**UNIVERSIDAD NACIONAL DE CÓRDOBA**

**FACULTAD DE CIENCIAS EXACTAS, FÍSICAS, QUÍMICAS Y NATURALES**

SISTEMAS DE COMPUTACIÓN

**TRABAJO:** Estación meteorológica

*Informe sobre latencia con diferentes kernels de Linux*

**ALUMNOS:**

* CANCINOS, José María
* OLIVA, Nahuel
* WORTLEY, Agustina

**AÑO:** 2019

**DEFINICIONES**

Latencia:

La latencia es el tiempo que transcurre entre el momento en que se recibe una petición de tarea y el momento en que ella es atendida.

Jitter:

Es la dispersión en el periodo de control de todas las tareas definidas en el sistema.

Rendimiento:

Tiempo en el que la CPU realmente realiza tareas productivas, ejecuta una tarea.

Multitarea:

La ejecución de múltiples tareas se realiza mediante asignación de un tiempo de ejecución o quantum a cada tarea/programa. El microprocesador ejecuta durante dicho tiempo un programa. Terminado el quantum, deja la tarea en un estado intermedio y toma otra tarea.

El tiempo de ejecución es muy pequeño. Así, se puede observar como si el procesador ejecuta tareas al mismo tiempo.

Cambio de contexto (o conmutación de tareas)

Tiempo en que la CPU salva el contexto de la tarea actual y carga el contexto de la nueva tarea para pasar a su ejecución.

Relación entre latencia y rendimiento:

Los términos rendimiento y la latencia son opuestos.

Al disminuir la latencia, el rendimiento de la CPU disminuye también.

Efectos de excesiva latencia

Latencia en sí misma no es un mal parámetro ya que siempre existirá un delay entre el momento en que ocurre la interrupción por parte de la tarea y el momento de su cumplimiento.

Pero cuando se produce una excesiva latencia, dicho delay excede el umbral de tiempo arbitrario de cada aplicación. Superar dicho umbral representa que una aplicación real time ha perdido su funcionalidad.

Kernels de baja latencia o RT

Kernels ideales para aplicaciones donde se necesite una rápida respuesta frente a estímulos externos. El rendimiento de la CPU es penalizado.

Un kernel de baja latencia altera el flujo de datos y peticiones del sistema operativo, de manera tal que dicho flujo ocurra sin interrupciones y con menos retraso entre la entrada y la salida de la señal, disminuyendo la latencia.

Tiempo real

El tiempo en que tarda la CPU en responder a un evento es conocido y predecible

**TIPOS DE KERNELS:**

* Kernel genérico: es el kernel por defecto.
* Kernels de baja latencia (o soft real time): kernel generico configurado para obtener una menor latencia.
* Kernels de tiempo real (o hard real time)

