Servidor Web Tomcat

José Manuel Cuevas Muñoz, Daniel Ranchal Parrado, Carlos Romeo Muñoz

9 de Diciembre 2018

Índice general

Índice de figuras

Capítulo 1

Definición del Sistema

1.1. Objetivos y definición del sistema

El objetivo que se pretende es hacer pruebas exhaustivas, es decir, hacer un benchmarking para calcular el rendimiento del servicio web Tomcat. Para hacer este tipo de pruebas, se parte de la siguiente información. El sistema operativo bajo el que se están haciendo las pruebas es CentOS, cuyo servidor está alojado en Azure.
Para la instalación de Tomcat, se ha utilizado el servicio de virtualización docker para instalarlo de una manera sencilla. Para que el host, es decir, el servidor pueda dar el servicio que provee el contenedor de docker, hay que ligar el puerto del contenedor en el que está sirviendo Tomcat con el puerto que nosotros queramos en el host, aunque los autores de este guión y trabajo hemos elegido el 80, el puerto por excelencia para el servidor web.

1.2. Servicios y sus posibles resultados

El servidor web Tomcat nos provee una gran cantidad de utilidades para aprovechar al máximo este mismo. La primera y la más importante es el despliegue de una página web. Tomcat nos permite hacerlo de manera estática, es decir, configurar la aplicación antes de activar este servicio o de manera dinámica, que es lo más utilizado para servidores que están en producción. Para poder realizar esto de una manera sencilla, Tomcat nos da una herramienta llamada "Manager", que cómo se ha comentado antes, se pueden hacer despliegues y eliminación (undeploy) de cualquier aplicación web además de darnos una lista de las aplicaciones que ya están desplegadas.

Pero las utilidades de "Manager"no son sólo esas, este gestor nos da la opción de poder ver las estadísticas de las sesiones de cualquier aplicación y el estado del servidor. Otra de las funcionalidades que tiene, al igual que el servidor httpd y apache2, es la posibilidad de trabajar con "Virtual Host". Otra característica interesante que nos ofrece Tomcat es la gestión de los usuario y los roles en las distintas aplicaciones web. Esto evita que tengamos una tabla para cada aplicación web que se tenga. En temas de seguridad, Tomcat nos permite la configuración de los certificados SSL/TLS para que nuestra página tenga el protocolo https.
Para este experimento, se distinguirán los siguientes resultados:

- **La carga correcta de las imágenes en un tiempo considerablemente bueno.**
- **La carga correcta de las imágenes en un tiempo pésimo.**
- **La carga parcial de las imágenes.**
- **Fallo del servidor web. Ninguna imágen es servida.**

1.3. Métricas

Para poder medir la eficacia y la actuación del servidor web Tomcat se han considerado los siguientes parámetros a examinar:

- **Peticiones por segundo**
- **Tiempo por cada grupo de peticiones al mismo tiempo**
- **Tiempo por cada petición**
- **Uso de la CPU**
- **E/S Disco**
- **E/S Red**

1.4. Parámetros

Los parámetros que pueden afectar a la medición del rendimiento del servidor web Tomcat son los siguientes:

- **La lejanía con el centro de datos**
- **Los procesos que se ejecutan en segundo plano**
- **Estado de nuestra conexión a Internet**
- **Hardware empleado en el servidor**

Para poder mitigar los efectos que producen estos parámetros,la prueba se ejecutará varias veces y se harán los cáculos oportunos para representar de una manera correcta las estadísticas.

Capítulo 2

Evaluación del Sistema

Definición de la experimentación a realizar. Qu evaluacin se seguir, cual ser la carga de trabajo, qu experimentos se van a realizar y finalmente se analizarn los resultados de la experimentacin.

2.1. Tcnicas de evaluacin

Qu tcnica de evaluacin se seguir.

2.2. Carga de trabajo

Cual ser la carga de trabajo, en qu consiste.

2.3. Diseo de Experimentos

Cmo se han diseado los experimentos. En qu consiste la experimentacin a realizar.

2.4. Anlisis de los resultados

Estudio y anlisis de los resultados provenientes de la experimentacin. Tablas y grficas con los datos (se requiere un nmero suficiente de muestras resultado de la experimentacin)

Capítulo 3

Conclusiones y discusión

En primer lugar, se aprecia un fuerte problema en el hardware de este servidor debido al cuello de botella que aparece al ejecutar los benchmarks en la CPU, esto es fácil de comprender, debido a que la concurrencia toma protagonismo en estos tests, con algunos fijándola en valores como 10 peticiones concurrentes. Esta teoría es afirmada por las gráficas mostradas, donde vemos fácilmente como el procesador llega a funcionar a niveles cercanos al 100 % de utilidad durante la duración de las pruebas efectuadas.

