



## **Trabajo Práctico N° 5.3 – Taller de Laboratorio:**

### ***“Evaluación del Rendimiento de SI mediante Benchmarking y Monitoreo”***

**Objetivos:** Que el alumno logre:

- Aplicar las técnicas y métodos de monitoreo del rendimiento de SI/TI.
- Identificar las diferencias entre el benchmarking y el monitoreo de SW y HW.
- Desarrollar la habilidad práctica del proceso de benchmarking y monitoreo de sistemas y redes de computadoras.
- Conocer herramientas actuales para benchmarking y monitoreo de sistemas, tanto para plataformas propietarias como libres.

#### **Referencia Temática:**

- **Unidad 5: La Administración de Recursos de SI/TI – Subst. Seguimiento de SW y HW**
  - *Evaluación del Rendimiento de un Sistema Informático:*
    - Técnicas y Herramientas de Benchmarking y Monitoreo.
    - Magnitudes y Variables
    - Tipos de Carga.
    - Representación de mediciones

**Consultas extra-áulicas:** **A coordinar con la Cátedra (15' por grupo).**

**Fechas de Exposición Grupal y Conclusión:** **Miércoles 30/09/2020** → **Cronograma a definir.**

**Modalidad de desarrollo:** Grupal ▪ **PARTE 1:** En horario extra-áulico → **Subir al CVG el día previo a Expo.**  
▪ **PARTE 2:** En horario extra-áulico, posterior al Taller.

#### **Consignas:**

**NOTA:** Para el desarrollo del Trabajo Práctico y a los efectos de lograr los objetivos de aprendizaje propuestos, se debe respetar el orden de las consignas.

### **PARTE 1 → GRUPAL – Equipos Formales, previo al Taller**

1. **Investigar y probar** las herramientas que se indican a continuación (en el CVG disponen de una lista de posibles fuentes):

a) Comandos y/o archivos que brindan información del sistema:

# mount	# free	# meminfo	# fdisk	# df
# lscpu	# ps	# pstree		

b) Aplicaciones sencillas de Benchmarking y Monitoreo provistas por GNU/Linux:

#top	# htop	# nttcp	# iptraf
------	--------	---------	----------

c) Herramientas libres de Benchmarking y Monitoreo (Ver guías adjuntas como **Anexo 1** y **Anexo 2**):

<b>Sysbench</b>	<b>Apache Benchmark</b>	<b>Nagios</b>	<b>MRTG</b>
-----------------	-------------------------	---------------	-------------

2. **Interpretar los resultados de cada una de las pruebas realizadas, según las siguientes pautas:**

- i) Tipo de herramienta utilizada (según técnica implementada)
- ii) Magnitudes que mide u observa
- iii) Variables utilizadas para medir las magnitudes
- iv) Técnica de activación / Tipo de carga (según corresponda)
- v) Tipos de resultados obtenidos: tipo de variables, forma de presentación, ¿qué indican o muestran?
- vi) Volcar los resultados obtenidos por la herramienta en una planilla de cálculo, generar la gráfica más apropiada e interpretarla.
- vii) Como Administrador de Recursos: ¿qué decisión podría tomar a partir de lo concluido en la consigna anterior)?

## PARTE 2 → Posterior al Taller, en horario extra-áulico

### 3. Resolver el problema planteado a continuación, según las consignas que se indican:

Tras ejecutar la orden `top` en un sistema informático monoprocesador que utiliza Linux como sistema operativo, se ha obtenido la información de actividad que se muestra en la imagen siguiente. Basándose en tal información, se pide:

- ¿Cuánta memoria física tiene este sistema?
- ¿Qué porcentaje de memoria física está usada actualmente?
- ¿Cuál es la utilización media del procesador?
- ¿Cuál es la carga media a lo largo de los últimos 15 minutos?
- ¿Cómo es la evolución de la carga medida del sistema, ascendente o descendente?
- ¿Hay algún proceso en ejecución con baja prioridad? ¿Cuál?
- ¿Hay algún proceso residente en disco debido a intercambio de memoria (swapped out)? ¿Cuál?

```
1:27pm up 1 day, 1:11, 3 users, load average: 2.46, 0.80, 0.28
53 processes: 48 sleeping, 5 running, 0 zombie, 0 stopped
CPU states: 82.5% user, 0.5% system, 17.0% nice, 0.0% idle
Mem: 256464K av, 251672K used, 4792K free, 0K shrd, 22792K buff
Swap: 136512K av, 1956K used, 134556K free, 67484K cached
```

