

Práctica - Benchmarking

Nombre: An Wei Pham Luo, Alejandro Povedano Atienza

En esta práctica usaremos la máquina física con el usuario "Usuario local".

Primera Parte. Leer las instrucciones antes de comenzar con la ejecución de las sentencias.

UnixBench es una herramienta de evaluación de un servidor Linux. Esta herramienta incluye evaluación de CPU, memoria, E/S, etc. Como resultado muestra un resumen agregado en el indicador conocido como Index Score.

Instala el *benchmark* UnixBench (github.com/kdlucas/byte-unixbench):

```
$ wget https://github.com/kdlucas/byte-unixbench/archive/v5.1.3.tar.gz
$ tar xzf v5.1.3.tar.gz
$ cd byte-unixbench-5.1.3/UnixBench
$ make
```

La ejecución por defecto tarda casi 30 minutos con una CPU y el doble con más de una. Para **reducir el tiempo de ejecución a unos 9 minutos**, vamos a reducir el número de repeticiones a 3 (una sola repetición para los programas más lentos) y a eliminar la segunda ejecución en paralelo:

```
$ ./Run -i 3 -c 1
```

Mientras se ejecuta, revisa su funcionamiento y las opciones que ofrece leyendo el fichero USAGE.

Observa también el código de algunos de los programas que lo componen: Whetstone, Dhrystone, Hanoi... ¿Qué hacen? ¿A qué tipo de *benchmark* corresponden? ¿Qué intentan medir? ¿En qué unidades proporcionan sus resultados? ¿Cómo se calcula el índice global de rendimiento ("Index Score" o "BYTE Index") a partir de los resultados proporcionados por cada programa?

Entrega: Copia los resultados y responde a las preguntas.

Whetstone: benchmark sintético que mide el rendimiento de las operaciones de punto flotante. Las unidades son: MWIPS (millones de instrucciones Whetstone por segundo).

Dhrystone: benchmark sintético que mide el rendimiento de las operaciones enteras (integer). Las unidades son: lps (loops per second).

Hanoi: benchmark sintético que resuelve un problema recursivo (torres de Hanoi)

Para calcular el Index Score de cada benchmark, se divide el score obtenido entre el baseline y se multiplica por 10.

Para el global se hace la media de todos los index (1722.4)

Benchmark Run: lun abr 01 2019 16:05:38 - 16:15:07
8 CPUs in system; running 1 parallel copy of tests

Dhrystone 2 using register variables	50631265.3 lps (10.0 s, 2 samples)
Double-Precision Whetstone	3590.7 MWIPS (19.2 s, 2 samples)
Excl Throughput	5457.0 lps (29.1 s, 1 samples)

File Copy 1024 bufsize 2000 maxblocks	859134.0 KBps (30.0 s, 1 samples)
File Copy 256 bufsize 500 maxblocks	220467.0 KBps (30.0 s, 1 samples)
File Copy 4096 bufsize 8000 maxblocks	2410781.0 KBps (30.0 s, 1 samples)
Pipe Throughput	1508795.0 lps (10.0 s, 2 samples)
Pipe-based Context Switching	227182.6 lps (10.0 s, 2 samples)
Process Creation	16096.8 lps (30.0 s, 1 samples)
Shell Scripts (1 concurrent)	14092.1 lpm (60.0 s, 1 samples)
Shell Scripts (8 concurrent)	4539.7 lpm (60.0 s, 1 samples)
System Call Overhead	1070429.8 lps (10.0 s, 2 samples)

System Benchmarks Index Values	BASELINE	RESULT	INDEX
Dhrystone 2 using register variables	116700.0	50631265.3	4338.6
Double-Precision Whetstone	55.0	3590.7	652.9
Exec1 Throughput	43.0	5457.0	1269.1
File Copy 1024 bufsize 2000 maxblocks	3960.0	859134.0	2169.5
File Copy 256 bufsize 500 maxblocks	1655.0	220467.0	1332.1
File Copy 4096 bufsize 8000 maxblocks	5800.0	2410781.0	4156.5
Pipe Throughput	12440.0	1508795.0	1212.9
Pipe-based Context Switching	4000.0	227182.6	568.0
Process Creation	126.0	16096.8	1277.5
Shell Scripts (1 concurrent)	42.4	14092.1	3323.6
Shell Scripts (8 concurrent)	6.0	4539.7	7566.1
System Call Overhead	15000.0	1070429.8	713.6

=====

System Benchmarks Index Score	1722.4
-------------------------------	--------

Segunda Parte. Leer las instrucciones antes de comenzar con la ejecución de las sentencias.

iozone es una utilidad de benchmarking de sistemas de ficheros que soporta E/S asíncrona, mmap, pthreads, etc.

