

Planificación prospectiva de los recursos

El administrador del sistema tiene que asegurarse de que las máquinas que administra estén operativas y ofrezcan un servicio de calidad a los usuarios, en función de los objetivos de rendimiento y de fiabilidad asignados.

Pero también tiene que anticipar la evolución de las necesidades, para poder de esta manera prevenir posibles saturaciones de recursos (cuellos de botella) adaptando, para ello, las configuraciones. Para llevarlo a cabo, tiene que implementar herramientas de recogida de datos y de seguimiento del rendimiento.

1. El paquete sysstat

El paquete `sysstat` ofrece un conjunto de herramientas de monitorización del estado de los recursos gestionados por el sistema, en particular del comando `sar`, de origen Unix, y del comando `iostat` para vigilar las entradas/salidas.

a. La recogida de información con sysstat

Una vez que el paquete haya sido instalado, posiblemente haya que configurarlo para que la herramienta `sysstat` recoja información con un intervalo regular.

Debian: activar la herramienta gracias al archivo `/etc/default/sysstat` (`ENABLED` a `true`).

Red Hat: la recogida de información está configurada por defecto para guardar 7 días. El archivo de configuración es `/etc/sysconfig/sysstat`, variable `HISTORY`.

La recogida de información es ejecutada por una tarea `cron`, que puede estar definida en `cron.daily` o mediante `/etc/cron.d/sysstat`.

La información se guarda en un archivo de registro del directorio `/var/log/sysstat` o `/var/log/sa`. La rotación de esos archivos de registro es automática, una vez al

mes.

b. El comando sar

Este comando funciona en dos modos:

Inmediato

Si le indicamos un intervalo de medida `t` y un número de medidas `n`. Mostrará el consumo de recursos instantáneo, con `n` medidas cada `t` segundos.

Diferido

Sin intervalo de medida `t`, el comando analiza los registros generados por `sysstat`.

Sus opciones permiten seleccionar los diferentes tipos de recursos, incluyendo visualizar el uso (procesador, memoria, entradas/salidas y red).

Procesador

Es el recurso visualizado por defecto, cuando el comando `sar` se utiliza sin opción.

Ejemplo

```
sar 3 5
Linux 2.6.32-431.20.3.el6.x86_64 (beta)    14/03/2020    _x86_64_    (2 CPU)

14:39:53   CPU   %user   %nice  %system  %iowait  %steal   %idle
14:39:56   all    0,00    0,00    0,50    0,00    0,00   99,50
14:39:59   all    0,00    0,00    0,17    0,83    0,00   99,00
14:40:02   all    0,33    0,00    1,50    1,00    0,00   97,17
14:40:05   all    0,00    0,00    0,17    0,00    0,00   99,83
14:40:08   all    0,00    0,00    0,33    0,00    0,00   99,67
Media:     all    0,07    0,00    0,53    0,37    0,00   99,03
```

Uso de `sar` en modo instantáneo, 5 medidas de la actividad del procesador, cada 3 segundos.

Memoria

La opción `-r` permite monitorear el uso de la memoria viva:

```
sar -r
Linux 2.6.32-431.20.3.el6.x86_64 (beta)    14/03/2020    _x86_64_    (2 CPU)

13:29:02    LINUX RESTART

14:29:13    LINUX RESTART

14:30:01  kbmemfree kbmemused %memused kbbuffers kbcached kbcommit %commit
14:40:01   3409348   447560   11,60   33092   219008   205828   2,62
Media :   3409348   447560   11,60   33092   219008   205828   2,62
```

Los valores de ocupación de la memoria pueden llevar a engaño, debido a la caché de las entradas/salidas. Linux intenta optimizar el uso de la memoria viva. Cada vez que lee los datos desde un espacio de almacenamiento, conserva en la medida de lo posible su caché de entradas/salidas. Mientras siga habiendo memoria viva disponible, la caché de entradas/salidas irá aumentando. Si un aumento de carga necesita más memoria, el núcleo reducirá automáticamente el tamaño de la caché para poder asignar la memoria solicitada.

Por lo tanto, es normal tener poca memoria viva disponible si el tamaño de la caché de entradas/salidas es importante. Para estimar el margen de maniobra del núcleo, es más lógico observar el conjunto de memoria disponible + caché de entradas/salidas.

El comando `top` clasifica por defecto los procesos según su consumo de tiempo de procesador, pero al pulsar la tecla `M`, podemos solicitar una clasificación sobre el consumo de la memoria.

