

USB 2.0 VS 3.0

Comparación de velocidades de transferencia entre versiones de USB

Diciembre 2016

1. RESUMEN

Nuestro objetivo en este estudio es comprobar si realmente la versión de USB 3.0 es más rápida que la 2.0 y hacer una aproximación de su ratio de mejora.

Calculamos la velocidad de transferencia de cada uno de los 27 ficheros que hemos escogido según la versión del puerto, desde un mismo PenDrive a un mismo ordenador; obteniendo así una variable diferencia de velocidades.

No se suele usar esta terminología estadística en informes técnicos

Nuestros resultados, con una confianza del 95%, nos conducen a rechazar la hipótesis nula y poder afirmar que claramente la velocidad entre versiones no es la misma, teniendo una clara ventaja a favor de la versión 3.0 con un ratio de 1.71 veces más rápido. Aún así, hemos encontrado resultados aislados en algunos ficheros que podrían producir debilidades en el estudio, haciendo que el resultado varíe del esperado. Además, se dice "un riesgo del 5%", no una confianza del 95% (eso es para intervalos de confianza)

En conclusión, la versión de USB3.0 es más rápida que la 2.0 y convendría empezar a utilizar el nuevo modelo siempre que se pueda.

2. INTRODUCCIÓN

a. Nuestro objetivo

Como estudiantes de informática, independientemente de la rama de especialidad, es sustancial conocer cómo evoluciona la tecnología y la importancia de mejorar, y un buen ejemplo es el de los puertos USB.

El USB 3.0 fue diseñado a finales del 2008 y, según Wikipedia, la versión 3.0 puede llegar a ser 10 veces más rápida que la 2.0.

¿Pero es cierto que es más rápida? ¿Cual es la verdadera diferencia de velocidad en un USB 3.0 estándar? ¿Se ve modificada la velocidad según el peso del archivo?

Para encontrar respuesta a todas esas preguntas, decidimos escoger este tema como proyecto del bloque 7 para responderlas mediante su estudio.

b. Material

Lo primero que necesitamos es un dispositivo compatible con ambas versiones, así que escogemos un PenDrive 3.0 Asus de 16GB con sistema de archivos FAT32.

Seguidamente, un ordenador con ambos puertos para poder realizar el estudio bajo las mismas condiciones. Elegimos un ordenador de la Facultad de los laboratorios de Sistemas Operativos.

Para que no haya ningún tipo de influencia no deseada, limpiamos la caché del ordenador antes de cada copia y formateamos el PenDrive.

Por último, escogemos 27 ficheros de diferentes formatos y pesos, pero de un tamaño suficiente para poder ver una diferencia sustancial de tiempo.

¿Tan pequeños?

Nombre	Peso	Nombre	Peso	Nombre	Peso
archivo.txt	2,1 GB	cloud.exe	675,5 kB	0506.mov	2,5 GB
archivo1.txt	1,8 GB	mont.mp4	32,8 MB	0345.jpg	9,5 MB
archivo2.txt	931,1 MB	x-men.mkv	3,4 GB	0146.jpg	8,7 MB
doc.docx	4,9 kB	ceni.avi	1,6 GB	Arvalis.jpg	2,4 MB
calen.odt	19,7 kB	south.avi	168,7 MB	0067.NEF	20,2 MB
pic.pdf	7,2 MB	0020.mov	1,9 GB	0066.NEF	21,5 MB
music dir	451,6 MB	0081.mov	261,8 MB	12.png	1,9 MB
mrrobot dir	3,3 GB	0390.mov	2,3 GB	inside.mp3	4,7 MB
PS.exe	93,2 MB	0944.mov	109,8 MB	TFRt.mp3	3,9 MB

Figura 2.1. Tabla de ficheros y sus pesos

c. Método

Puesto que para comprobar si la velocidad es dependiente del formato sería necesario otro experimento, en este estudio asumiremos que la velocidad es independiente del formato.

Para realizar el proyecto necesitamos efectuar el estudio con un mismo fichero y examinar su velocidad en los dos tipos. Así pues, tenemos un diseño apareado. Decidimos tener **una variable explicativa, el peso del fichero, y dos variables respuesta que definen el tiempo que tarda en realizarse una copia del fichero desde el PenDrive hasta un directorio del disco duro del ordenador, para más adelante tener la variable diferencia de velocidades.**

Nuestra hipótesis nula sostiene que la velocidad de la versión USB3.0 es igual a la del USB2.0. Si todo procede como se espera, deberemos rechazarla, siendo la **velocidad del nuevo modelo mayor que la del antiguo (contraste unilateral).** razonable

$$H_0 : \mu_3 = \mu_2 \text{ o también } \mu_D = 0$$

$$H_1 : \mu_3 > \mu_2 \text{ o también } \mu_D > 0$$

Cumplimos, de momento, una de las dos premisas necesarias: nuestras variables son **muestras apareadas**. Una vez recogidos los datos, deberemos comprobar la **normalidad** de la variable diferencia.