Descarga la última versión estable (*stable tarball*) de IOzone (www.iozone.org), descomprime/desempaqueta, ve al directorio `iozone3_*/src/current` y ejecuta `make linux`.

Consulta la página de manual de iozone:
\$ `man ../../docs/iozone.1`

Obtén resultados con un fichero de pruebas de 100MB:
\$ `./iozone -s 100m`

Repite las pruebas añadiendo la opción `-I`. ¿Para qué sirve? ¿Qué efecto tiene en los resultados?

Repite las pruebas añadiendo la opción `-r 16k` junto con la anterior. ¿Para qué sirve? ¿Qué efecto tiene en los resultados?

Entrega: Copia los resultados y escribe un breve análisis de los mismos respondiendo a las preguntas.

./iozone -s 100m															
		kB freread	reclen	write	rewrite	read	reread	read	write	read	rewrite	read	fwrite	frewrite	fread
The top row is records sizes, the left column is file sizes															
Writer Report															
	4		8	16	32	64	128	256	512	1024	2048	4096	8192	16384	
102400	1974735														
Re-writer Report															
	4		8	16	32	64	128	256	512	1024	2048	4096	8192	16384	
102400	1834728														
Reader Report															
	4		8	16	32	64	128	256	512	1024	2048	4096	8192	16384	
102400	5138454														
Re-reader Report															
	4		8	16	32	64	128	256	512	1024	2048	4096	8192	16384	
102400	5785573														
Random Read Report															
	4		8	16	32	64	128	256	512	1024	2048	4096	8192	16384	
102400	4247530														
Random Write Report															
	4		8	16	32	64	128	256	512	1024	2048	4096	8192	16384	
102400	2560951														
Backward Read Report															
	4		8	16	32	64	128	256	512	1024	2048	4096	8192	16384	
102400	2062728														
Record Rewrite Report															
	4		8	16	32	64	128	256	512	1024	2048	4096	8192	16384	
102400	3328791														
Stride Read Report															
	4		8	16	32	64	128	256	512	1024	2048	4096	8192	16384	
102400	4524980														
Fwrite Report															
	4		8	16	32	64	128	256	512	1024	2048	4096	8192	16384	
102400	3181909														
Re-fwrite Report															
	4		8	16	32	64	128	256	512	1024	2048	4096	8192	16384	
102400	1610774														
Fread Report															
	4		8	16	32	64	128	256	512	1024	2048	4096	8192	16384	
102400	5040106														
Re-fread Report															
	4		8	16	32	64	128	256	512	1024	2048	4096	8192	16384	

Todos los resultados son iguales excepto la columna freread, que indica el rendimiento de lectura del fichero usando la librería fread(). El fichero en cuestión ya había sido leído con anterioridad.

- iozone -i [test-type]
- 0=write/rewrite
 - 1=read/re-read
 - 2=random-read/write
 - 3=Read-backwards
 - 4=Re-write-record
 - 5=stride-read
 - 6=fwrite/re-fwrite
 - 7=fread/Re-fread,
 - 8=random mix
 - 9=pwrite/Re-pwrite
 - 10=pread/Re-pread

- 11=pwritev/Re-pwritev
- 12=preadv/Re-preadv

`./iozone -a -i 0 -r 32:` define el tamaño del fichero de registro del test

Tercera Parte. Leer las instrucciones antes de comenzar con la ejecución de las sentencias.

Iperf3 es una utilidad de benchmarking de red para TCP y UDP.