PID	USER	PRI	NI	SIZE	RSS	SHARE	STAT	LC	%CPU	%MEM	TIME	COMMAND
6221	pau	0	0	600	600	484	R	0	27.5	0.2	0:43	sieve
6223	pau	0	0	596	596	480	R	0	27.3	0.2	0:28	qnap
6224	pau	0	0	596	596	480	R	0	27.3	0.2	0:26	qnap
6230	pau	0	10	584	562	468	R N	0	17.0	0.2	1:64	trilog
6231	pau	19	0	980	980	776	R	0	0.3	0.3	0:00	top
2715	root	20	0	23176	6212	2064	S	0	0.1	2.4	0:01	X
1	root	20	0	84	64	44	S	0	0.0	0.0	0:03	init
2	root	20	0	0	0	0	SW	0	0.0	0.0	0:00	keventd
3	root	20	0	0	0	0	SW	0	0.0	0.0	0:00	kapmd
4	root	20	19	0	0	0	SWN	0	0.0	0.0	0:00	ksoftiq
5	root	20	0	0	0	0	SW	0	0.0	0.0	0:11	kswapd
6	root	2	0	0	0	0	SW	0	0.0	0.0	0:00	bdfldsh
7	root	20	0	0	0	0	SW	0	0.0	0.0	0:00	kupdatd
8	root	19	0	0	0	0	SW	0	0.0	0.0	0:01	kinoded
366	root	20	0	600	596	488	S	0	0.0	0.2	0:00	syslogd
369	root	20	0	972	964	360	S	0	0.0	0.3	0:00	klogd
405	root	20	0	0	0	0	SW	0	0.0	0.0	0:00	khubd
487	bin	20	0	332	316	244	S	0	0.0	0.1	0:00	portmap
530	at	20	0	340	312	252	S	0	0.0	0.1	0:00	atd
554	root	14	0	900	816	684	S	0	0.0	0.3	0:02	sshd

### 4. Gráficos de Kiviati:

- Instalar y analizar la herramienta **Systemometer**<sup>1</sup>, según las pautas dadas en la consigna 2, ítems i) a v).
- Construir y analizar el diagrama de Kiviati correspondiente a un sistema con tres recursos (CPU, canal de E/S y enlace de red), para el cual se han obtenido los siguientes datos de las utilidades para los 8 estados posibles (1= Recurso Activo; 0= Recurso Inactivo):

<sup>1</sup> **Systemometer** : <https://systemometer-personal-edition.uptodown.com/windows> (Última visita: 14/8/20).

CPU	Canal de E/S	Enlace de Red	Tiempo Utilizado (%)
1	1	1	10
1	1	0	15
1	0	1	5
1	0	0	10
0	1	1	15
0	1	0	20
0	0	1	5
0	0	0	20

¿Qué puede concluir luego de analizar el gráfico resultante?

- iii) Volcar en una planilla de cálculo algunas de las variables monitoreadas en la consigna 3 y generar un **gráfico radial de Kiviat** (en la pág. 61 y sgtes. de la bibliografía del autor Puigjaner se sugiere una serie de pasos a seguir para construirlo). ¿Qué interpretaciones puede realizar a partir de analizar el gráfico resultante?

5. Luego de haber ejecutado y analizado las pruebas propuestas en el Anexo 1 para el “**Test de I/O**” con la herramienta **SysBench**, suponga que una ejecución de tales pruebas arroja como resultado lo siguiente:

```
Operations performed: 29580 Read, 19720 Write, 62984 Other = 112284 Total
Read 462.19Mb Written 308.12Mb Total transferred 770.31Mb (2.5676Mb/sec)
164.32 Requests/sec executed
```

Test execution summary:

```
total time: 300.0167s
total number of events: 49300
total time taken by event execution: 279.2442
per-request statistics:
  min: 0.01ms
  avg: 5.66ms
  max: 302.36ms
  approx. 95 percentile: 15.31ms
```

¿Cómo interpretaría los resultados obtenidos, relacionándolos con las **variables operacionales** estudiadas?

Por ej.:Cuál sería la **Productividad del disco (Xd)**?Cuál sería su **Tiempo Medio de Servicio (Sd)**?Cuál sería su **Utilización (Ud)**? Qué **otras variables** se pueden deducir?