Por otro lado, un problema de memoria puede traducirse en un aumento de las colas de espera de entradas/salidas. De hecho, si el sistema tiene muy poca memoria viva, almacenará en el disco cada vez más páginas de memoria (swap), lo que ralentizará considerablemente el rendimiento, hará crecer la solicitud de acceso a los discos y provocará una espiral de saturación: cada vez habrá más procesos en espera de entradas/salidas.

Discos

La opción `-b` permite vigilar el uso de los discos:

```
sar -b
Linux 2.6.32-431.20.3.el6.x86_64 (beta)    14/03/2020    _x86_64_    (2 CPU)

13:29:02    LINUX RESTART

14:29:13    LINUX RESTART

14:30:01    tps    rtps    wtps    bread/s    bwrtn/s
14:40:01    0,31    0,01    0,31    0,56    2,47
Media:    0,31    0,01    0,31    0,56    2,47
```

Otras opciones

La opción `-A` permite obtener el conjunto de la información recogida.

La opción `-f` permite especificar qué archivos se utilizarán para recoger esta información.

Las opciones `-s` (*start*) y `-e` (*end*) permiten limitar el período de observación.

Ejemplo

```
sar -r -s 14:30:00 -e 14:50:00 -f /var/log/sa/sa14
Linux 2.6.32-431.20.3.el6.x86_64 (beta)    14/03/2021    _x86_64_    (2 CPU)

14:30:01    kbmempfree    kbmempused    %mempused    kbbuffers    kbcached    kbcommit    %commit
14:40:01    3409348    447560    11,60    33092    219008    205828    2,62
Media:    3409348    447560    11,60    33092    219008    205828    2,62
```

Estado de la memoria el 14 de marzo de 2021 entre las 14:30 h y las 14:50 h.

2. El daemon collectd

`collectd` es un programa, open source, de recogida de datos de la actividad del sistema. Se trata de un daemon que mide a intervalos regulares (10 segundos por defecto) los valores de diferentes contadores que reflejan el rendimiento del sistema, que almacena en archivos.

Su objetivo es analizar el rendimiento (falta de recursos y cuellos de botella) y la planificación prospectiva de recursos (seguimiento de la evolución del consumo de recursos).

`collectd` no permite explotar directamente la información obtenida, pero puede realizar archivos de recogida de la información según diferentes formatos estándares, que pueden ser manipulados por aplicaciones de presentación de datos. Por defecto, usa el formato RRD (*Round Robin Database*), formato muy usado por las aplicaciones de presentación gráfica de datos estadísticos.

`collectd` es muy eficiente y consume pocos recursos. Está escrito en C y concebido de manera muy modular. Dispone de muchos plug-ins para conectarlo a otras aplicaciones, ya sea para monitorearlas o para que integren sus resultados (Apache, DNS, MySQL, Oracle, Nagios...).

`collectd` puede funcionar en red, enviando datos a uno o varios servidores (en modo *push*, es decir, por iniciativa de los sistemas locales y sin *polling* desde un servidor). La transferencia de datos puede hacerse en IPv4 o IPv6, en modo unicast o multicast, o incluso también a través de un proxy. También puede funcionar en modo local, sin necesidad de ninguna configuración de red.

a. Instalación

La instalación puede hacerse a partir de los archivos fuente (ver <http://collectd.org>) o de un paquete en formato Debian o Red Hat (en este caso hay que usar los paquetes de Fedora).

Ejemplo

```
apt-get install collectd
```

```
Leyendo lista de paquetes... Hecho
```

```
Creando árbol de dependencias
```

```

Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
[...]
Se necesita descargar 16,5 MB de archivos.
Se utilizarán 47,4 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] S
[...]

```

b. Configuración

La configuración por defecto permite ejecutar inmediatamente el daemon. Para un uso más personalizado, habría que modificar el archivo de configuración, `/etc/collectd.conf` o `/etc/collectd/collectd.conf`.

