Universidad de Granada

Ingeniería de Servidores

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

3 de mayo 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.

4. Evaluación de prestaciones con IOStat

Vamos a efectuar un análisis de prestaciones sobre varios tipos de discos duros, determinando la velocidad de cada disco en el copiado de archivos. Los discos duros podrán tener particiones de diferente tipo y tamaño.

4.1. Definición del sistema

El principal objetivo del estudio es comparar la velocidad del copiado de archivos en varios discos duros con particiones en distintos formatos. En otras palabras, determinar cuál es el tipo de almacenamiento más rápido. El estudio se centrará por tanto en discos duros. Se dispondrá de dos discos duros conectados a la misma placa madre, uno que alojará los datos a transferir, y otro que será el encargado de recibir los datos transferidos.

Obviaremos del equipo al que conectaremos los discos duros todo aquellos componentes que no interfieran en la transferencia de archivos.

4.2. Servicios proporcionados por el sistema

El servicio principal proporcionado por el sistema es la transferencia de datos entre discos duros.

4.3. Métricas

Llamaremos modelo de almacenamiento a la combinación de: sistema de archivos + tipo de conexión del disco duro + velocidad de rotación …. Para cada modelo de almacenamiento, se evaluará la velocidad de transferencia. Por tanto, la métrica empleada para evaluar las prestaciones del disco duro será: **número de kilobytes escritos en el dispositivo por segundo** (wKB/s en IOStat).

Asimismo, se hará un análisis de los siguientes datos referentes a los dispositivos, que condicionan la tasa de transferencia ($\approx$ velocidad de transferencia):

- Número de peticiones de escritura completadas en el dispositivo por segundo. (w/s en IOStat).
- Tamaño de las peticiones de escritura al dispositivo (avgrq-sz es la media de dichos tiempos en IOStat).

- Tamaño de la cola en el dispositivo durante la transferencia. (`avgqu-sz` representa la media del tamaño de la cola en IOStat).
- Tiempo que tardan las peticiones de escritura en ser servidas. (`w_await` proporciona la media de dicho tiempo en IOStat).
- Ancho de banda del dispositivo (`%util` en IOStat).

4.4. Parámetros

Se trabajará únicamente con discos duros magnéticos de 3.5" (diámetro del disco). Así los parámetros del sistema que influyen en las prestaciones serán:

- Tipo de conexión del disco duro (IDE, SATA, USB 2.0, USB 3.0 …).
- Tamaño de memoria caché del disco duro.
- Velocidad de rotación del disco duro (3600, 5400, 7200 rpm).
- Tipo de particionado del disco duro (NTFS, FAT32, …).

  [5]

Los parámetros de la carga que influyen en las prestaciones son:

- Tamaño de transferencia.
- Número de transferencias simultáneas.

Para seleccionar dichos parámetros nos hemos basado en el hecho empírico de que cuando se realiza una única transferencia, el ancho de banda del dispositivo no alcanza el máximo, mientras que si se están realizando varias transferencias simultáneas, el ancho de banda medio está muy cerca del 100 % y por tanto, el disco duro disminuye sus prestaciones.

4.5. Factores

Los factores seleccionados para este estudio son:

- Tipo de conexión del disco duro.
- Tipo de particionado del disco duro.
- Tamaño de caché del disco duro.
- Revoluciones por minuto del disco duro.
- Número de transferencias simultáneas $n = 1, 2, \ldots 5$

A pesar de que se han seleccionado como factores los tipos de disco duro y conexión, y la velocidad de rotación, debido a limitaciones de disponibilidad de hardware habrá casuísticas que no podrán evaluarse. Los experimentos se realizarán con el equipo sobre el que se testeará liberado de cargas innecesarias, por lo que consideraremos el error introducido entre medidas en distintos discos duros y el error introducido entre pruebas al mismo disco duro con distinto número de transferencias simultáneas insignificantes.

4.6. Técnica de evaluación

Se dispondrá de un equipo de sobremesa con las siguientes características hardware y software para realizar el experimento:

- Sistema Operativo: Ubuntu 14.04 LTS, 32 bits
- Memoria RAM: 1 GB
- CPU: Intel Celeron 3.06 GHz.