Justifique analíticamente sus respuestas, indicando los cálculos realizados en cada caso.

### Bibliografía y Fuentes de Referencia:

- PUIGJANER, Ramón y otros. “**Evaluación y Explotación de Sistemas Informáticos**”. Editorial Síntesis. España. 1995.
- MOLERO, Xavier; JUIZ, Carlos; RODEÑO, M. J. “**Evaluación y Modelado del Rendimiento de los Sistemas Informáticos**”. Pearson Educación S.A. Madrid (España). 2004.
- DOMÍNGUEZ DORADO, Manuel; ZARANDIETA MORÁN, José Antonio. “**Evaluación de Nagios para Linux**”. Trabajo de Investigación - Escuela Politécnica de Cáceres - Universidad de Extremadura. España. 2003. - [http://nagios.sourceforge.net/download/contrib/documentation/misc/Nagios\\_spanish.pdf](http://nagios.sourceforge.net/download/contrib/documentation/misc/Nagios_spanish.pdf)
- Mc VOY, Larry; STAELIN, Carl. “**LMBench**” - <http://www.bitmover.com/lmbench/>
- TERCEIRO, Javier. “**LINUX BENCHMARK I**”. 23/4/2013 - <http://www.elmundoenbits.com/2013/04/benchmark-I.html#.WblCBfOLTDe> (Última visita: 14/8/2020)
- TERCEIRO, Javier. “**LINUX BENCHMARK III**”. 07/05/2013 - <http://www.elmundoenbits.com/2013/05/benchmark-III.html#.WblLmvOLTDe> (Última visita: 14-8-2020)

## Trabajo Práctico N° 5.3 – Taller de Laboratorio:

### *“Evaluación del Rendimiento de SI mediante Benchmarking y Monitoreo”*

#### ANEXO 1: Testeo de CPU y Dispositivos de E/S

A continuación, realizaremos una serie de **pruebas de estrés a sistemas GNU/Linux**. Pero ¿Por qué realizamos este tipo de pruebas en nuestros servidores? Pues bien, un servidor al que acceda un número bastante elevado de usuarios, tendrá que tener un buen rendimiento para soportar esa carga de trabajo, en caso contrario, los usuarios experimentarán tiempos ociosos o paradas molestas. Luego, dependiendo de qué uso se le dé al equipo, será mejor idea tener un buen rendimiento en CPU, en I/O, en red, etc.

Para esto, vamos a utilizar inicialmente el software **SysBench**, que permite realizar diferentes tipos de test sobre la máquina para conocer su rendimiento en cuanto a cálculos en CPU, I/O, accesos a memoria, etc.

Para instalarlo, ejecutamos el siguiente comando:

```
shell> sudo apt-get install sysbench
```

Una vez instalado tendremos ya a nuestra disposición el comando con el que podremos realizar los tests.

#### Test de CPU

Con este comando podremos testear el rendimiento de la CPU de nuestro equipo, obligándolo a realizar algunos cálculos que se consideran costosos, en términos de recursos de HW y mostrándonos el tiempo final que lleva la operación. Esto es especialmente útil en máquinas virtuales o compartidas, en las que no sabemos el chip que tenemos debajo ni cuánta prioridad nos dio el administrador.

```
shell> sysbench --test=cpu --num-threads=1 --cpu-max-prime=20000 run
```

Como vemos, se le indica el proceso a ejecutar y también el número de hilos que nos interesa que emplee. Esto nos da un resultado final que es el que tendremos que comparar con los parámetros de rendimiento que hayamos definido como “aceptables”.

#### **ACTIVIDAD PARA LOS ALUMNOS:**

1. Ejecutar las mismas pruebas en diferentes máquinas, registrando la configuración de cada una.
2. Luego, registrar los resultados obtenidos en cada caso (capturando pantallas, por ejemplo), interpretarlos y compararlos.

#### Test de I/O

Este test permite evaluar la velocidad de transferencia de I/O de los discos, algo que suele definir el rendimiento final de un equipo, la entrada/salida. Aquí existen muchas opciones que se le pueden definir para realizar las comprobaciones, pero lo principal es preparar los archivos con los que se operará. Se recomienda emplear archivos que dupliquen por lo menos la cantidad de RAM disponible en el sistema. Si esto no es posible, habrá que indicarle que no emplee memoria caché, para evitar obtener resultados no reales.