Nuestro estadístico t sigue una distribución t-student de 26 grados de libertad, y la región crítica con un riesgo del 5% es 1.705618. El intervalo de confianza de 95% se construye:

$$IC(\mu_D, 0,95) = \bar{d} \pm t_{n-1, 1-\alpha} \sqrt{\frac{S_D^2}{n}} \quad \hat{t} = \frac{\bar{D} - \mu_0}{S_D / \sqrt{n}}$$

3. REALIZACIÓN

a. Script

Para poder realizar los cálculos de manera más rápida y eficiente decidimos escribir un script que limpia primeramente todas las cachés (páginas, inodos y entradas de directorios) y luego hace el cálculo del tiempo de la copia de los parámetros que le indicamos. Se ejecuta con el siguiente comando:

```
# ./copia.sh A B
```

donde *A* es la ruta origen (por ejemplo /media/alumne/PenDrive/archivo.txt) y *B* es la ruta destino (/home/alumne/Desktop/).

```
#!/bin/bash
echo "Borrando cachés..."
sudo sh -c "sync && echo 3 > /proc/sys/vm/drop_caches"
echo "Borrado completado."
echo ""
echo "Tiempo al copiar:"
time cp $1 $2
```

Figura 3.1. Código de copia.sh (script)

El comando *time* da tres tipos diferentes de tiempo: el total (*real*), el tiempo que el sistema tarda en realizar una llamada al sistema (*sys*) y el tiempo usuario, el tiempo que la CPU espera al realizar una acción (*user*). Nosotros utilizamos el tiempo real de CPU.

b. Ejecución

Cada fichero lo copiamos al PenDrive **previamente formateado desde otro ordenador**, conectamos el PenDrive al puerto 3.0 del ordenador escogido y ejecutamos el script con la nueva dirección. Mientras tanto no realizamos ninguna acción más con el ordenador para dejarle el máximo de CPU posible. Una vez realizada la copia por la versión 3.0, desconectamos y conectamos el PenDrive al puerto 2.0, y volvemos a ejecutar el script de copia.

4. RESULTADOS

Podemos encontrar en el anexo los gráficos de los resultados para cada versión. En ellos nos encontramos con algunos datos a resaltar, como que en dos ficheros, el tiempo de transferencia es menor con el USB2.0 y, también, que la desviación de la nueva versión es mucho mayor que la anterior.

A partir de ahora, la columna *Nombre* pasará a llamarse *N*, y los nombres de los archivos serán números y estarán ordenados por peso para que sea más visible en los gráficos.

Versión	Media (MB/s)	Se (MB/s)
2.0	21.81928	13.23406
3.0	45.31496	33.12774

Figura 4.1. Velocidad Media y Desviación para cada versión

a. Premisas

Debemos buscar si nuestra variable diferencia cumple la premisa de Normalidad. En los gráficos del anexo *b* nos encontramos con que **no parece seguir** la distribución Normal y por ello transformamos nuestras variables a su **logaritmo**. Ahora obtenemos un gráfico de la variable diferencia que se ajusta mucho más a una Normal.