Por tanto la técnica de evaluación seleccionada es **medición sobre un sistema real**. Emplearemos el programa IOStat para efectuar la monitorización de la transferencia, tomando muestreos de información cada 2 segundos durante la transferencia de los ficheros.

4.7. Carga de trabajo

La carga consistirá en efectuar copias simultáneas de varios ficheros al disco duro. Se ha optado por mantener un fichero de tamaño fijo (599.8 MB) que se copiará 1,2,3…5 veces de forma simultánea desde uno de los discos del sistema al disco duro a evaluar.

4.8. Diseño experimental

Se empleará un diseño multi-factorial fraccionado, dado que no es posible evaluar todos los niveles de todos los factores que afectan al rendimiento.

4.9. Análisis de datos

4.10. Presentación de datos

5. Desarrollo experimental

Dispondremos de varios discos duros (entre paréntesis se incluye el código por el que identificaremos al disco de ahora en adelante):

[M1] Maxtor 6K040L0, IDE, 2MB caché, 7200 rpm.
[S1] Seagate ST34321A IDE, 128KB caché, 5400 rpm.
[S2] Seagate ST320410A IDE, 2MB caché, 5400 rpm.
[S3] Seagate ST320413A IDE, 512KB caché, 5400 rpm.
[S4] Seagate ST3320613AS SATA, 16MB caché, 7200 rpm.
[WD1] Western Digital WD800JD SATA, 8MB caché, 7200 rpm.
[WD2] Western Digital WDBUZG0010BBK externo, USB 3.0, 5400 rpm.

Se ha empleado la versión 10.0.2 del paquete `sysstat`.

Se emplearán particiones de tipo: ext4, FAT32, NTFS, exceptuando el disco [WD1], que sólo se ha podido evaluar formateado en ext4. Aunque la conexión incluida en [WD2] era USB 3.0, sólo ha podido evaluarse con USB 2.0, debido a las características del equipo empleado.

Se ha confeccionado un script `bash` que recoge para los discos duros que le indiquemos(a través de su punto de montaje en el sistema de archivos), las medias de todos los datos proporcionados por IOStat, tanto a nivel de CPU, como en la evaluación de dispositivos en un archivo de nombre `averages`. También proporciona los datos recogidos para cada parámetro, ordenados temporalmente según se recogieron en un archivo de nombre `data`. Realiza 5 iteraciones, realizando en cada iteración i copias simultáneas, donde i es el número de la iteración. Y almacena los datos

recogidos en una carpeta de nombre el modelo de disco duro (obtenido mediante el comando `lsblk`), con los datos recogidos clasificados en directorios identificados por el número de copias simultáneas. El script se incluye en el anexo.

Asimismo, el mencionado script se apoya en otro realizado en `python`, de nombre `iostat_plotter`, disponible para su descarga en GitHub. El software original, dado un archivo de salida de IOStat ejecutado de la forma:

```
iostat -c -d -x -t -m interval count > [archivo salida]
```

Generaba un informe `.html` de los resultados, así como gráficas de ellos. El software se ha modificado para cambiar el idioma de las gráficas, y para obtener archivos de texto plano con los resultados (medias y listado de resultados, esto es, los archivos `averages` y `data` descritos anteriormente). Se incluye, ya modificado, en el anexo.

6. Resultados

Cuadro 1: Tabla de resultados para [M1]

MB/s transferidos	**ext4**	**FAT32**	**NTFS**
1 copia	55.296		
2 copias simultáneas			
3 copias simultáneas			
4 copias simultáneas			
5 copias simultáneas			
Media pet. escritura completadas por segundo			
1 copia	443.368		
2 copias simultáneas			
3 copias simultáneas			
4 copias simultáneas			
5 copias simultáneas			
Tamaño medio peticiones escritura(KB)			
1 copia	255.432		
2 copias simultáneas			
3 copias simultáneas			
4 copias simultáneas			
5 copias simultáneas			
Longitud media de la cola de escritura			
1 copia	73.584		
2 copias simultáneas			
3 copias simultáneas			
4 copias simultáneas			
5 copias simultáneas			
Ancho de banda para el dispositivo			
1 copia			
2 copias simultáneas			
3 copias simultáneas			
4 copias simultáneas			
5 copias simultáneas			

