

# Comparativa de rendimiento: VirtualBox y VMware

# Índice

<b>1. Introducción</b>	<b>3</b>
1.1. Motivación . . . . .	3
1.2. Objetivo . . . . .	3
<b>2. Método</b>	<b>3</b>
2.1. Variables . . . . .	3
2.2. Plan de recogida . . . . .	3
2.3. Material . . . . .	4
2.4. Plan estadístico . . . . .	4
<b>3. Resultados</b>	<b>5</b>
<b>4. Discusión</b>	<b>7</b>
4.1. Conclusiones . . . . .	7
4.2. Limitaciones y trabajos futuros . . . . .	8
<b>5. Anexo</b>	<b>9</b>

## 1. Introducción

### 1.1. Motivación

Debido a la frecuente necesidad en el ámbito de la informática de hacer uso de múltiples sistemas operativos en una misma máquina, a veces es inviable tenerlos instalados todos, ya sea por: incompatibilidad, versiones preliminares que pueden afectar los otros SO, etc. Este problema se puede solucionar emulando los diferentes sistemas operativos que queremos ejecutar en un "host", el sistema de la máquina, que se encargará a través de programas de emular los distintos SO que deseemos.

### 1.2. Objetivo

El objetivo consiste en evaluar cuál de los dos programas más famosos en la virtualización, VirtualBox y VMware, tiene mayor rendimiento. Para eso recopilaremos los tiempos que tardan cada uno en ejecutar un "benchmark". En nuestro caso, el sistema host será Windows 10 y el SO a emular será Ubuntu, además, el benchmark consistirá en un pequeño programa escrito en C++ que calcula Fibonacci(43).

## 2. Método

### 2.1. Variables

Nuestras variables a estudiar son:

- "VirtualBox, VMware", variable explicativa.
- "Tiempo de ejecución, en segundos, de Fibonacci(43)", variable respuesta.

### 2.2. Plan de recogida

Para realizar la obtención de datos de forma que estos sean lo más independiente posible, es necesario controlar el efecto que podría causar el hecho de obtener los valores de forma consecutiva.

Este hecho lo solventaremos recogiendo las muestras a partir de un sistema Windows recién inicializado y ejecutando la máquina virtual cada vez que vayamos a

calcular el tiempo del benchmark (obtenido con la función *time* que ofrecen los sistemas Linux), posteriormente se apagará la máquina virtual para proceder a la obtención del siguiente dato.

Este proceso se llevará a cabo en el mismo ordenador.

### 2.3. Material

El material utilizado es el siguiente:

- Ordenador: i7-3770 @ 3.4 GHz
- SO: Windows 10 Pro
- SO Virtualizado: Ubuntu 16.04.1 64 bits, 8GB HDD, 1GB RAM, 1 core
- R x64 v3.0.1
- Benchmark fibonacci(43), código proporcionado en el anexo
- VirtualBox v5.0.20
- VMware v12.5.2

### 2.4. Plan estadístico

$B :=$  Tiempo ejecución VirtualBox

$W :=$  Tiempo ejecución VMware

Pruebas de hipótesis:

$$\begin{cases} H_0 : \mu_B = \mu_W \Leftrightarrow \mu_d = \mu_B - \mu_W = 0 \\ H_1 : \mu_B \neq \mu_W \end{cases}$$

Premisas:

- Variable con distribución normal
- Muestra independiente

- Prueba bilateral debido a la incertidumbre de que opción responde mejor
- Igualdad de varianzas

Estadístico:

