

Organización y Arquitectura de Computadoras

2020-1

Práctica 1: Medidas de Desempeño

Profesor: José de Jesús Galaviz Casas *

Límite de entrega: Agosto 19, 2019

1. Objetivos

Generales:

- El alumno aprenderá cómo medir el desempeño de sistemas de cómputo.

Particulares:

Al finalizar la práctica el alumno tendrá la capacidad de:

- Ejecutar pruebas de desempeño con la batería de pruebas *phoronix* [**Phoronix**].
- Realizar estudios comparativos del desempeño de sistemas de cómputo.
- Determinar la medida de tendencia central adecuada a la muestra tomada para sintetizar el desempeño.

2. Requisitos

■ Conocimientos previos:

- El sistema operativo *GNU/Linux* y el manejo de comandos en una terminal.
- Conceptos básicos de medidas de desempeño: tiempo de respuesta, rendimiento y pruebas de desempeño, consultar [**Jain**].
- Las propiedades básicas de las siguientes medidas de tendencia central:
 - Media aritmética.

*Diseñada por Roberto Monroy Argumedo

- Media aritmética ponderada.
- Media armónica.
- Media geométrica.

■ **Tiempo de realización sugerido:**

5 horas.

■ **Número de colaboradores:**

Equipos de 4 integrantes.

■ **Software a utilizar:**

- Una computadora personal con sistema operativo *GNU/Linux*.
- El paquete *phoronix-test-suite*, el cual puede ser encontrado en los repositorios de las principales distribuciones de Linux o es posible descargarlo de forma gratuita directamente del sitio web [**Phoronix**].

3. Planteamiento

El diseñador de arquitecturas de cómputo debe tomar varias decisiones que tienen impacto directo tanto en el costo como en el desempeño del sistema, por lo que requiere de mecanismos eficaces para medir y comparar el rendimiento de los sistemas de cómputo. La forma de llevar a cabo el análisis es a través de la ejecución de baterías de pruebas. Nuestra tarea será ejecutar algunas pruebas representativas para conocer el rendimiento de nuestra computadora personal y compararlo con el de otros equipos de cómputo.

4. Desarrollo

4.1. Phoronix-test-suite

Phoronix-test-suite es una batería de pruebas de código abierto desarrollada por *Phoronix.com* bajo la licencia *GNU GPL*. Algunas de sus características que la hacen útil para nuestro propósito son:

- Se encuentra disponible para distintos sistemas operativos.
- Arquitectura extendible.
- Más de 450 perfiles de pruebas.
- Más de 150 baterías de pruebas.
- La descarga e instalación de pruebas es automática.
- Detección detallada de software y hardware.

4.2. Instalación

La plataforma se encuentra disponible en los repositorios de las principales distribuciones de Linux, instala el paquete a través del manejador de paquetes de tu distribución. En otro caso, es posible descargar el paquete de la página oficial e instalarlo de forma manual. En caso de requerir alguna dependencia el programa informará cuál paquete es necesario al ejecutarse alguna prueba.

4.3. Uso

El uso de la batería de pruebas es a través de la línea de comandos, los comandos básicos para ejecutar las pruebas son:

- Mostrar todas las pruebas disponibles:
`$ phoronix-test-suite list-tests`
- Para conocer más información acerca de alguna prueba en particular:
`$ phoronix-test-suite info <nombre de la prueba>`
- Para ejecutar una prueba:
`$ phoronix-test-suite benchmark <nombre de la prueba>`

Las pruebas no se incluyen en la instalación del paquete, por lo que la primera vez que se ejecute alguna prueba, los archivos necesarios serán descargados y las dependencias faltantes serán instaladas a través del manejador de paquetes de la distribución.

- Para desintalar una prueba:
`$ phoronix-test-suite remove-installed-test <nombre de la prueba>`

4.4. Pruebas

A continuación, se encuentra una lista y una breve descripción de las pruebas que serán utilizadas en esta práctica. En el sitio web de *Open Benchmarking* [[OpenBenchmarking](#)] se pueden encontrar las pruebas disponibles y resultados enviados por otros usuarios.

■ GZIP Compression

Esta prueba mide el tiempo necesario para comprimir un archivo usando GZIP.