Cuadro 2: Tabla de resultados para [M1]

MB/s transferidos	**ext4**	**FAT32**	**NTFS**

1 copias simultáneas	55.296	61.770	17.239
2 copias simultáneas	67.077	39.753	16.564
3 copias simultáneas	75.665	37.083	16.885
4 copias simultáneas	74.104	35.284	18.129
5 copias simultáneas	74.046	30.305	15.460
Media pet. escritura completadas por segundo			
1 copias simultáneas	443.368	497.128	138.541
2 copias simultáneas	537.268	339.491	133.580
3 copias simultáneas	606.042	322.767	137.490
4 copias simultáneas	600.829	317.436	148.088
5 copias simultáneas	601.842	280.018	126.727
Tamaño medio peticiones escritura(KB)			
1 copias simultáneas	255.432	254.476	254.606
2 copias simultáneas	255.617	232.524	253.313
3 copias simultáneas	255.542	228.435	251.001
4 copias simultáneas	252.472	221.354	250.423
5 copias simultáneas	251.938	216.399	249.447
Longitud media de la cola de escritura			
1 copias simultáneas	73.584	138.106	18.372
2 copias simultáneas	159.496	143.351	18.202
3 copias simultáneas	188.057	146.220	19.254
4 copias simultáneas	197.096	151.750	21.029
5 copias simultáneas	204.960	156.967	20.958
Ancho de banda para el dispositivo			
1 copias simultáneas	91.008	109.296	30.076
2 copias simultáneas	118.308	102.124	29.264
3 copias simultáneas	136.771	102.657	31.133
4 copias simultáneas	141.505	106.701	34.549
5 copias simultáneas	146.497	109.798	32.426

Anexo

```
#!/bin/bash

# Rutas de montaje de los discos a analizar
PATHS="/media/usuario/5206ba37-ffc6-4ba5-9de3␣/media/usuario/NTFS␣/"
# Nombre de fichero que se copiará
TFILE="testfile"
# Nombre que se le dará a cada uno de los ficheros transferidos
DEST="copiedfile"
# Número máximo de transferencias simultáneas
LIMIT=5
# Frecuencia de muestreo de iostat
FREQ="2"
# Flags de iostat
FLAGS="-y␣-c␣-d␣-x␣-t␣-m"
export LC_NUMERIC=en_US.UTF-8 LC_TIME=en_US.UTF-8


# Se borran las carpetas de destino
for p in ${PATHS}
do
    rm -r $p/temp/* 2> /dev/null
done


for k in `seq 1 $LIMIT`
do
    NUM_COPIES=$k

    # Para cada disco y un número de copias simultáneas k dado...
    for p in ${PATHS}
    do
        PIDS=""
        HDD=$(lsblk -io MODEL,MOUNTPOINT | sed '/^\s*$/d' |
        grep -B1 "$p" | head -1)
        DIR=${HDD// }
        DEVICE=$(df -h | grep ".*$p*" | grep -o "^[^[:blank:][:digit:]]*")

        LOG=./${DIR}/log-${k}.out

        mkdir $p/temp 2> /dev/null
        mkdir ${DIR} 2> /dev/null
        (iostat ${FREQ} ${FLAGS} ${DEVICE} > ${LOG})&
        IOSPID=$!

        for i in `seq 1 $NUM_COPIES`
        do
            cp ${TFILE} $p/temp/${i}${DEST} &
            PIDS="${PIDS}␣$!"
        done

        # Se espera a que terminan las copias para seguir
        wait ${PIDS}

        # Interrumpe la ejecución de IOStat. Las copias ya han terminado
        kill ${IOSPID} &> /dev/null
        wait ${IOSPID} &> /dev/null

        # Genera las gráficas y los archivos de datos
```

```
        ./iostat_plotter_v3.py ${LOG}
        mv REPORT ${LOG%.out}

        # Limpia el directorio para la siguiente ejecución
        rm -r $p/temp/* 2> /dev/null
    done

    sleep 2
done
```

Referencias

[1] Ben Mildren. MySQL Team Technical Lead. Pythian.
Monitoring IO performance using iostat and pt-diskstats. MySQL Conerence and 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.

[5] *Guía y tipos de discos duros*
url: http://discosduros.org/tipos-de-discos-duros/.