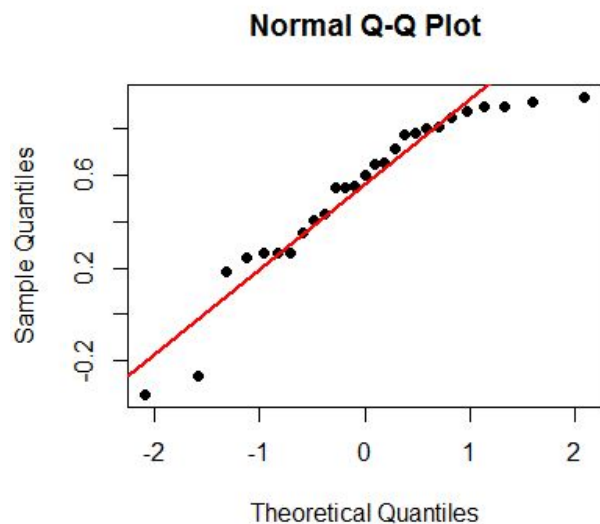
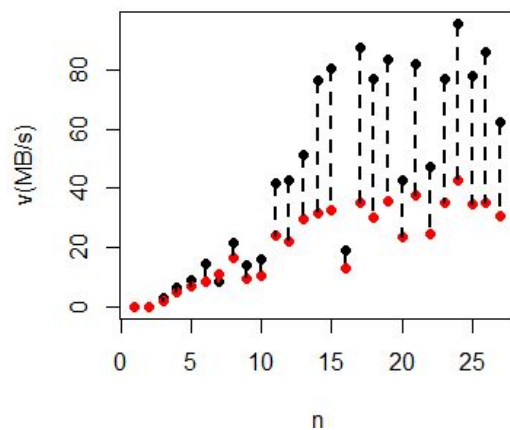
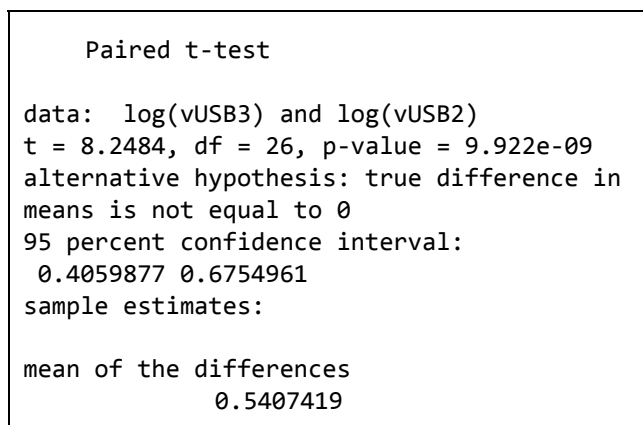


Figura 4.2. qqnorm de la diferencia de logaritmos

b. Hipótesis Nula

Como podemos ver en la figura 4.4 resulta bastante claro que la velocidad del USB3.0 es mayor que la del USB2.0. Hacemos un t.test (figura 4.3) que nos confirma nuestra idea. Con éste observamos que el p-valor es tan pequeño que **rechazamos la hipótesis nula** de que las velocidades son iguales sin ningún problema (el p-valor es muy inferior al riesgo del 5%). Así, siendo nuestra hipótesis alternativa que la velocidad de la nueva versión es mayor que la de la anterior, debemos comprobar ahora cuántas veces es mayor.

este gráfico es poco útil, mejor ratio versus tamaño.



Figuras 4.3 y 4.4. t.test de la diferencia de logaritmos y

gráfico de comparación de velocidades para cada fichero (Negro = 3.0, Rojo=2.0)

c. Ratio

Además del p-valor, también obtenemos que la media de la diferencia de logaritmos es 0.5407419. Tomando este valor con un intervalo de confianza del 95% (0.4059877, 0.6754961):

$$\log(v_{USB3}) - \log(v_{USB2}) = 0.54 \Rightarrow \log\left(\frac{v_{USB3}}{v_{USB2}}\right) = 0.54 \Rightarrow \frac{v_{USB3}}{v_{USB2}} = e^{0.54} = 1.71$$

Es decir, el **USB 3.0 es 1.71 veces más rápido que el USB 2.0.**

acompañar también con el IC: (1.50, 1.97)

5. DISCUSIÓN

Con estos resultados y gráficos observamos que claramente la versión 3.0 tiene más ventaja sobre los ficheros grandes que los pequeños y que convendría utilizarla, ya que los datos cada vez son más pesados. Aun así, el ratio que encontramos no alcanza el mencionado por Wikipedia, ya que como hemos visto el 3.0 es sólo 1.71 veces más rápido, lejos del 10 veces mayor que nos apunta la web. Esta diferencia puede ser dada por algunas limitaciones, entre ellas:

- El PenDrive no fuera óptimo según sus especificaciones.
- El proceso de copiado se pausara a media ejecución.
- La gran diferencia de pesos entre los ficheros.
- Pocos ficheros de gran peso y muchos de poco

Este último hecho podría habernos causado algunos datos erróneos que, al tener pocos ficheros, hizo que tuvieran mucha relevancia y nos desviaron del resultado correcto. **Tampoco hay que flagelarse. Es más fácil pensar que los de wikipedia (o sus referencias) se han pasado.**

Nuestra última pregunta, sobre si la velocidad varía según el peso, se puede responder observando la figura 5.1, que nos muestra una clara diferencia en el USB3.0 según aumentamos el peso. Podemos ver que el USB2.0 sigue una curva de crecimiento sin cambios bruscos (exceptuando algunos pocos valores), mientras que en la versión 3.0 encontramos archivos con los que la diferencia disminuye considerablemente (no sigue la pauta uniforme de su versión anterior).

