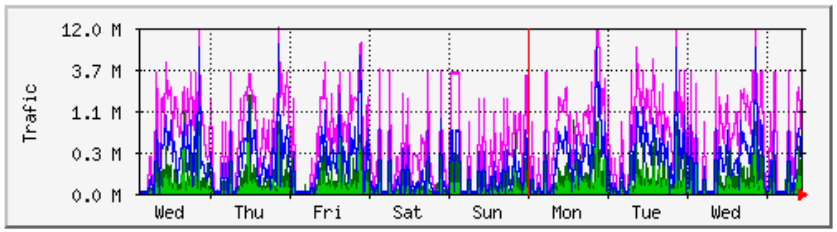
The statistics were last updated **Thursday, 12 November 2009 at 10:50**,
at which time **'hermes.math.polytechnique.fr'** had been up for **16 days, 9:09:44**.

`Daily' Graph (5 Minute Average)



	Max	Average	Current
In:	759.3 kb	29.3 kb	447.8 kb
Out:	11.1 Mb	277.5 kb	369.0 kb

`Weekly' Graph (30 Minute Average)



	Max	Average	Current
In:	1778.2 kb	32.0 kb	23.1 kb
Out:	11.5 Mb	297.5 kb	118.0 kb

MRTG : Bilan

Avantages :

- mise en œuvre très simple, graphiques un peu configurables,

Défauts :

- pas plus de deux données sur un même graphe,
- pas de centralisation des données et graphes,
- pas de système de notification.

Munin

Munin utilise un agent spécifique (*munin-node*) à installer sur chaque machine à superviser qui collecte les informations locales et écoute sur le port TCP 4949. Toutes les cinq minutes (par une crontab) la console contacte les agents munin, stocke les informations collectées dans des bases RRD et produit des pages web et des graphiques. Il est possible également de récupérer et de tracer des données obtenues par des requêtes SNMP.

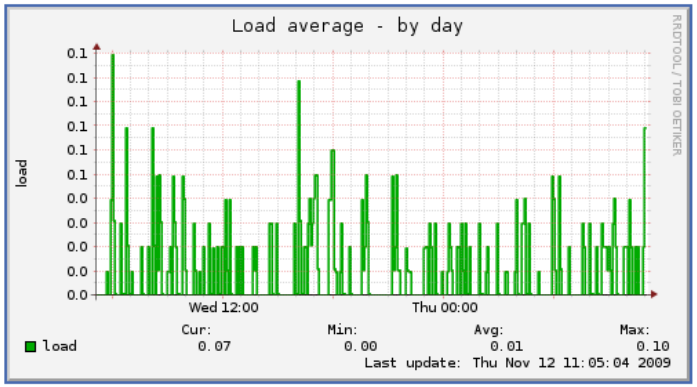
Munin fournit également un tableau de bord synthétique. En cas de dépassement des valeurs d'alertes, Munin peut envoyer un message à Nagios.

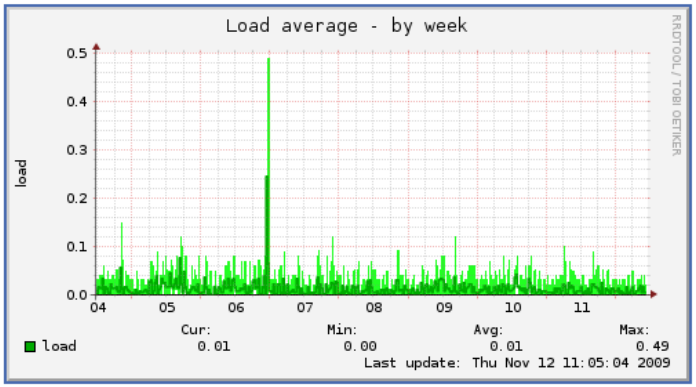
Munin : Configuration

```
# The next three variables specifies where the location of the RRD
# databases, the HTML output, and the logs, severally.
dbdir   /var/lib/munin
htmldir /var/www/munin
logdir  /var/log/munin
rundir  /var/run/munin

# Where to look for the HTML templates
tmpldir /etc/munin/templates

# Drop somejuser@fnord.comm and anotheruser@blibb.comm an email everytime
# something changes (OK -> WARNING, CRITICAL -> OK, etc)
#contact.someuser.command mail -s "Munin notification" somejuser@fnord.comm
#
# For those with Nagios, the following might come in handy.
#contact.nagios.command /usr/sbin/send_nsca -H nagios.host.com -c /etc/send_nsca.cfg

[chronos.math.polytechnique.fr]
    address 127.0.0.1
    use_node_name yes

[daphne.math.polytechnique.fr]
    address 129.104.3.2
    use_node_name yes
    df._dev_sdc1.warning 95
    df._dev_sdc1.critical 98
```