Comenzamos por lo tanto creando el entorno de trabajo (los archivos):

```
shell> sysbench --test=fileio --file-total-size=10G prepare
```

Una vez creados, sólo queda ejecutar el test y esperar los resultados.

```
shell> sysbench --test=fileio --file-total-size=10G \
--file-test-mode=rndrw --init-rng=on \
--max-time=300 --max-requests=0 run
```

Para finalizar, borramos todo el espacio de disco que hemos ocupado, pues ya tenemos los resultados deseados.

```
shell> sysbench --test=fileio --file-total-size=10G cleanup
```

#### **ACTIVIDAD PARA LOS ALUMNOS:**

1. Ejecutar las mismas pruebas en diferentes máquinas, registrando la configuración de cada una.
2. Luego, registrar los resultados obtenidos en cada caso, interpretarlos y compararlos.

#### Fuentes:

- TERCEIRO, Javier. “**LINUX BENCHMARK I**” - <http://www.elmundoenbits.com/2013/04/benchmark-I.html#.WblCBfOLTDe> (Última visita: 14-8-2020).
- Página “**man**” de SysBench.

## Trabajo Práctico N° 5.3 – Taller de Laboratorio:

### *“Evaluación del Rendimiento de SI mediante Benchmarking y Monitoreo”*

#### ANEXO 2: Rendimiento de un servidor web

Ahora veremos cómo realizar pruebas de rendimiento sobre un servidor web Apache que contenga una página web. Para ello, vamos a emplear el comando **ab** (**A**pache **B**enchmark **t**ool), que se instala automáticamente al instalar el paquete *apache2-utils* en nuestro sistema.

**ab** permite simular una solicitud de conexión a una web, pero también simular la conexión de un cierto número de clientes, así como la concurrencia de varios threads simultáneamente. De esta forma podemos comprobar, de manera aproximada, el rendimiento que tendría nuestra web con una carga de trabajo real.

#### Instalación

Como ya se mencionó, únicamente hay que instalar el paquete *apache2-utils*,

```
shell> apt-get install apache2-utils
```

#### Uso

**ab** únicamente requiere como argumento la URL que deseamos examinar, pero también permite el paso de parámetros, lo cual facilitará enormemente el chequeo del sistema. En nuestro caso vamos a indicarle dos parámetros, el primero de ellos, opción -n, indica el número de conexiones que se van a realizar. El segundo, opción -c, indica la concurrencia de las conexiones, es decir, el número de conexiones simultáneas a la web. Por ejemplo:

```
shell> ab -n 1000 -c 25 http://www.tu_URL.com/
```

Tras la ejecución del comando, nos presentará los resultados. Con los resultados delante, sólo queda interpretarlos.

Lo primero de todo, dejar claro que **ab** únicamente se ejecuta sobre la URL especificada, en este caso sobre */index.html* y no sobre toda la web. Esto hay que tenerlo en cuenta, pues no es lo mismo un *html* estático que un *html* que lo genera un *php* con mucha información y acceso a base de datos.

También tener en cuenta que todos los complementos adicionales que no sean *html*, léase *javascript*, *css*, *imágenes*, etc. no entran en la carga, ni por lo tanto en la comprobación de rendimiento. Aun así, *Apache Benchmark* es una muy buena herramienta para la realización de test a servidores web.

#### Interpretación

Aunque prácticamente toda la información mostrada es importante, cabe destacar por ejemplo:

- el tiempo que tarda en ejecutarse el test (Time taken for tests),
- la cantidad de datos transmitidos (Total transferred).
- la relación entre el tiempo de conexión (Connection Time)
- el tiempo que tarda en generarse el documento de salida (Connect / Total).

#### ACTIVIDAD PARA LOS ALUMNOS:

1. Ejecutar las mismas pruebas en diferentes servidores y/o con diferentes páginas, registrando la configuración de cada uno.
2. Luego, registrar e interpretar los resultados obtenidos en cada caso y compararlos.
3. Qué utilidad concreta le podría dar a los resultados obtenidos? Dar algunos ejemplos.

#### Fuentes:

- TERCEIRO, Javier. “**LINUX BENCHMARK III**” - <http://www.elmundoenbits.com/2013/05/benchmark-III.html#.WbILmvOLTDC> (Última visita: 14-8-2020)
- Página “**man**” de ApacheBenchmark.