Esta pregunta se tenía que haber resuelto ajustando un modelo lineal, no "a ojo".

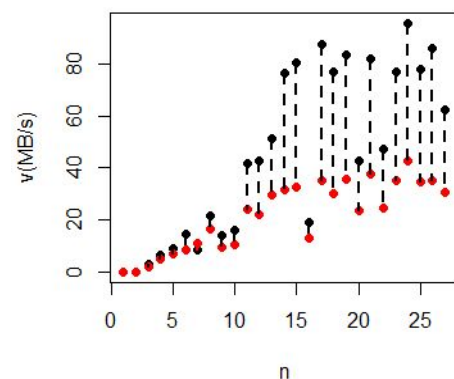


Figura 5.1. gráfico de comparación de velocidades para cada fichero (Negro = 3.0, Rojo=2.0)

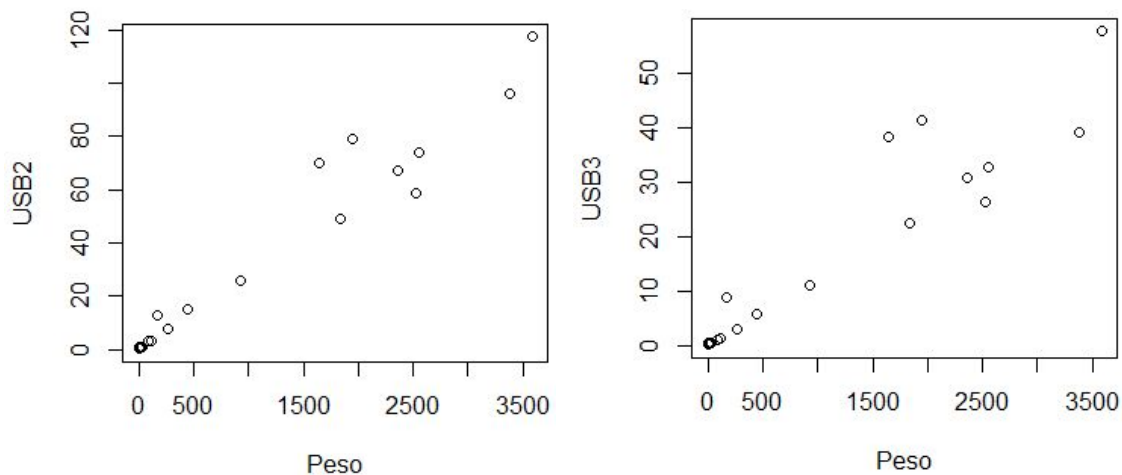
Por último concluimos que, efectivamente, el USB3.0 es mejor que el 2.0. La diferencia se hace más notable cuanto más grande es el fichero; en los ficheros pequeños no llegaríamos a apreciar el cambio como usuarios.

6. ANEXOS

a. Gráficos de los resultados

mejor con escala logarítmica

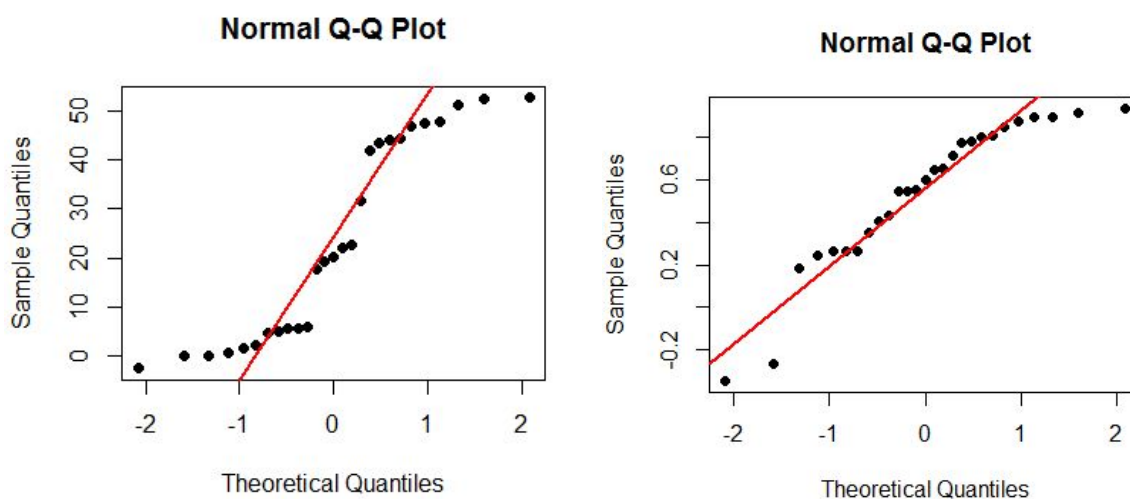
Aquí podemos ver los gráficos de los resultados. Nótese los valores del eje Y.