**LABORATORIO:**

**Herramienta cyclictest**

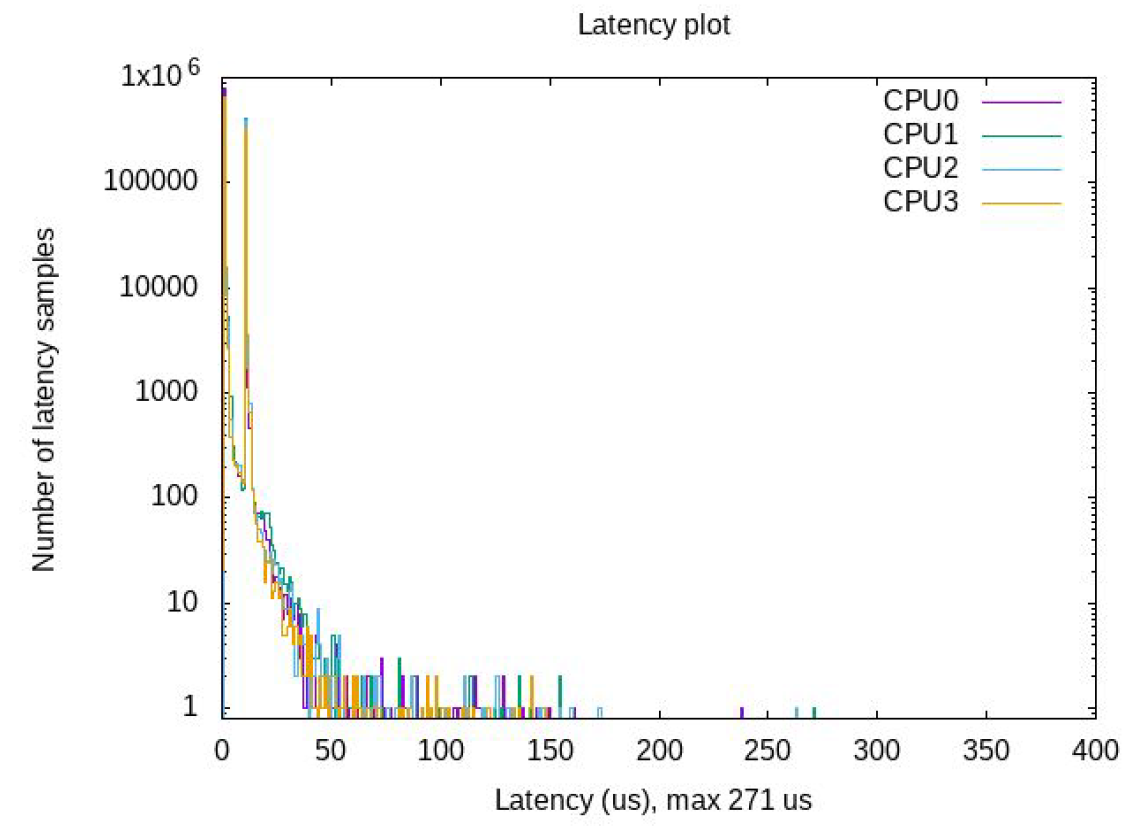
Cyclictest mide el tiempo que transcurre entre que el timer de una tarea expira y el tiempo en el que realmente el hilo corre el timer de la tarea.

1. **KERNEL GENÉRICO**

Se ha obtenido el script para correr la prueba “cycletest” y obtener el gráfico que permite divisar los valores.

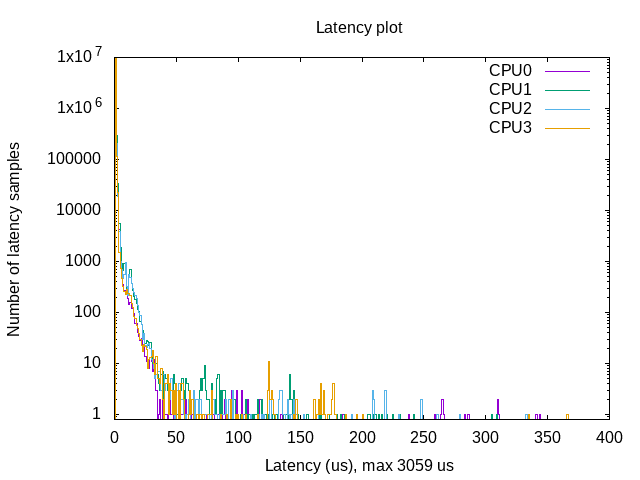
Se ha probado con los siguientes valores:

1. 10⁶



Se observa que la latencia máxima registrada es de . Existen dos picos donde se completan la mayor cantidad de tareas cuya latencia es de (para el primer pico) o (para el segundo pico). El jitter es amplio.

B)



Se realiza la prueba con . Se observa que la mayor cantidad de tareas ha tenido una latencia de y . La latencia máxima ha sido muy grande. . Por lo tanto, el jitter es muy grande.