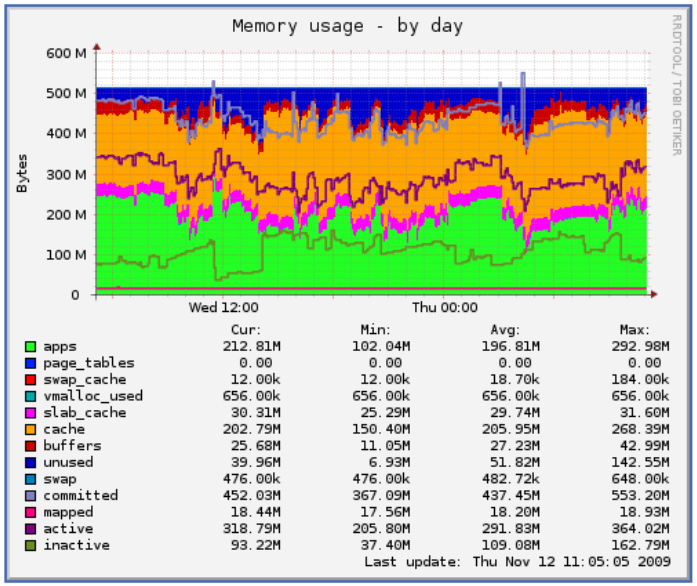
Munin : exemple

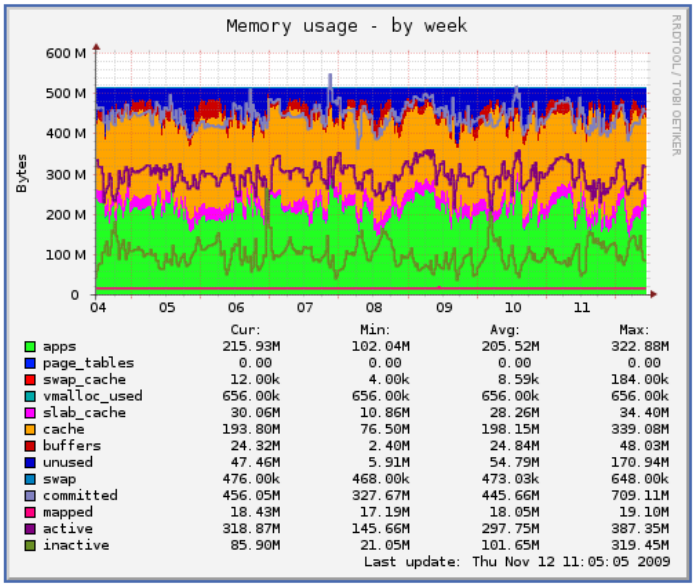
:: Load average





:: Memory usage





Munin : bilan

Avantages :

- très facile à installer : à la première exécution, munin-node découvre sur le serveur toutes les données accessibles ;
- adaptables à toutes les situations grâce à des plugins.
- possibilité d'envoyer des notifications

Défaut :

- trop de données immédiatement disponibles, pas si simple de les enlever.