Ejemplo

Archivo de configuración por defecto, después de una instalación en una distribución Debian (sin comentarios).

```

vi /etc/collectd/collectd.conf
# sin comentarios
FQDNLookup true
LoadPlugin syslog
<Plugin syslog>
    LogLevel info
</Plugin>
LoadPlugin battery
LoadPlugin cpu
LoadPlugin df
LoadPlugin disk
LoadPlugin entropy
LoadPlugin interface
LoadPlugin irq
LoadPlugin load
LoadPlugin memory
LoadPlugin processes
LoadPlugin rrdtool

```

```

LoadPlugin swap
LoadPlugin users
<Plugin df>
    FSType rootfs
    FSType sysfs
    FSType proc
    FSType devtmpfs
    FSType devpts
    FSType tmpfs
    FSType fusectl
    FSType cgroup
    IgnoreSelected true
</Plugin>
<Plugin rrdtool>
    DataDir "/var/lib/collectd/rrd"
</Plugin>
<Include "/etc/collectd/collectd.conf.d">
    Filter "*.conf"
</Include>

```

Los diferentes elementos que se tienen que monitorear están especificados a través de plug-ins específicos (CPU, disco, memoria, interfaces de red, procesos, etc.). El formato de salida es RRD, el directorio de los datos es `/var/lib/collectd/rrd`.

En el directorio de recogida de datos, encontramos un subdirectorío para cada servidor monitoreado.

```

ls -l /var/lib/collectd/rrd/debian10/
total 112
drwxr-xr-x 2 root root 4096 abril 28 15:48 battery-0
drwxr-xr-x 2 root root 4096 abril 28 15:48 cpu-0
drwxr-xr-x 2 root root 4096 abril 28 15:48 cpu-1
drwxr-xr-x 2 root root 4096 abril 28 15:48 df-boot
drwxr-xr-x 2 root root 4096 abril 28 15:48 df-home
drwxr-xr-x 2 root root 4096 abril 28 15:48 df-root
drwxr-xr-x 2 root root 4096 abril 28 15:48 df-tmp
drwxr-xr-x 2 root root 4096 abril 28 15:48 df-var
drwxr-xr-x 2 root root 4096 abril 28 15:48 disk-dm-0

```

```

drwxr-xr-x 2 root root 4096 abril 28 15:48 disk-dm-1
drwxr-xr-x 2 root root 4096 abril 28 15:48 disk-dm-2
drwxr-xr-x 2 root root 4096 abril 28 15:48 disk-dm-3
drwxr-xr-x 2 root root 4096 abril 28 15:48 disk-dm-4
drwxr-xr-x 2 root root 4096 abril 28 15:48 disk-dm-5
drwxr-xr-x 2 root root 4096 abril 28 15:48 disk-dm-6
drwxr-xr-x 2 root root 4096 abril 28 15:48 disk-sda
drwxr-xr-x 2 root root 4096 abril 28 15:48 disk-sda1
drwxr-xr-x 2 root root 4096 abril 28 15:48 disk-sda2
drwxr-xr-x 2 root root 4096 abril 28 15:48 disk-sda5
drwxr-xr-x 2 root root 4096 abril 28 15:48 entropy
drwxr-xr-x 2 root root 4096 abril 28 15:48 interface-enp0s10
drwxr-xr-x 2 root root 4096 abril 28 15:48 interface-lo
drwxr-xr-x 2 root root 4096 abril 28 15:48 irq
drwxr-xr-x 2 root root 4096 abril 28 15:48 load
drwxr-xr-x 2 root root 4096 abril 28 15:48 memory
drwxr-xr-x 2 root root 4096 abril 28 15:48 processes
drwxr-xr-x 2 root root 4096 abril 28 15:48 swap
drwxr-xr-x 2 root root 4096 abril 28 15:48 users

```

En el directorio del servidor local, encontramos toda una serie de directorios para los diferentes recursos de los que se recoge.

Inicio

El daemon está instalado como un servicio, por lo tanto puede ser activado manualmente gracias al comando:

```
systemctl start collectd
```

A continuación puede ser configurado en arranque automático:

```
systemctl enable collectd
```

Ejemplo

systemctl status collectd

systemctl status collectd

collectd.service - Statistics collection and monitoring daemon

Loaded: loaded (/lib/systemd/system/collectd.service; enabled;

vendor preset: enabled)

Active: active (running) since Tue 2020-04-28 15:48:12 CEST; 4min 27s ago

[...]

c. Explotación de los datos de collectd

Como se ha visto, `collectd` no permite explotar directamente la información obtenida. Hay que conectarlo a aplicaciones de presentación de datos, programas específicos u otro tipo de software.

Usa por defecto el formato RRD (*Round Robin database*), formato estándar para los archivos de datos estadísticos. Puede generar gráficos o tablas, con programas de tipo RRDtool, kcollectd o ddraw, o con programas específicos usando las bibliotecas especializadas (Python, Perl, C, Java...).

collection3

collection 3 es una aplicación de presentación de datos recogidos por `collectd`, está integrada en el paquete `collectd`. El paquete `collectd` contiene, a título de ejemplo, un conjunto de scripts CGI de generación de gráficos, escritos en Perl.