Figuras 6.1 y 6.2. Gráficos del tiempo de copia (eje Y, en segundos) respecto su peso (eje X, en MB)

b. Premisas

Según podemos ver en el gráfico, la variable diferencia no sigue una distribución normal. Así, al tener la necesidad de utilizar una que siga esta distribución decidimos utilizar la variable diferencia de logaritmos, que claramente muestra un mejor comportamiento que sus otras posibles transformaciones.



Figuras 6.3 y 6.4. qqnorms de la diferencia y diferencia de logaritmos

c. Script R

```
## Lectura de los datos
datos <- read.table("clipboard", header=TRUE, sep='\t')

## Región crítica
qt(0.95,26)

## Gráficos de los resultados
#v2.0
windows()
plot(USB2~Peso, data = datos)
#v3.0
windows()
plot(USB3~Peso, data = datos)

## Variable diferencia
dif <- datos$vUSB3-datos$vUSB2
apply(datos[,5:6],2,mean)
apply(datos[,5:6],2,sd)
# Normalidad
windows()
qqnorm(dif,pch=19)
qqline(dif,lwd=2,col=2)

## Variable diferencia de logaritmos
dif <- log(datos$vUSB3)-log(datos$vUSB2)
# Normalidad
windows()
qqnorm(dif,pch=19)
qqline(dif,lwd=2,col=2)

## Gráfico de comparación de velocidades
windows()
plot(datos$n,datos$vUSB3,pch=19,cex=1,xlab="n",ylab="v(MB/s)")
points(datos$n,datos$vUSB2,pch=19,col=2,cex=1)
for (i in 1:n) {
  if(datos$vUSB3[i]>datos$vUSB2[i]) {co <- 1 ; lt <- 2}
  else {co <- 2 ; lt <- 1}
segments(datos$n[i],datos$vUSB3[i],datos$n[i],datos$vUSB2[i],col=co,lty
=lt,lwd=2 )
}

## t.test
t.test(log(datos$vUSB3),log(datos$vUSB2),paired=TRUE)
```

d. Tabla de resultados

Aquí adjuntamos la tabla que hemos utilizado para el script de R.

¿por qué tantos decimales?

N	Peso (MB)	USB2 (s)	USB3 (s)	vUSB2 (MB/s)	vUSB3 (MB/s)
1	0.00478516	0.193	0.273	0.02479357513	0.01752805861
2	0.01923828	0.266	0.221	0.0723243609	0.08705104072
3	0.65966797	0.3	0.235	2.198893233	2.807097745
4	1.9	0.38	0.291	5	6.529209622
5	2.4	0.341	0.262	7.038123167	9.160305344
6	3.9	0.465	0.27	8.387096774	14.44444444
7	4.7	0.431	0.564	10.90487239	8.333333333
8	7.2	0.432	0.331	16.66666667	21.75226586
9	8.7	0.93	0.618	9.35483871	14.0776699
10	9.5	0.915	0.595	10.38251366	15.96638655
11	20.2	0.834	0.483	24.2206235	41.82194617
12	21.5	0.968	0.506	22.2107438	42.49011858
13	32.8	1.112	0.638	29.49640288	51.41065831
14	93.2	2.933	1.223	31.77633822	76.2060507
15	109.8	3.341	1.366	32.86441185	80.3806735
16	168.7	12.687	8.921	13.29707575	18.91043605
17	261.8	7.453	2.989	35.12679458	87.58782201
18	451.6	14.968	5.86	30.17103153	77.06484642
19	931.1	25.986	11.124	35.83083199	83.70190579
20	1638.4	69.999	38.261	23.40604866	42.8216722
21	1843.2	48.792	22.486	37.7766847	81.97100418
22	1945.6	79.223	41.289	24.55852467	47.12150936
23	2355.2	67.306	30.692	34.99242267	76.73660889
24	2528.4	58.818	26.442	42.98684076	95.62060359
25	2560	73.776	32.811	34.69963132	78.02261437
26	3379.2	96.237	39.219	35.11331401	86.16231928
27	3584	117.267	57.53	30.56273291	62.29793151