Ganglia

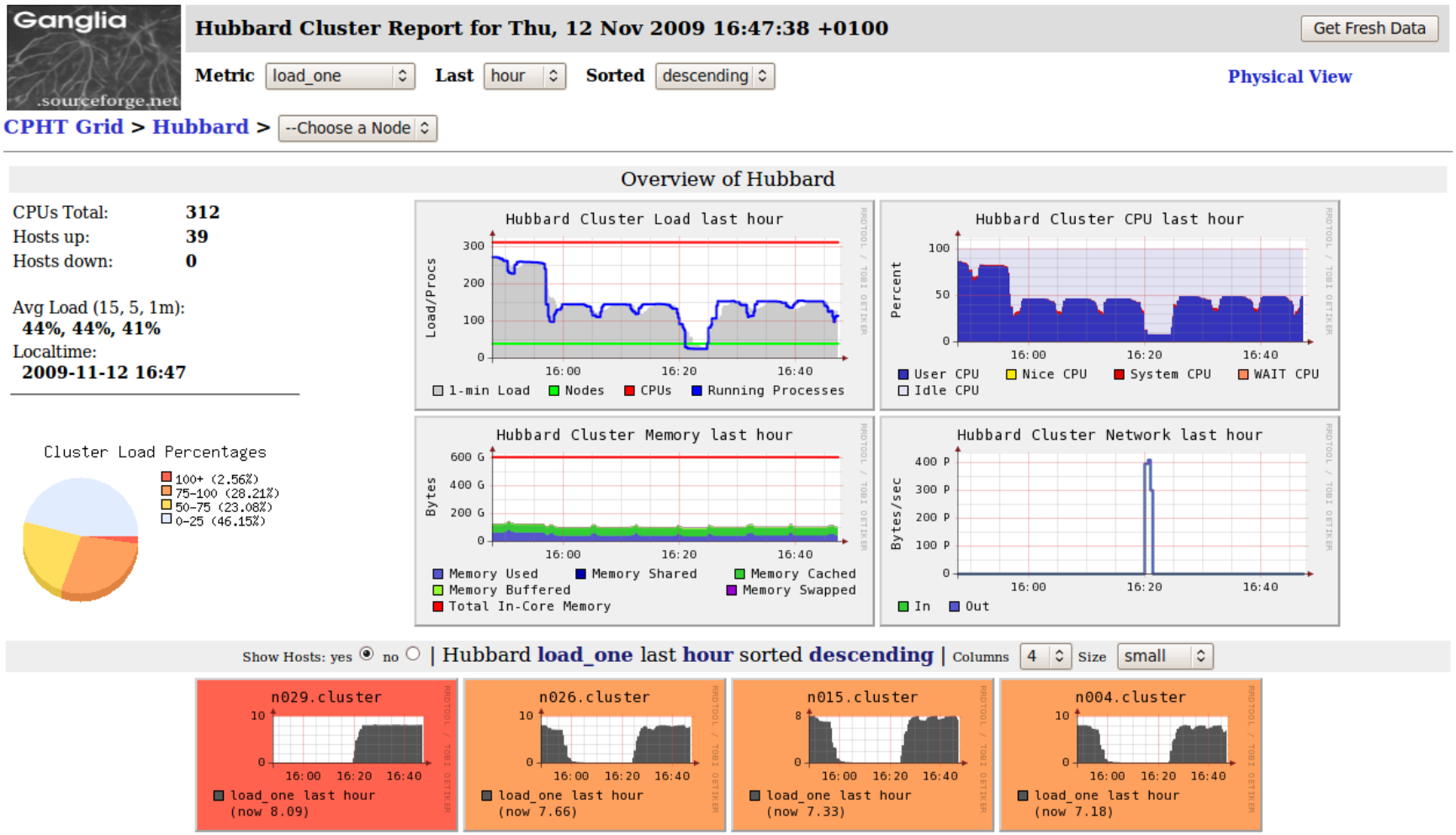
Ganglia est un système de supervision distribué plus particulièrement destiné aux clusters et aux grilles de calcul.

Ganglia : Architecture

Chaque nœud exécute un démon `gmond` qui envoie les données vers le démon `gmond` sur le nœud maître du cluster. Un démon `gmetad` sur la console de supervision de la grille collecte les informations de chaque nœud maître, met à jour des bases RRD et crée des résumés au format XML. Enfin, une application web écrite en PHP rassemble les informations et crée les graphiques à la demande.

Il est possible d'ajouter des *métriques* (données) par la commande en ligne `gmetric`. On peut donc écrire des scripts pour étendre les possbilités de ganglia.

Ganglia : Exemple



Ganglia : Bilan

Avantages :

- Intégré aux distributions orientées cluster HPC, configuration minimale dans ce cas,
- Extensible

Défauts :

- Trop orienté HPC
- Pas de système de notification intégré.

Cacti

Cacti est un système d'acquisition de données et de création de graphiques basés sur RRDtool. C'est un produit pur PHP/MySQL : configuration par l'interface web, tout est stocké dans une base MySQL.

Son ambition est d'utiliser toute la puissance de RRDtool tout en gommant la complexité technique.