Ejemplo

Después de haber instalado el paquete Debian.

cd /usr/share/doc/collectd/examples**ls**

bin etc lib README share

Estos diferentes directorios tienen que ponerse en un directorio virtual gestionado por un servidor web. Los scripts en Perl necesitan la instalación previa de diferentes bibliotecas y módulos Perl (ver el archivo `README`).

collectd-web

Como ilustración de las aplicaciones de formato y presentación de los datos recogidos por `collectd`, vamos a presentar una solución open source especialmente concebida para explotar los datos de `collectd`, a partir de cualquier servidor web (Apache, Nginx, o incluso un simple servidor Python dedicado): `collectd-web`.

Instalación

La instalación se hace a partir de un archivo comprimido. Hay que copiar la arborescencia de los directorios en un directorio gestionado por un servidor web. Para ello, usaremos el servidor `apache2`, que ya está instalado en el sistema, y lo configuraremos con un nuevo servidor virtual.

Instalación de los scripts y copia en un directorio del servidor `apache2`:

```
wget https://github.com/downloads/httpdss/collectd-web/collectd-web_0.4.0.tar.gz
tar -zxvf collectd-web_0.4.0.tar.gz
mv collectd-web /var/www/collectd
chmod +x /var/www/collectd/cgi-bin/*.cgi
chmod +x /var/www/collectd/cgi-bin/*.pl
```

Para que los scripts Perl funcionen correctamente hay que instalar la biblioteca Perl de gestión del formato RRD, así como diferentes módulos Perl (usando el comando `cpan install`, proporcionado con Perl). El script `/var/www/collectd/check_deps.sh`, proporcionado con `collectd-web`, permite comprobar que todos los módulos estén instalados correctamente.

```
apt-get install librrds-perl
[...]
cpan install Config::General
[...]
cpan install Regexp::Common
[...]
cpan install HTML::Entities
[...]
```

```

cpan install CGI
[...]
cpan install RRD:Simple
[...]
cpan install JSON
[...]
cd /var/www/collectd
./check_deps.sh
Carp looks ok
CGI looks ok
CGI::Carp looks ok
HTML::Entities looks ok
URI::Escape looks ok
RRDs looks ok
Data::Dumper looks ok
JSON looks ok

```

Configuración del servidor `apache2` con la página `collectd` por defecto

```

vi /etc/apache2/sites-available/000-default.conf
#DocumentRoot /var/www/html
DocumentRoot /var/www/collectd

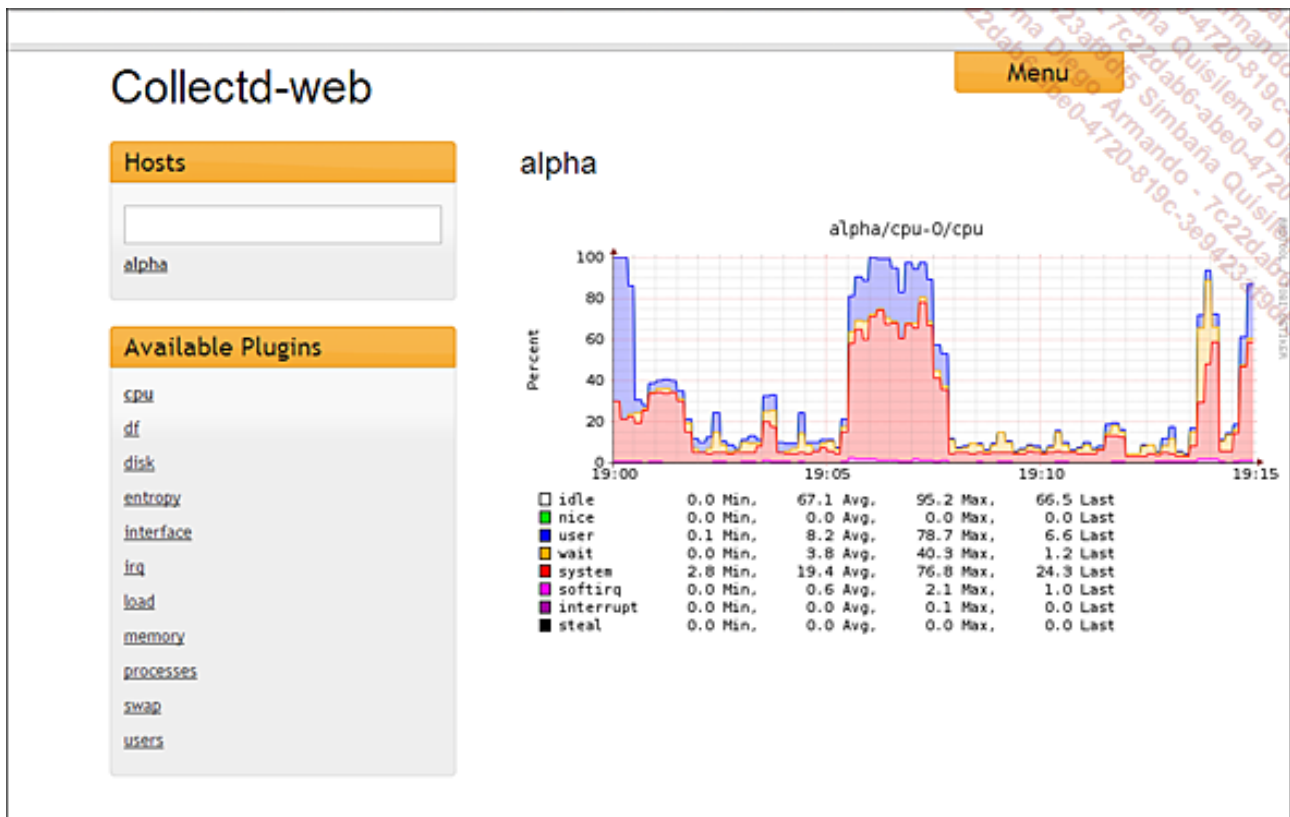
```