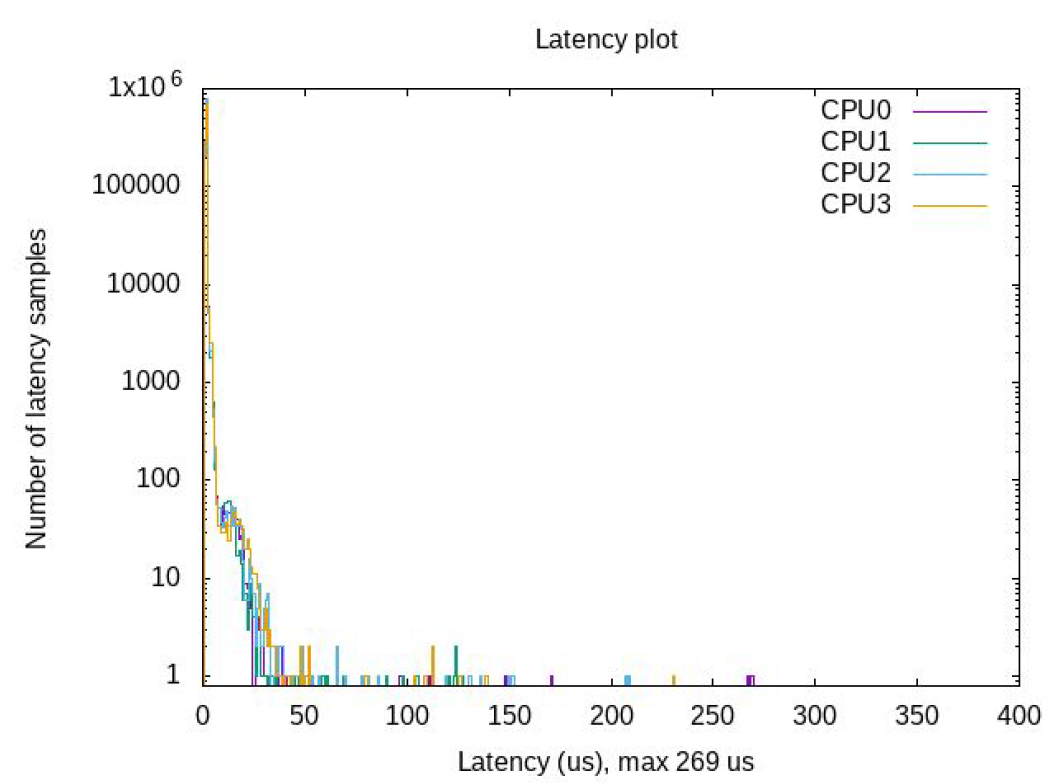
1. **KERNEL DE BAJA LATENCIA**

Se ha instalado el kernel de baja latencia mediante el comando:

- sudo apt-get install linux-lowlatency

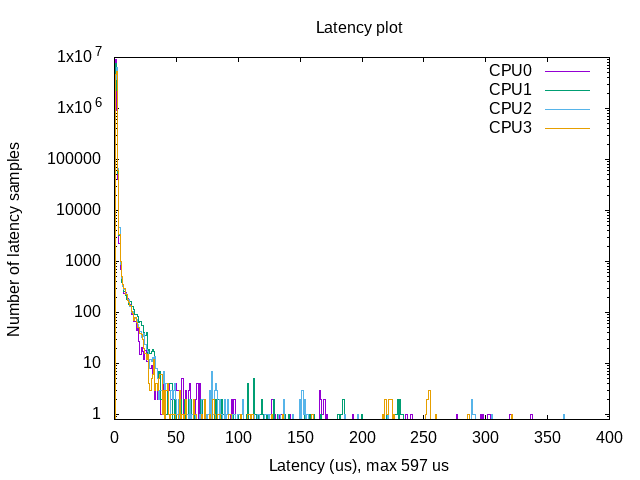
Luego, se ha corrido el script anteriormente nombrado y se han obtenido los siguientes gráficos.