`pts/compress-gzip`

■ DCRAW

Esta prueba mide el tiempo que toma convertir varias imágenes de alta resolución en formato RAW NEF a imágenes en formato PPM usando dcraw.

`pts/dcraw`

- **FLAC Audio Encoding**

Esta prueba mide cuanto tiempo toma codificar un archivo muestra de audio en formato WAV a un archivo en formato FLAC.

`pts/encode-flac`

- **GnuPG**

Esta prueba mide el tiempo que toma encriptar un archivo usando GnuPG.

`pts/gnupg`

- **REDIS**

Redis es un motor de bases de datos en memoria de código abierto, la prueba mide el número de solicitudes que el servidor puede responder por segundo.

`pts/redis`

- **Timed MAFFT Alignment**

Esta prueba mide el tiempo que le toma alinear 100 secuencias de piruvato descarboxilasa.

`pts/mafft`

- **Timed MrBayes Analysis**

Esta prueba mide el tiempo que le toma realizar un análisis bayesiano de un conjunto de secuencias del genoma de primates con el fin de estimar su filogenia.

`pts/mrbayes`

- **Timed MPlayer Compilation**

Esta prueba mide el tiempo que toma compilar el reproductor MPlayer.

`pts/build-mplayer`

- **Timed PHP Compilation**

Esta prueba mide el tiempo que toma complilar PHP 5.

`pts/build-php`

5. Entrada

No hay datos de entrada para esta práctica.

6. Salida

Cada miembro del equipo deberá entregar un reporte con la información requerida en las secciones de ejercicios y preguntas.

7. Variables libres

En el ejercicio 4 el alumno deberá plantear un caso de uso para una computadora, es decir, deberá pensar en las necesidad de un usuario en particular para decidir cuál equipo es el adecuado después de haber realizado el estudio.

8. Procedimiento

1. En equipo, elaboren una forma para reportar los resultados de las pruebas.
La forma deberá solicitar la siguiente información:
 - Nombre del alumno.
 - Datos de la computadora:
 - Fabricante y modelo de la computadora.
 - Fabricante, modelo, frecuencia, número de núcleos y arquitectura del procesador.
 - Capacidad de memoria RAM y de cachés de los procesadores.
 - Capacidad, tipo y velocidad del disco duro.
 - Distribución de linux y versión del kernel.
 - Una tabla con dos columnas para reportar los resultados, la primer columna será el nombre de la prueba y la segunda el resultado indicando la unidad de medida.
2. Cada miembro del equipo deberá ejecutar las pruebas en una computadora y reportar los resultados en una forma. Las computadoras deberán diferir en al menos una característica (sistema operativo, procesador, capacidad de memoria RAM, etc.).

9. Ejercicios

1. Identifica cuales de las pruebas miden el tiempo de respuesta y cuales miden el rendimiento.
2. Usando la medida de tendencia central adecuada y tu reporte de resultados, calcula:
 - La medida de tiempo de respuesta.
 - La medida de rendimiento.

Agrega los resultados obtenidos a tu reporte y compártelo con tus compañeros de equipo.

3. Una vez que tengas los reportes de tus compañeros, fija tu computadora como computadora de referencia. Calcula los tiempos normalizados y obtén la medida de tendencia central adecuada de cada una de las computadoras. Agrega los resultados obtenidos a tu reporte.

10. Preguntas

- a) ¿Cuál computadora tiene el mejor tiempo de ejecución? Comparada con la computadora con la peor medida de tiempo de ejecución ¿por qué factor es mejor la computadora? Enuncia el resultado de la forma “El tiempo de ejecución de la computadora A es x veces _____ que la computadora B ”.
- b) ¿Cuál computadora tiene el mejor rendimiento? Comparada con la computadora con el peor desempeño ¿por qué factor es mejor la computadora? Enuncia el resultado de la forma “El rendimiento de la computadora A es x veces _____ que la computadora B ”.
- c) De acuerdo a la computadora de referencia, ¿cuál computadora tiene el mejor desempeño y cuál computadora tiene el peor desempeño?
- d) De entre los atributos de cada máquina comparada, ¿cuáles resultan determinantes en la pérdida o ganancia de desempeño?