Universidad de Granada

Ingeniería de Servidores

IOStat, comprobación experimental de su funcionamiento y significado de algunas medidas que más afecten al rendimiento

27 de abril de 2015

Índice

1. Resumen

Monitorizar el empleo de discos es una función básica en las competencias de un administrador de un servidor, o incluso podría ser aprovechado por un usuario medio.

A lo largo de este texto se analizará un monitor de dispositivos llamado `IOStat`, que viene por defecto instalado en algunos sistemas UNIX y se constatará mediante la ejecución de un experimento en que se medirán distintos tipos de medios de almacenamiento mediante un experimento confeccionado a tales fines: se probarán distintos tipos de disco duro: IDE,SATA; distintos formatos de partición: NTFS, FAT32,…; y distintas revoluciones por minuto: 3600 rpm, 5400 rpm y 7200 rpm.

Asimismo, también efectuaremos un experimento en el que analizaremos su uso como monitor de CPUs: se probarán varios tipos de CPU, variando factores como la velocidad(en Hz), el número de cores, …para correr un programa del que es conocido que somete a la CPU a mucha carga.

2. Introducción

Cuando como administradores de un sistema estamos en la obligación de ofrecer una determinada calidad de servicio (cubrir necesidades mínimas de los usuarios que emplearán el sistema), y deseamos mejorar tanto la utilización de CPU como la utilización de memoria de disco, podemos usar un monitor de CPU o de dispositivos, respectivamente.

La pregunta es por qué es deseable optimizar esta parte de un sistema operativo, cuando hay muchos más campos que se pueden mejorar en un sistema. La respuesta a esta pregunta se basa en que en el empleo de disco se encuentran los mayores cuellos de botella en un ordenador. Porque una transferencia a disco es del orden de 2000000 de veces más lenta que una transferencia a caché de nivel L1, y del orden de 40 veces más lenta que la memoria RAM, y por tanto es potenciablemente optimizable. [1]

IOstat es un ejemplo de estos monitores. Se trata de un sampling monitor (monitor que funciona a intervalos regulares de tiempo), de tipo software, que recoge información sobre el empleo de CPU y de dispositivos, basándose en el empleo de la información almacenada en los sistemas UNIX en `/proc`.

Para proporcionar una calidad de servicio determinada, en ocasiones es importante detectar los cuellos de botella en el empleo de la memoria o de la CPU. Y `IOStat` puede ayudarnos a llevar a cabo ese fin. Por ejemplo, nos podría ayudar a rebalancear la carga entre discos duros; si por ejemplo tenemos varios discos duros y uno de ellos se emplea más que el resto, en términos de cantidad de datos totales escritos o leídos, puede ayudarnos a redistribuir los archivos a los que más se accede, localizando el disco duro en el que se encuentran; o si incluso uno de los discos duros es más rápido que el resto (información que también puede estraerse de los informes que genera el monitor), podemos colocar los archivos de los usuarios que más uso generen en dicho disco duro, o incluso los datos a los que más se accede, para optimizar el sistema. En IOStat no existe interacción con el administrador/analista, obteniendo sus datos de `/proc/diskstats` En muchas ocasiones, los archivos más accedidos en un sistema son los del sistema operativo, y por tanto el monitor nos podrá ayudar a determinar a qué disco o partición mover los archivos en función de la rapidez de cada una. [2]

Su uso principal es la monitarización de la memoria, ya que para la monitarización de la CPU existen herramientas mucho más completas, como el comando de UNIX `top`, pero constituyen sólo un front-end al sistema de archivos `/proc`, y no puede mostrar información que un sistema UNIX de por sí tampoco pueda. [3]

3. Memoria

3.1. IOStat [4]

El comando `iostat`, del paquete `sysstat` está disponible en UNIX empleado para monitorizar la carga de E/S de un sistema, así como el uso de la CPU. Para la monitorización E/S se basa en computar el tiempo que un dispositivo de E/S está activo en relación a su ratio medio de transferencia. El comando está escrito en C, y está disponible bajo licencia GNU Public License v2.

El comando genera dos informes estadísticos: uno sobre el uso de CPU y otro sobre uso de E/S. Para mostrar cada uno, ocultando el otro, basta usar `iostat -c` o `iostat -d` respectivamente. Respecto a las estadísticas CPU, cabe mencionar que si se está ejecutando el comando en un sistema multiprocesador, las estadísticas mostradas constituyen una media de todos los núcleos. Por defecto se muestran ambos informes. La primera vez que se ejecuta el comando, muestra las estadísticas recopiladas desde la última vez que se reinició el sistema. Las sucesivas veces se muestran estadísticas acumuladas desde la última vez que el comando reunió información sobre el sistema. La sintaxis del comando es:

```
iostat [opciones] [dispositivo] interval count
```

Donde `opciones` son los distintos flags que se le pueden pasar al comando, entre los que destacan, aparte de los ya mencionados:

- `-j {ID | LABEL | PATH | UUID |…}`: muestra para cada dispositivo la etiqueta, el UUID,…correspondiente, en función de lo que hayamos indicado en el comando en lugar del descriptor de bloques /dev/sdX-n usado para identificar a los dispositivos por defecto.
- `-k`: muestra las estadísticas expresándolas en kilobytes por segundo.
- `-m`: muestra las estadísticas expresándolas en megabytes por segundo.
- `-p`: proporciona estadísticas no sólo para cada dispositivo, sino también para las particiones presentes en cada dispositivo.
- `-t`: hace que en los informes se incluya también un timestamp con la hora a la que corresponden.
- `-x`: muestra estadísticas expandidas. Si no se usa esta opción, se muestran para cada dispositivo solamente los valores: `tps, kB_read/s, kB_wrtn/s, KB_read, kB_wrtn`. Si se emplea dicha opción se muestra toda la información descrita en 3.1.2, excepto los campos mencionados anteriormente.
- `-y`: omite el primero de los informes (información acumulada desde el último reinicio).
- `-z`: omite en los informes aquellos dispositivos para los que no se registró actividad durante la acumulación de información.