1. 10⁶



Con un kernel de baja latencia, la latencia máxima ha sido de . La mayor cantidad de tareas ha tenido una latencia aproximadamente de . El jitter ha disminuido.

1. 10⁷



La latencia máxima ha sido de . Esto equivale a una gran disminución (una sexta parte) frente al valor obtenido con un kernel genérico. La mayor cantidad de tareas tiene una latencia entre y aproximadamente. Y el jitter ha disminuido relativamente.

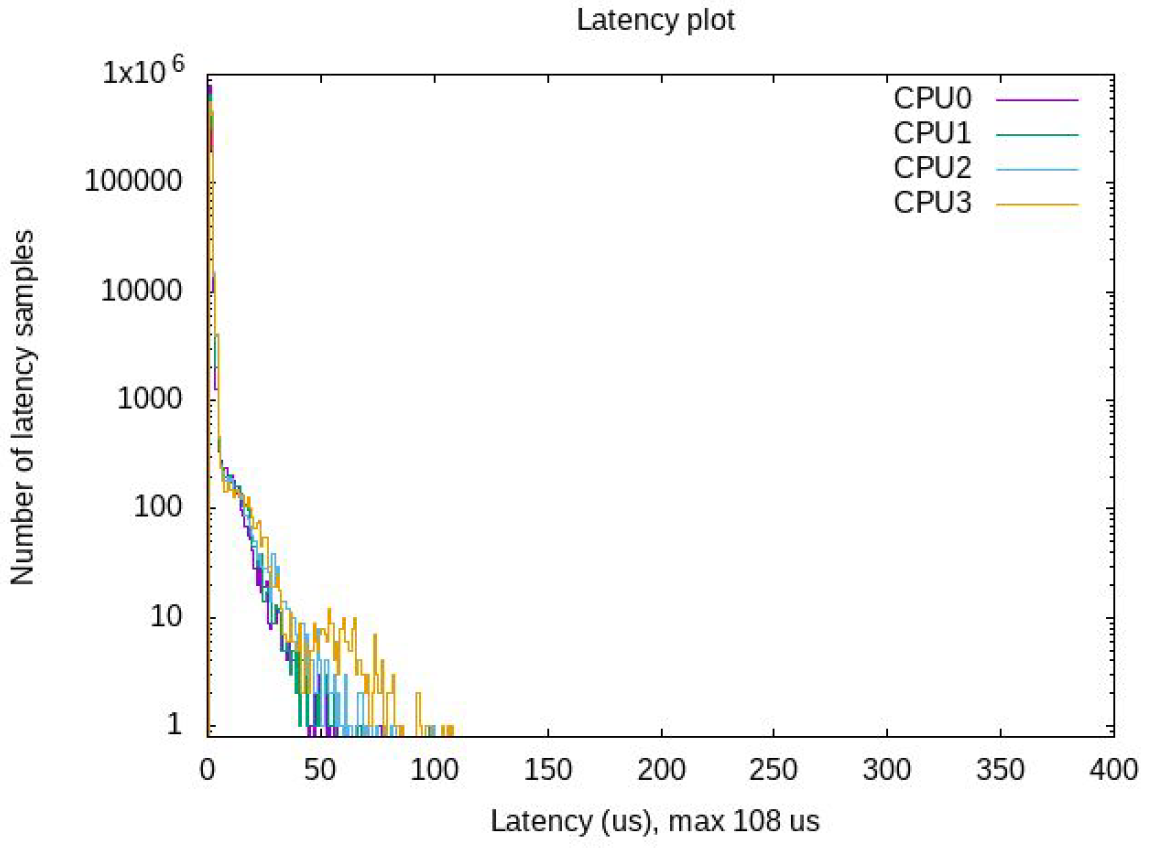
1. **KERNEL DE TIEMPO REAL**

La imagen de kernel de tiempo real empleada fue:

- Linux-image-4.16.12-rt5-klue-rt\_amd64.deb

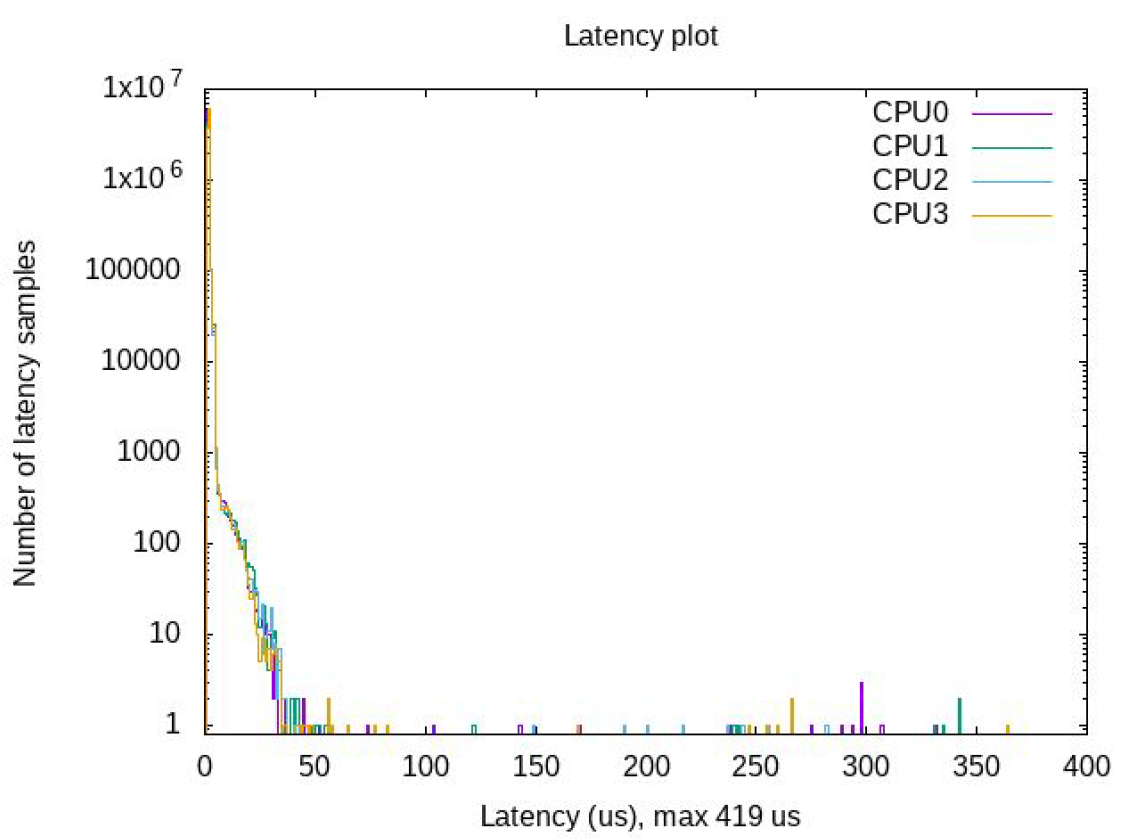
Los resultados de la prueba fueron:

1. 10⁶



Finalmente, al corer la prueba en un kernel de tiempo real, la latencia máxima obtenida se ha reducido a una tercera parte del valor anterior. El valor actual es . La dispersión, por lo tanto, ha disminuido considerablemente estando entre y . Y la mayor cantidad de tareas se realiza con una latencia casi cero.