Cacti : Architecture

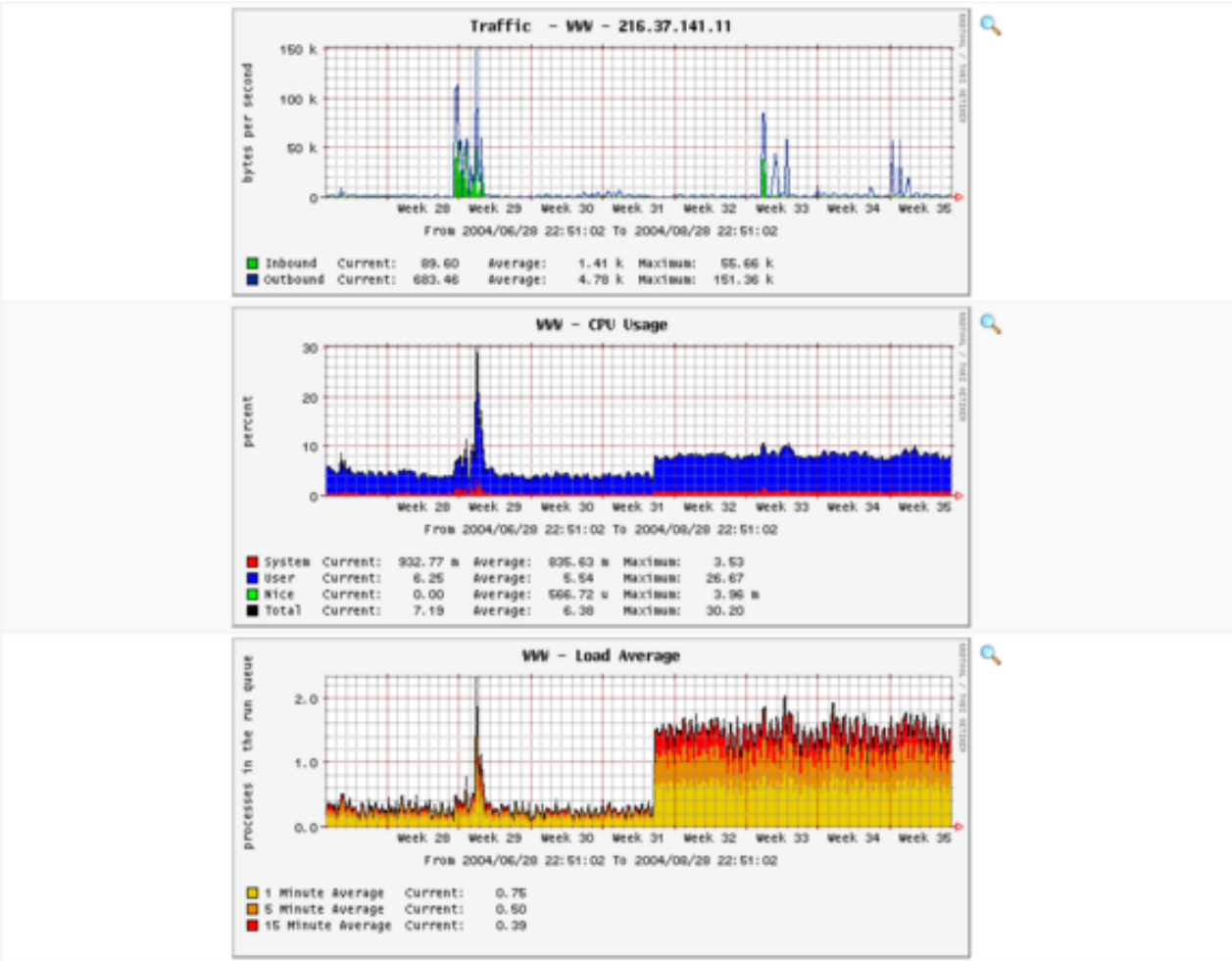
On définit d'abord les sources de données. Elles seront obtenues par SNMP ou via un script.

Cacti relève l'ensemble des données toutes les 5 minutes.

On crée ensuite des graphes associés aux données. Les graphes sont produits à la demande.

Pour faciliter la configuration, les types de données, les graphes et les équipements à monitorer sont stockés sous forme de modèle (*template*).

Cacti : Exemple



Cacti : Bilan

Avantages :

- produit bien fourni, facile à installer pour le non-administrateur,
- templates,

Défaut :

- configuration au clic, pas de fichier texte à éditer (sauvegarder, versionner).

Autres applications de supervision graphique

cricket basé sur MRTG avec quelques améliorations : RRDtool systématique, graphes à la demande, meilleur suivi des interfaces via SNMP.

torrus est une alternative à MRTG ou Cacti, mais moins centré sur SNMP.

ntop est orienté supervision réseau (tri par protocoles, etc.)