Uso

Para acceder al programa, basta con usar un navegador web y solicitar la URL del servidor donde está instalado `collectd`:

```
http://nombreServidor
```

La página principal muestra la lista de los servidores supervisados por `collectd` y, una vez que uno de ellos haya sido seleccionado, permite elegir el recurso sobre el que queramos ver las estadísticas.



3. Las soluciones de supervisión

La monitorización de la actividad de servidores múltiples en el seno de una red compleja es difícil y necesita generalmente herramientas especializadas. Se pueden implementar soluciones globales, que cubran el conjunto de elementos de hardware interconectados (switches, routers, firewalls, bahías de almacenamiento, servidores, etc.), con ayuda de agentes que utilizan protocolos de supervisión (SNMP) y proporcionan información en tiempo real hacia los servidores de supervisión.

Estos programas complejos pueden ser propietarios (HP Openview o IBM Tivoli) u open source, los más conocidos en esta última categoría son Nagios, MRTG, Cacti e Icinga2 (originalmente un fork de Nagios).

Ejemplo

Consola de supervisión de Nagios:

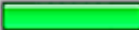

Nagios Tactical Monitoring Overview

Tactical Monitoring Overview
 Last Updated: Fri Jan 11 11:48:14 CST 2008
 Updated every 90 seconds
 Nagios® 3.0rc1 - www.nagios.org
 Logged in as nagiosadmin

Monitoring Performance

Service Check Execution Time: 0.02 / 10.28 / 0.445 sec
 Service Check Latency: 0.00 / 0.85 / 0.153 sec
 Host Check Execution Time: 0.26 / 4.06 / 3.147 sec
 Host Check Latency: 0.00 / 0.94 / 0.535 sec
 # Active Host / Service Checks: 17 / 175
 # Passive Host / Service Checks: 0 / 0

Network Outages
 0 Outages

Network Health
 Host Health: 
 Service Health: 

Hosts

| | | | |
|--------|---------------|-------|-----------|
| 0 Down | 0 Unreachable | 17 Up | 0 Pending |
|--------|---------------|-------|-----------|

Services

| | | | | |
|------------|-----------|-----------|--------|-----------|
| 2 Critical | 4 Warning | 0 Unknown | 169 Ok | 0 Pending |
|------------|-----------|-----------|--------|-----------|

2 Unhandled Problems
 4 Unhandled Problems

Monitoring Features

| Flap Detection | Notifications | Event Handlers | Active Checks | Passive Checks |
|--|---|---|---|---|
| Enabled All Services Enabled No Services Flapping All Hosts Enabled No Hosts Flapping | Enabled All Services Enabled All Hosts Enabled | Enabled All Services Enabled All Hosts Enabled | Enabled All Services Enabled All Hosts Enabled | Enabled All Services Enabled All Hosts Enabled |

2/32 **Tactical Overview**