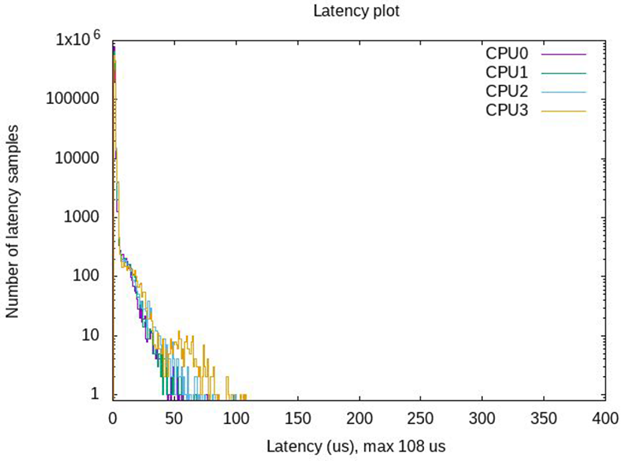
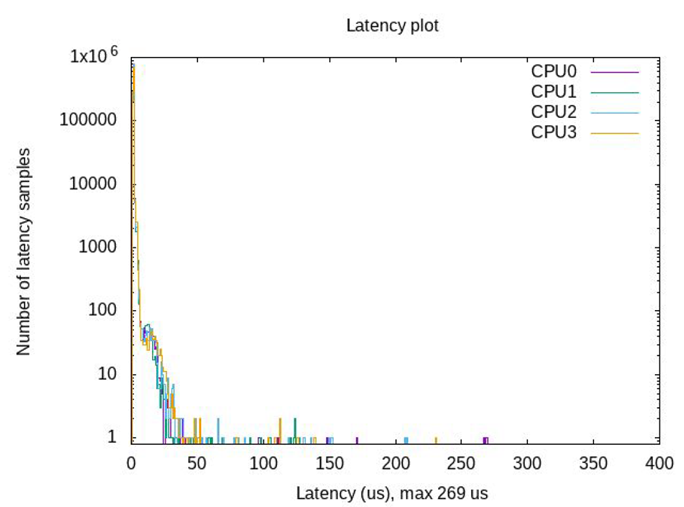
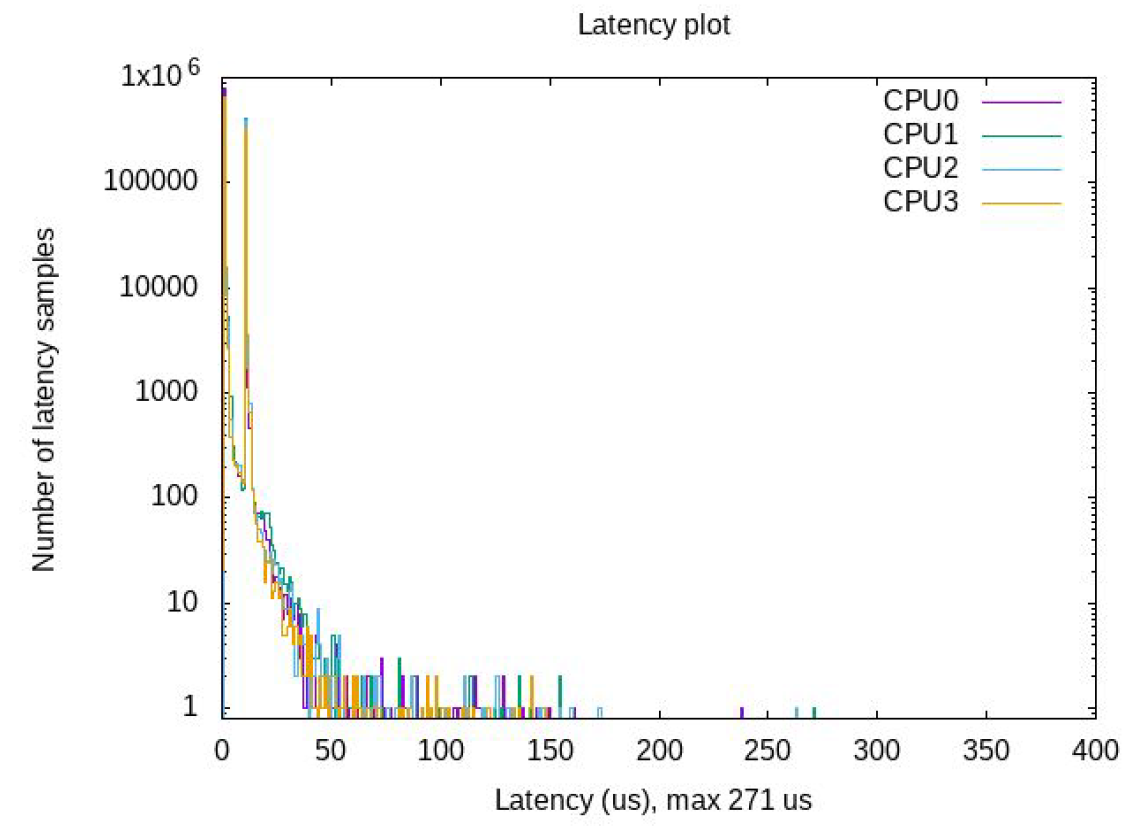
1. 10⁷