NetworkWeathermap permet de visualiser l'état d'un réseau sur la base de données provenant de MRTG ou de Cacti.

et j'en oublie sûrement !

Tableaux de bord

Tableaux de bord

Ces logiciels ont pour fonctions de :

- vérifier la disponibilité des services et des ressources,
- réagir aux alertes en notifiant l'administrateur ou en redémarrant des services,
- synthétiser l'état du système d'information sur une page web.

Ces logiciels permettent souvent de créer des graphiques avec les données obtenues, mais sans la flexibilité des logiciels de suivi graphique.

Monit

Monit est fait pour superviser un serveur unique et les services réseaux accessibles depuis ce serveur.

Monit peut également servir d'agent à un serveur M/Monit tournant sur une console centrale. Contrairement à Monit, M/Monit n'est pas libre.

Monit : architecture

Monit est fait pour superviser les services tournant sur une machine donnée et agir en cas de problème (mail à un administrateur, redémarrage d'un service planté, arrêt en cas de surcharge, etc.)

Monit peut également faire des contrôles d'intégrité de fichiers (pour recharger un service après modification de la configuration ou signaler une intrusion).

Monit : Configuration

```
# Monit control file
set daemon 120                          # Poll in 2-minute intervals
set logfile syslog facility LOG_daemon # Default facility is LOG_USER
set mailserver mail.foo.bar             # Default smtp server is localhost
set alert sysadm@foo.bar                # Alert system admin on any event
set httpd port 2812 address localhost
    allow localhost
    allow admin:monit

check process apache with pidfile "/usr/local/apache/logs/httpd.pid"
      start = "/etc/init.d/httpd start"
      stop = "/etc/init.d/httpd stop"
      if failed port 80 and protocol http
         and request "/cgi-bin/printenv" then restart
      if cpu usage is greater than 60 percent for 2 cycles then alert
      if cpu usage > 98% for 5 cycles then restart
      if 2 restarts within 3 cycles then timeout
  alert foo@bar.baz
```

Monit : Exemple

Monit Service Manager

Monit is running on imac.local with *uptime, 0m* and monitoring:

System	Status	Load	CPU	Memory
localhost	running	[0.14] [0.21] [0.15]	5.4%us, 1.9%sy	46.6% [1468724 kB]

Process	Status	Uptime	CPU	Memory
apache	running	11h 26m	0.0%	0.2% [8396 kB]

Filesystem	Status	Space usage	Inodes usage
HD1	accessible	58.2% [277474.5 MB]	58.2% [71097477 objects]
TM	accessible	52.3% [249694.0 MB]	52.3% [63921674 objects]

File	Status	Size	Permission	UID	GID
monit	accessible	342492 B	0555	0	0

Directory	Status	Permission	UID	GID
usr	accessible	755	0	0

Host	Status	Protocol(s)
quake3	online with all services	[generic] at port 27980
tildeslash	online with all services	[IMAP] at port 993 | [HTTP] at port 80
reddit	Connection failed	[HTTP] at port 80
fwd.pulver.com	online with all services	[SIP] at port 5060

Monit : Bilan

Avantages :

- Petit, très simple et efficace.
- Actions de réparation très simples à mettre en œuvre.

Défauts :

- Monit seul est mono-serveur.
- La solution complète n'est pas libre.

Xymon/Hobbit

Xymon est le nouveau nom de Hobbit. C'est une réécriture complète du vénérable Big Brother.

Xymon/Hobbit : Fonctionnement

Une console centrale collecte et stocke les données de supervision (actuelles et historique). Elle héberge généralement l'interface web.

Des serveurs effectuent des tests réseaux (ping, FTP, SMTP, LDAP(S), etc.) et envoient leurs rapports à la console centrale.