Instala iPerf3 (www.iperf.fr):

```
$ wget https://github.com/esnet/iperf/archive/3.1.2.tar.gz
$ tar xzf 3.1.2.tar.gz
$ cd iperf-3.1.2/
$ ./configure
$ make
```

Consulta la página de manual de iperf3:

```
$ man src/iperf3.1
```

Realiza pruebas de ancho de banda con TCP y UDP entre dos máquinas del laboratorio. Por ejemplo, arranca un servidor TCP:

```
$ src/iperf3 -s
```

Y, desde otra máquina (u otro terminal, para probar), arranca un cliente:

```
$ src/iperf3 -c <dirección IP>
```

Para obtener la dirección IP de la máquina:

```
$ ip addr show
```

O bien:

```
$ /sbin/ifconfig
```

Entrega: Copia los resultados y escribe un breve análisis de los mismos.

iperf es una herramienta sencilla que nos permite medir parámetros relacionados con el ancho de banda de nuestra red.

Al crear la conexión entre un servidor y un cliente observamos los siguientes resultados:

src/iperf3 -c 147.96.82.139

Connecting to host 147.96.82.139, port 5201

[4] local 147.96.82.139 port 51834 connected to 147.96.82.139 port 5201

[ID]	Interval	Transfer	Bandwidth	Retr	Cwnd
[4]	0.00-1.00	sec 8.68 GBytes	74.5 Gbits/sec	0	1.25 MBytes
[4]	1.00-2.00	sec 8.56 GBytes	73.5 Gbits/sec	0	1.25 MBytes
[4]	2.00-3.00	sec 8.76 GBytes	75.3 Gbits/sec	0	1.25 MBytes
[4]	3.00-4.00	sec 8.61 GBytes	74.0 Gbits/sec	0	1.31 MBytes
[4]	4.00-5.00	sec 8.57 GBytes	73.6 Gbits/sec	0	1.50 MBytes
[4]	5.00-6.00	sec 8.68 GBytes	74.6 Gbits/sec	0	1.56 MBytes
[4]	6.00-7.00	sec 8.71 GBytes	74.8 Gbits/sec	0	1.56 MBytes
[4]	7.00-8.00	sec 8.79 GBytes	75.5 Gbits/sec	0	1.56 MBytes
[4]	8.00-9.00	sec 8.73 GBytes	75.0 Gbits/sec	0	1.56 MBytes
[4]	9.00-10.00	sec 8.70 GBytes	74.7 Gbits/sec	0	1.56 MBytes

[ID]	Interval	Transfer	Bandwidth	Retr	
[4]	0.00-10.00	sec 86.8 GBytes	74.6 Gbits/sec	0	sender
[4]	0.00-10.00	sec 86.8 GBytes	74.5 Gbits/sec		receiver

iperf Done.

src/iperf3 -s

Server listening on 5201

Accepted connection from 147.96.82.139, port 51832

[5] local 147.96.82.139 port 5201 connected to 147.96.82.139 port 51834

[ID]	Interval	Transfer	Bandwidth
[5]	0.00-1.00	sec 8.34 GBytes	71.6 Gbits/sec
[5]	1.00-2.00	sec 8.55 GBytes	73.5 Gbits/sec
[5]	2.00-3.00	sec 8.76 GBytes	75.2 Gbits/sec
[5]	3.00-4.00	sec 8.61 GBytes	73.9 Gbits/sec
[5]	4.00-5.00	sec 8.57 GBytes	73.6 Gbits/sec
[5]	5.00-6.00	sec 8.69 GBytes	74.6 Gbits/sec
[5]	6.00-7.00	sec 8.70 GBytes	74.7 Gbits/sec
[5]	7.00-8.00	sec 8.80 GBytes	75.6 Gbits/sec
[5]	8.00-9.00	sec 8.72 GBytes	74.9 Gbits/sec
[5]	9.00-10.00	sec 8.73 GBytes	75.0 Gbits/sec
[5]	10.00-10.04	sec 325 MBytes	68.2 Gbits/sec

[ID]	Interval	Transfer	Bandwidth	
[5]	0.00-10.04	sec 0.00 Bytes	0.00 bits/sec	sender

[5] 0.00-10.04 sec 86.8 GBytes 74.3 Gbits/sec receiver

Server listening on 5201

En los resultados podemos ver que la tasa de transferencia entre terminales es muy alta (Transfer = 8.34 Gbytes, Bandwidth=71.6 Gbits/sec). Creemos que estas tasas son tan elevadas porque la conexión cliente-servidor se realiza en el mismo ordenador, usando dos terminales.