La varianza "pooled":

$$s = \sqrt{\frac{(n_B - 1)s_B^2 + (n_W - 1)s_W^2}{(n_B - 1) + (n_W - 1)}}$$

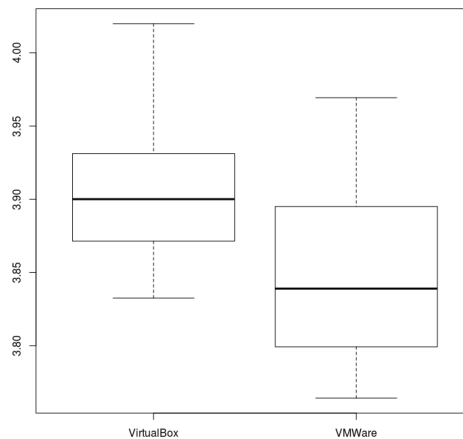
El estadístico para comparación de medias:

$$\hat{t} = \frac{\bar{B} - \bar{W}}{s\sqrt{\frac{1}{n_B} + \frac{1}{n_W}}} \sim t_{n_B+n_W-2}$$

### 3. Resultados

Una vez realizado el plan de recogida de 20 datos de cada programa, el summary y el boxplot conjunto es:

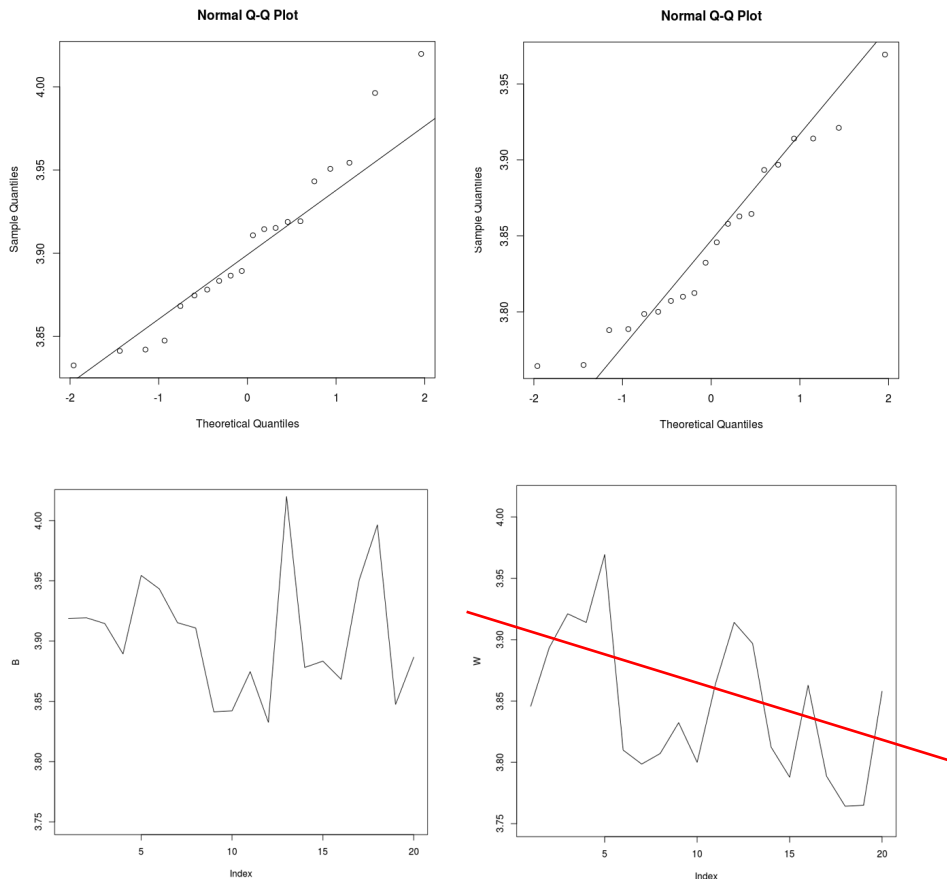
```
summary(B); sd(B);
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 3.833  3.873   3.900   3.904  3.925   4.020
[1] 0.05065868
summary(W); sd(W);
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 3.764  3.800   3.839   3.845  3.894   3.969
[1] 0.05806888
```



En primer lugar observamos como no se aprecia una diferencia significativa entre varianzas.