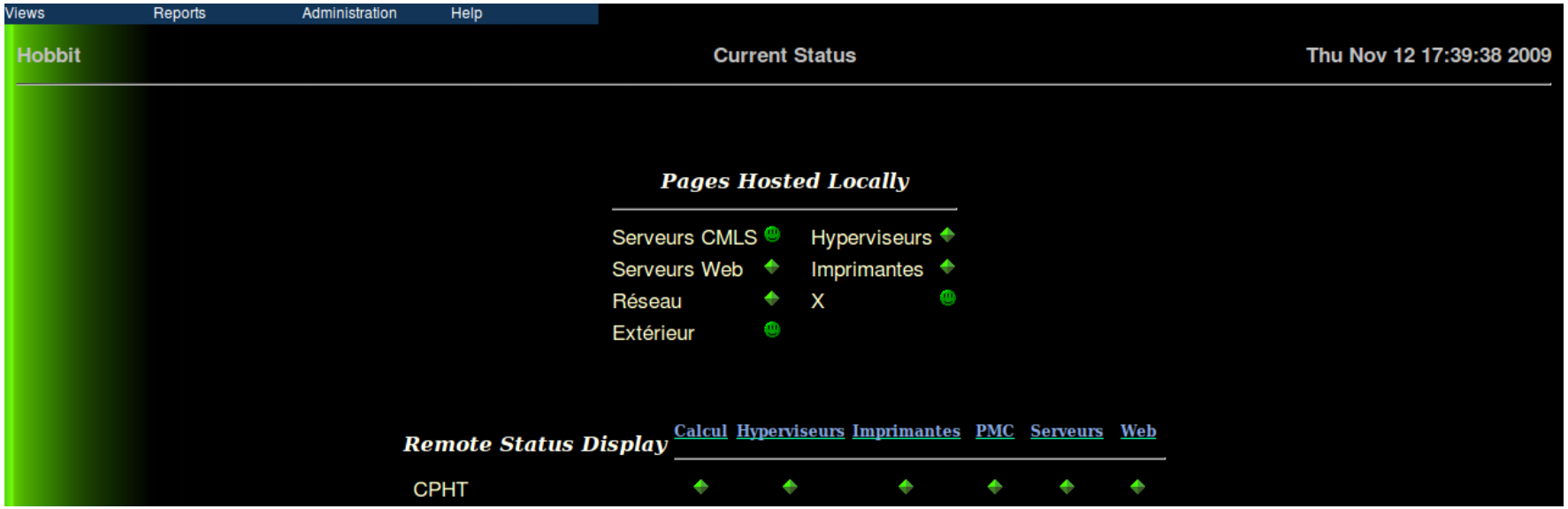
Un agent tourne sur chaque serveur à superviser et publie ses résultats (charge CPU, mémoire, disques, ports réseaux) à la console centrale.

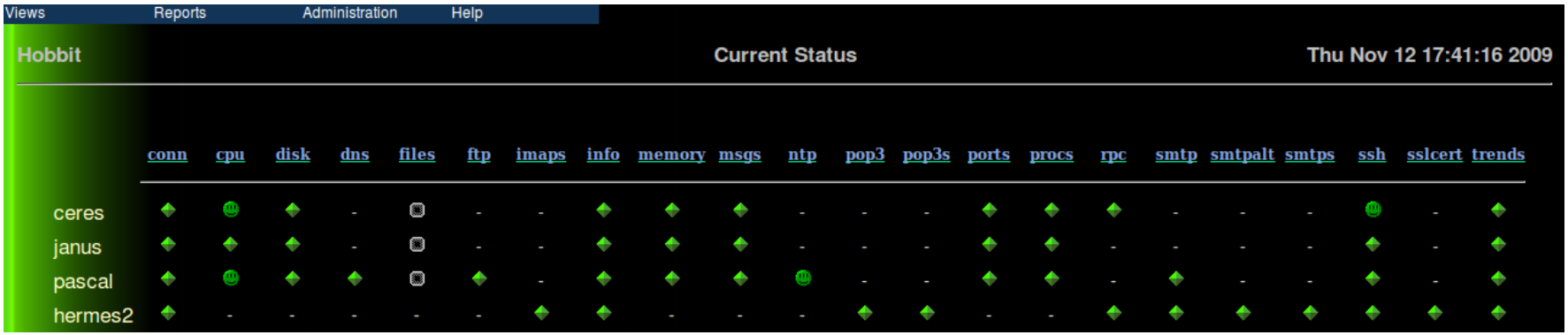
En cas de problème, une alerte est envoyée au responsable du service.