Para una prueba con muestras, la latencia ha disminuido casi y se ha obtenido una valor de latencia máxima de . El jitter ha disminuido. Y la mayor cantidad de tareas se ha realizado con latencia tendiente a cero.

RESÚMEN

|  |
| --- |
| Prueba con muestras:   * Kernel genérico (gráfico superior izquierdo) * Kernel de baja latencia (superior derecho) * Kernel de tiempo real (inferior izquierdo) |



GENÉRICO**:**

Latencia máxima:

Jitter (hasta 10 tareas): medio. a

Pico: 10⁶ en y entre

BAJA LATENCIA:

Latencia máxima: – poca diferencia con respecto a kernel genérico.

Jitter: medio. La mayoría está entre a

Pico: 10⁶ en

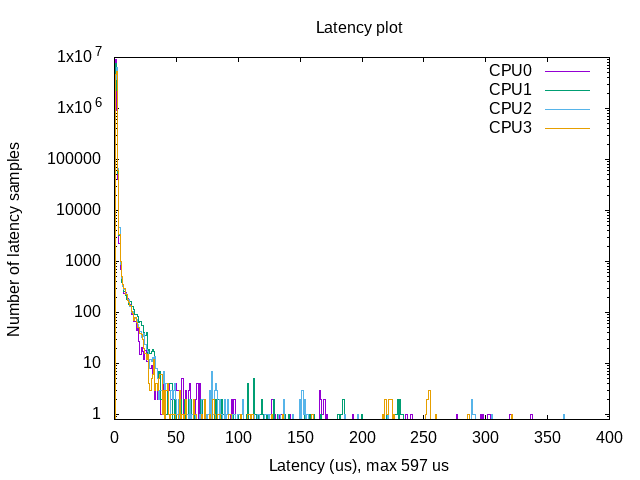
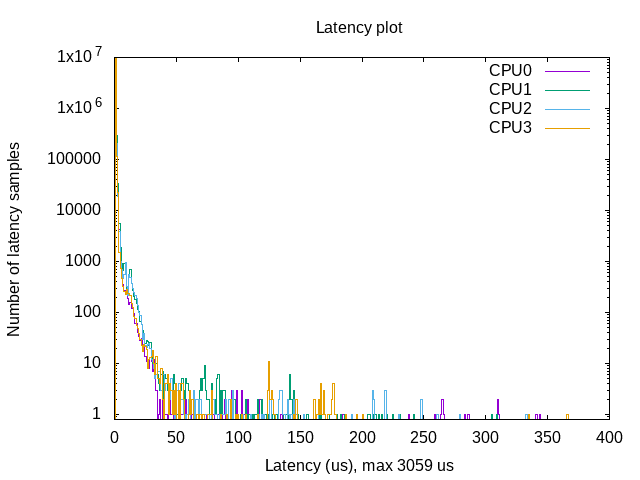
TIEMPO REAL:

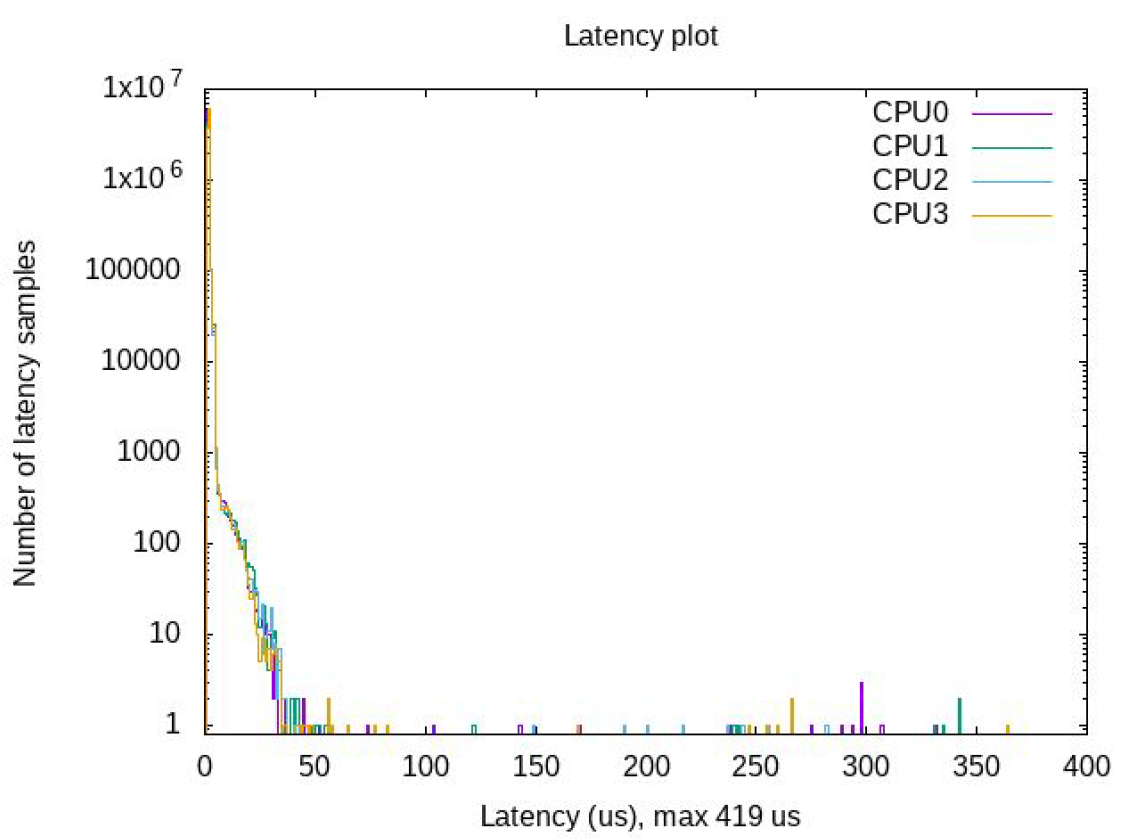
Latencia máxima: – gran diferencia con respecto a kernels anteriores.

Jitter: amplio. La mayoría está entre a

Pico: 10⁶ en

|  |
| --- |
| Prueba con muestras:   * Kernel genérico (gráfico superior izquierdo) * Kernel de baja latencia (superior derecho) * Kernel de tiempo real (inferior izquierdo) |





GENÉRICO**:**

Latencia máxima:

Jitter (hasta 10 muestras): muy amplio. a

Pico: entre

BAJA LATENCIA:

Latencia máxima: – gran diferencia con respecto a kernel genérico.

Jitter: medio. La mayoría está entre a

Pico: entre

TIEMPO REAL:

Latencia máxima: – buena diferencia con respecto a kernel de baja latencia.

Jitter: bajo. La mayoría está entre a

Pico: entre