En cuanto al boxplot, se puede apreciar un desplazamiento hacia valores más altos en VirtualBox y una simetria de las dos gráficas respecto la mediana, lo cual nos indica: en primer lugar, que efectivamente las medias no son iguales y, en segundo lugar, que las variables siguen una distribución normal (aspecto que comentaremos proximately con más detalle).

Procedemos a comprobar la ausencia de patrones en los datos, haciendo uso de *plot*, y la comprobación de normalidad, a través de la *qqnorm* (instrucciones de R), en la izquierda VirtualBox y en la derecha VMware:



Observamos como los datos siguen de forma bastante precisa una distribución normal, a excepción de algunos valores puntuales que asociamos a la poca canti-

dad de datos recopilados (recordemos que son 20 valores de cada programa). En cuanto a la independencia, no se observa ningún patrón que pueda indicarnos lo contrario.

Una vez comprobadas las premisas, pasamos a realizar el estudio de los datos a través de la prueba de t de Student.

```
t.test(B,W,var.equal=TRUE)
```

```
Two Sample t-test
```

```
data: B and W
t = 3.4267, df = 38, p-value = 0.001481
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.02416415 0.09392965
sample estimates:
mean of x mean of y
 3.904352  3.845305
```

## 4. Discusión

### 4.1. Conclusiones

Partiendo de la hipótesis nula de igualdad de medias hemos obtenido el siguiente valor de estadístico:

$$\hat{t} = 3,4267$$

Teniendo en cuenta que la región de aceptación con una confianza del 95 % y el p-valor comparado con alfa son:

$$\text{Región aceptación} = [-2,024394; 2,024394]$$

$$p_{\text{valor}} = 0,001481 < 0,05 = \alpha$$

Podemos afirmar que las medias de los diferentes programas de virtualización tienen tiempos de ejecución diferentes.

Además, tenemos que el intervalo de confianza del 95 % es:

$$IC(\mu_d; 0,95) = [0,02416415; 0,09392965]$$

De donde se extrae que VirtualBox tarda entre 0,02 y 0,09 segundos más que VMware en ejecutar el benchmark.

## 4.2. Limitaciones y trabajos futuros

Debido a que el estudio se ha realizado en un ordenador, este puede influir negativamente en los datos, ya sea porque de repente un proceso en background adquiere más CPU o debido al fenómeno del caching, estos efectos los hemos intentado evitar planteando un proceso de recogida de datos con esto en mente, pero aún así es una dificultad que es casi imposible de evitar en su totalidad.

OK.  
Ahora no sabemos si el orden beneficia a una de las opciones

Como posibles trabajos futuros, una posible forma de completar el estudio consistiría en hacer la selección de la máquina virtual que toca recoger datos usando una secuencia aleatoria de 0 y 1. Además, el estudio se podría realizar en distintos sistemas operativos "host" y añadiendo mayor variedad de programas a estudiar, como podría ser Hyper-V (recientemente añadido a Windows 10).



## 5. Anexo

Tabla 1: Tiempo de ejecución del benchmark (en segundos)

No	VirtualBox	VMware
1	3.918853	3.845734
2	3.919244	3.893379
3	3.914507	3.921099
4	3.889350	3.914007
5	3.954420	3.969342
6	3.943232	3.809956
7	3.915213	3.798601
8	3.910839	3.807196
9	3.841294	3.832344
10	3.842100	3.800051
11	3.874621	3.864478
12	3.832599	3.914060
13	4.019847	3.896824
14	3.878126	3.812387
15	3.883365	3.787942
16	3.868260	3.862821
17	3.950760	3.788646
18	3.996347	3.764294
19	3.847482	3.765021
20	3.886576	3.857915

Código fuente: Fibonacci en C++

```
#include <iostream>
using namespace std;

int fibonacci(int n) {
    if (n == 1) return 0;
    if (n == 2) return 1;
    return fibonacci(n-1) + fibonacci(n-2);
}

int main() { fibonacci(43); }
```