Xymon/Hobbit : Configuration

```
# Master configuration file for Hobbit
#
page CMLS Serveurs CMLS
129.104.3.2     daphne # ssh ftp smtp
129.104.3.3     athena # ssh dns ntp
129.104.3.100   cache  # ssh squid
129.104.3.8     hermes # ssh pop3 pop3s imaps smtp smtps smtpalt depends=(smtps:hermes/smtp) depends=(smtpalt:hermes/smtp)
129.104.3.224   phebus # ssh cupsd
129.104.3.231   lechesis # ssh ldap ldaps

page hyperviseurs Hyperviseurs
129.104.3.251   zeus # ssh

page web Serveurs Web
127.0.0.1       chronos # bbd cont;http://chronos.math.polytechnique.fr/hobbit/;Current[[:space:]]Status HIDEHTTP
129.104.3.6     thot # ssh rpc cont;http://www.math.polytechnique.fr/;Laurent[[:space:]]Schwartz HIDEHTTP

page print Imprimantes
129.104.3.243   amphi
129.104.3.247   couleur
129.104.3.240   reunion
```

Xymon/Hobbit : Exemple





Xymon/Hobbit : Bilan

Avantages :

- Produit simple à installer, à configurer et à utiliser

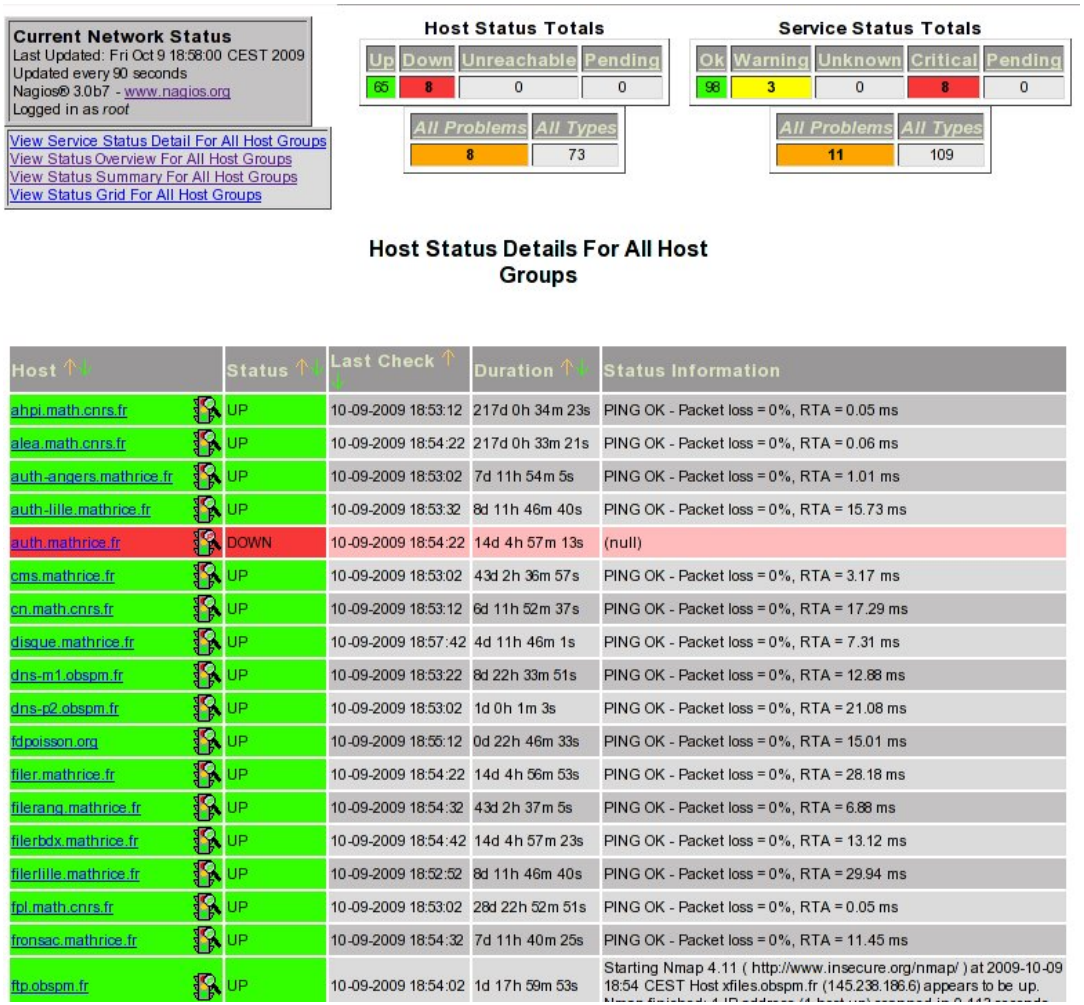
Défauts :

- Retrait d'un serveur ou d'un service peu intuitif
- Pas d'action de maintenance.