`interval` y `count` son dos números naturales. Son omitibles ambos, o sólo `count`. `interval` indica a `iostat` la duración del intervalo de tiempo, en segundos, durante el cual debe recopilar información; una vez vencido ese tiempo se mostrará por pantalla un reporte de información concerniente a ese periodo de tiempo. `count` indica al comando cuántos informes se quieren. Por ejemplo, si se llama al comando de la forma:

```
iostat 5 10
```

entonces `iostat` mostrará 10 informes, uno cada 5 segundos, tanto de estadísticas CPU como de estadísticas de memoria.

Si se omite el parámetro `count` entonces `iostat` mostrará información cada vez que venza el tiempo indicado en `interval`, hasta que se interrumpa la ejecución del comando.

Si se omiten ambos parámetros, se mostrará un único informe (CPU+memoria si no se ha indicado que se muestre uno de ellos solamente).

Por omisión se genera información para todos los dispositivos de memoria disponibles en el sistema, si no se indica lo contrario con el parámetro `dispositivo`. Por ejemlo, la siguiente llamada a `iostat`:

```
iostat /dev/sda /dev/sdb
```

generaría información únicamente para los dispositivos correspondientes a los descriptores `/dev/sda` y `/dev/sdb`.

3.1.1. Informe CPU

La información aportada incluye:

- `%user`: porcentaje de uso CPU en el nivel usuario (generados por la ejecución de aplicaciones).
- `%nice`: porcentaje de uso CPU correspondiente a procesos con prioridad cambiada.
- `%system`: porcentaje de uso CPU en el nivel kernel.
- `%iowait`: porcentaje de tiempo que la CPU ha estado ociosa esperando a peticiones E/S.
- `%steal`: porcentaje de tiempo empleado en espera por un núcleo virtual mientras el hipervisor servía a otro núcleo virtual. Este parámetro es útil cuando se están realizando virtualizaciones, por ejemplo con Virtualbox, y el número de virtualizaciones simultáneas es mayor que el número de núcleos físicos en la máquina.
- `%idle`: porcentaje de tiempo que la CPU estuvo ociosa mientras el sistema no tenía una petición E/S pendiente.

3.1.2. Informe dispositivos

La información mostrada puede incluir (en función de si se emplea la opción `-x` o no):

- `Device`: descriptor de dispositivo `/dev/sdX` o de partición `/dev/sdXn`, donde `X` es una letra única que identifica a cada dispositivo y `n` un número que identifica de forma unívoca cada partición para un dispositivo dado.
- `tps`: número de transferencias pro segundo a un dispositivo. Una transferencia es una petición E/S que puede incluir varias peticiones lógicas de E/S combinadas. Esto es, si se intentan leer varios datos desde disco duro simultáneamente, puede que se hallen en el mismo bloque de disco duro o en bloques contiguos, y varias peticiones de datos dan lugar a una única petición de E/S. Así, una petición de E/S no tiene un tamaño fijo.
- `KB_read/s`: número de KB leídos del dispositivo por segundo.
- `KB_wrtn/s`: número de KB escritos al dispositivo por segundo.
- `KB_read`: número de KB leídos en total desde el dispositivo.
- `KB_wrtn`: número de KB escritos en total al dispositivo.
- `rrqm/s`: número de peticiones de lectura combinadas por segundo.
- `wrqm/s`: núero de peticiones de escritura combinadas por segundo.
- `r/s`: número de peticiones de lectura(ya combinadas) completadas por segundo.
- `w/s`: número de peticiones de escritura(ya combinadas) completadas por segundo.
- `rKB/s`: equivalente a `KB_read/s`.

- `wKB/s`: equivalente a `KB_wrtn/s`.
- `avgrq-sz`: número medio de peticiones de lectura, en bloques (en los kernels más modernos 1 bloque equivale a 512KB) emitidas al dispositivo.
- `avgu-sz`: longitud media de la cola para las peticiones emitidas al dispositivo.
- `await`: media en milisegundos que tarda una petición en ser servida (incluyendo tiempo de servicio y de cola).
- `r_await`: media en milisegundos que tarda una petición de lectura en ser servida.
- `w_await`: media en milisegundos que tarda una petición de escritura en ser servida.
- `svctm`: media en milisegundos de tiempo de servicio para las peticiones E/S emitidas al dispositivo.
- `%util`: bandwith del dispositivo.

Referencias

[1] Ben Mildren. MySQL Team Technical Lead. Pythian.
Monitoring IO performance using iostat pt-diskstats. MySQL Conerence Expo 2013
url: http://www.percona.com/live/mysql-conference-2013/sites/default/files/slides/Monitoring-Linux-IO.pdf.

[2] Juan José Merelo
Solución de problemas en un sistema informático. Equilibrio de la carga de trabajo de E/S
url: http://geneura.ugr.es/~jmerelo/DyEC/Tema3/DyEC-Tema3.html.

[3] Sebastien Godard
IOStat README. Miscellaneous
url: https://github.com/sysstat/sysstat/blob/master/README.

[4] Linux User's Manual
man iostat.