



Universidad Rey Juan Carlos

E.T.S. INGENIERÍA DE TELECOMUNICACIÓN

SISTEMAS EMPOTRADOS Y DE TIEMPO REAL

Práctica 2

Autor:
Javier Izquierdo Hernández

October 26, 2023

Análisis con `cyclictest`

cyclictest						
	Laboratorios		RaspberryPi			
	Kernel NO RT		Kernel NO RT		Kernel RT	
	Latencia media (ns)	Latencia max (ns)	Latencia media (ns)	Latencia max (ns)	Latencia media (ns)	Latencia max (ns)
S1	3367	4748929	22499	121158	22039	70004
S2	6575	4792281	26072	390565	14956	83535
S3	6033	4546481	20394	118893	22845	69536

Los casos que más llaman la atención son: S3 en la Raspberry Pi con el kernel no RT y S2 en la Rbpi con el kernel RT.

En el primer caso se observa un decremento significativo en la latencia media comparándolo con la ejecución en *idle* y también desciende en menor medida la latencia máxima, pero esto puede deberse a la escasez de muestras, por lo tanto no vamos a entrar en esto. Este descenso se debe principalmente al mayor rango en el que se encuentran las latencias, causando una mayor aleatoriedad de las latencias, siendo más complicado de predecir. También hay que añadir que posiblemente la latencia máxima pueda variar de manera significativa con cada ejecución.

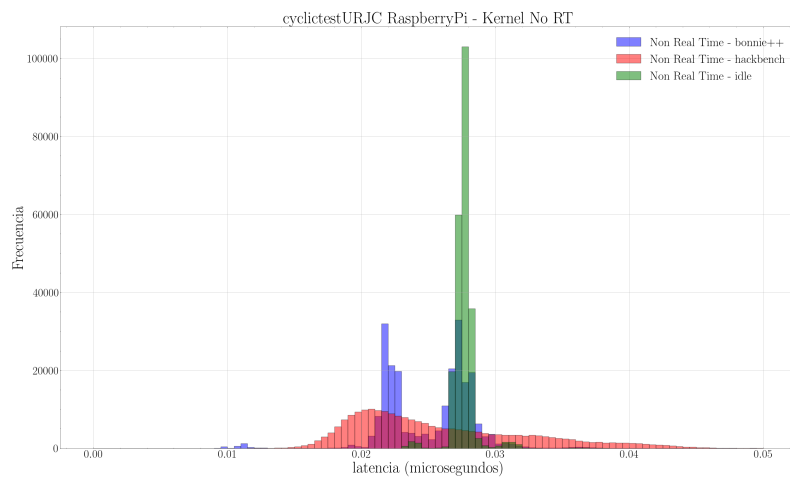
Y en el segundo casi se observa un descenso similar en la latencia media, pero la latencia máxima aumenta. Esto se puede observar mejor en los histogramas, pero se entiende al ver que al ejecutarlo con `hackbench` la mayoría de las latencias disminuyen, pero el dominio en el que se encuentran aumenta, lo que causa una mayor aleatoriedad.

Desarrollo de `cyclictestURJC`

<code>cyclictestURJC</code>						
	Laboratorios		RaspberryPi			
	Kernel NO RT		Kernel NO RT		Kernel RT	
	Latencia media (ns)	Latencia max (ns)	Latencia media (ns)	Latencia max (ns)	Latencia media (ns)	Latencia max (ns)
S1	3597	4589910	27715	94920	25250	94198
S2	5562	4804750	25960	144730	16921	85624
S3	3882	5362156	25212	230656	25932	80771

La tabla es similar a la anterior excepto en la Rbpi con el kernel no RT.

Aquí es la latencia media en *idle* la que ha sufrido un aumento, aunque también es significativamente menor que en el resto de casos. Esto también se puede explicar como lo hemos hecho anteriormente, y se comprueba al ver sus histogramas, que la latencia minima en *idle* esta comprendida en un rango muy pequeño y más predecible que en los otros.



En este histograma se puede apreciar lo que he discutido en el segundo caso de la primera tabla, que usando *hackbench* disminuye la latencia media, pero aumenta el rango en el que puede estar, lo que aumenta su aleatoriedad. Algo similar ocurre al usar *bonnie++*.