Nagios

Nagios est l'outil de tableau de bord le plus complet et configurable de cette présentation. Il sera détaillé plus amplement demain.

Nagios : Exemple



Current Network Status
Last Updated: Fri Oct 9 18:58:00 CEST 2009
Updated every 90 seconds
Nagios® 3.0b7 - www.nagios.org
Logged in as *root*

View Service Status Detail For All Host Groups
View Status Overview For All Host Groups
View Status Summary For All Host Groups
View Status Grid For All Host Groups

Host Status Totals

Up	Down	Unreachable	Pending
65	8	0	0

All Problems	All Types
8	73

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
98	3	0	8	0

All Problems	All Types
11	109

Host Status Details For All Host Groups

Host	Status	Last Check	Duration	Status Information
ahpi.math.cnrs.fr	UP	10-09-2009 18:53:12	217d 0h 34m 23s	PING OK - Packet loss = 0%, RTA = 0.05 ms
alea.math.cnrs.fr	UP	10-09-2009 18:54:22	217d 0h 33m 21s	PING OK - Packet loss = 0%, RTA = 0.06 ms
auth-angers.mathrice.fr	UP	10-09-2009 18:53:02	7d 11h 54m 5s	PING OK - Packet loss = 0%, RTA = 1.01 ms
auth-lille.mathrice.fr	UP	10-09-2009 18:53:32	8d 11h 46m 40s	PING OK - Packet loss = 0%, RTA = 15.73 ms
auth.mathrice.fr	DOWN	10-09-2009 18:54:22	14d 4h 57m 13s	(null)
cms.mathrice.fr	UP	10-09-2009 18:53:02	43d 2h 36m 57s	PING OK - Packet loss = 0%, RTA = 3.17 ms
cn.math.cnrs.fr	UP	10-09-2009 18:53:12	6d 11h 52m 37s	PING OK - Packet loss = 0%, RTA = 17.29 ms
disque.mathrice.fr	UP	10-09-2009 18:57:42	4d 11h 46m 1s	PING OK - Packet loss = 0%, RTA = 7.31 ms
dns-m1.obspm.fr	UP	10-09-2009 18:53:22	8d 22h 33m 51s	PING OK - Packet loss = 0%, RTA = 12.88 ms
dns-p2.obspm.fr	UP	10-09-2009 18:53:02	1d 0h 1m 3s	PING OK - Packet loss = 0%, RTA = 21.08 ms
fdpoisson.org	UP	10-09-2009 18:55:12	0d 22h 46m 33s	PING OK - Packet loss = 0%, RTA = 15.01 ms
filer.mathrice.fr	UP	10-09-2009 18:54:22	14d 4h 56m 53s	PING OK - Packet loss = 0%, RTA = 28.18 ms
filerang.mathrice.fr	UP	10-09-2009 18:54:32	43d 2h 37m 5s	PING OK - Packet loss = 0%, RTA = 6.88 ms
filerbdx.mathrice.fr	UP	10-09-2009 18:54:42	14d 4h 57m 23s	PING OK - Packet loss = 0%, RTA = 13.12 ms
filerlille.mathrice.fr	UP	10-09-2009 18:52:52	8d 11h 46m 40s	PING OK - Packet loss = 0%, RTA = 29.94 ms
fpl.math.cnrs.fr	UP	10-09-2009 18:53:02	28d 22h 52m 51s	PING OK - Packet loss = 0%, RTA = 0.05 ms
fronsac.mathrice.fr	UP	10-09-2009 18:54:32	7d 11h 40m 25s	PING OK - Packet loss = 0%, RTA = 11.45 ms
ftp.obspm.fr	UP	10-09-2009 18:54:02	1d 17h 59m 53s	Starting Nmap 4.11 (http://www.insecure.org/nmap/) at 2009-10-09 18:54 CEST Host xfiles.obspm.fr (145.238.186.6) appears to be up.

Nagios : Bilan

Avantages :

- C'est le standard de fait actuel,
- Finement configurable et extensible.

Défauts :

- Temps de configuration trop lourd pour un petit réseau

Autres applications de tableau de bord

Zabbix est un produit multi-plateformes très complet.

MonALISA

EyesOfNetwork est une solution complète basée sur CentOS, Nagios, Cacti et NetworkWeathermap.

et j'en oublie sûrement !

Conclusion

Il n'y a pas d'outil à tout faire. À chaque administrateur de trouver la combinaison de tableaux de bord, de suivis des journaux et de statistiques graphiques qui lui convient.