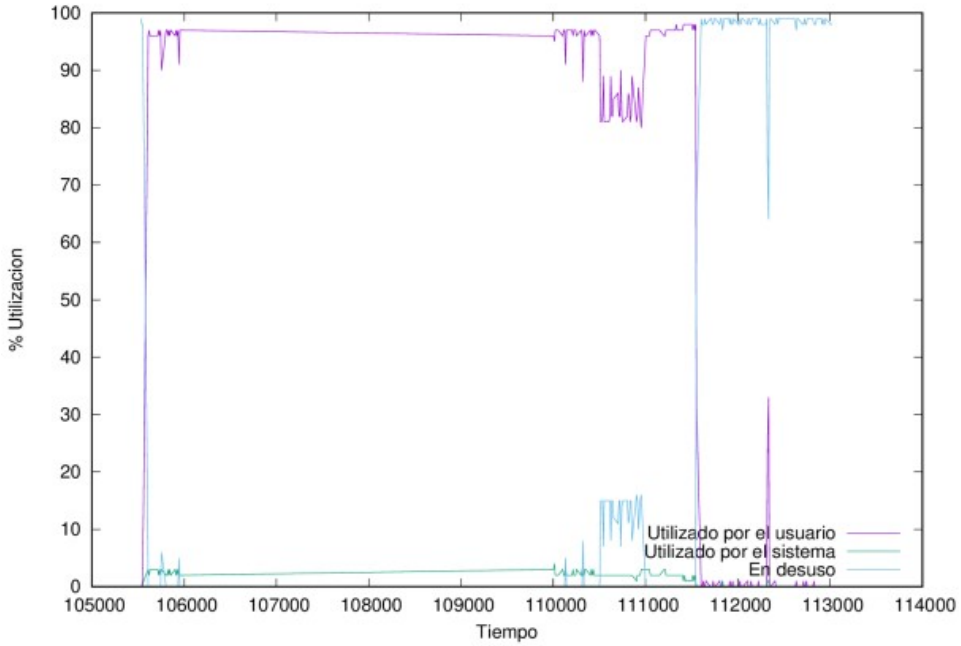

Figura 3.1: Uso de CPU con imágenes grandes

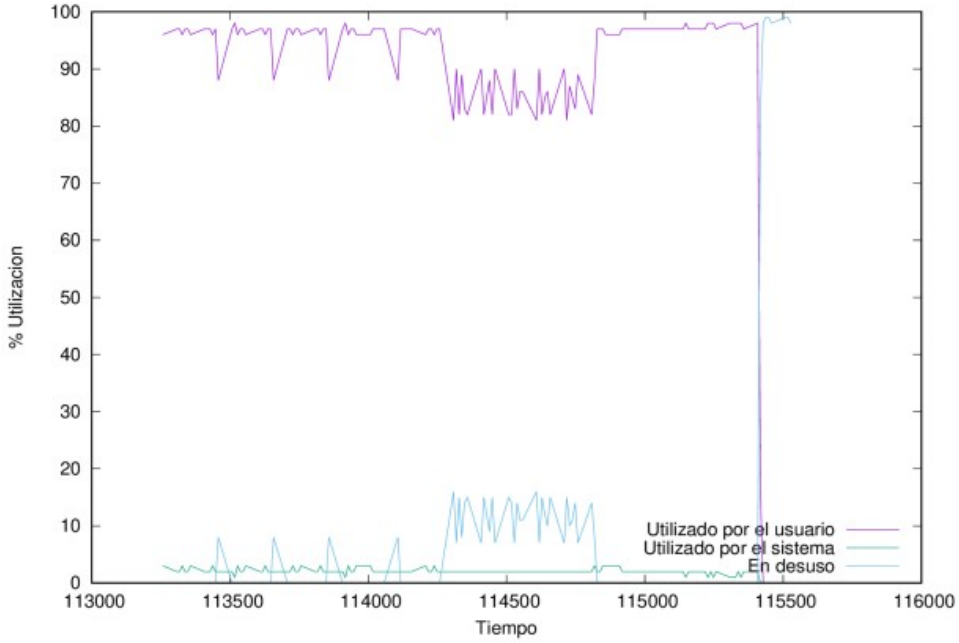

Figura 3.2: Uso de CPU con imágenes medianas

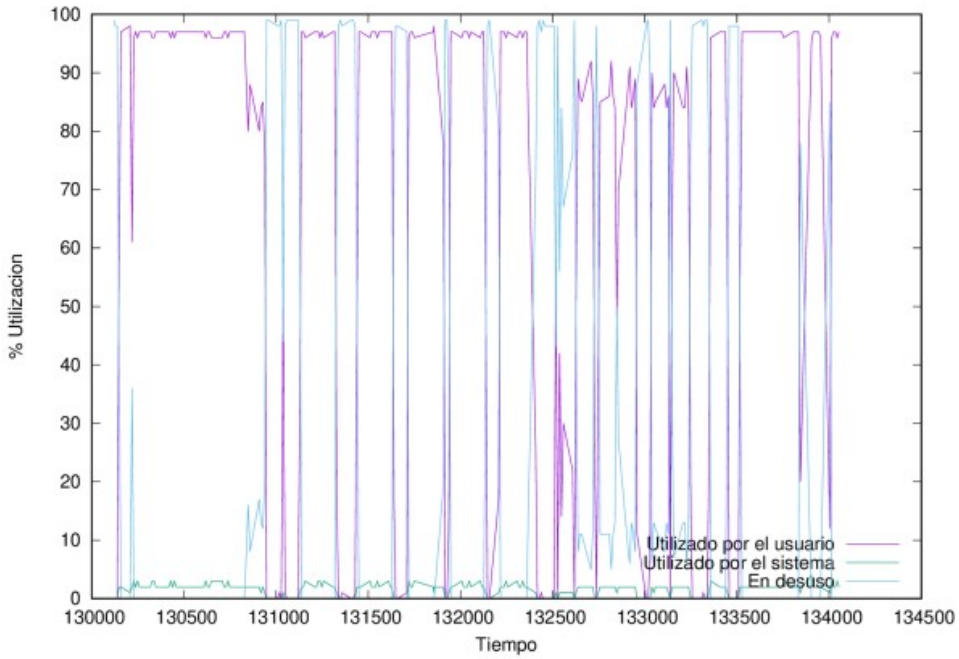

Figura 3.3: Uso de CPU con imágenes medianas

Además de esto, otro factor determinante en este caso es el bajo número de transferencias por segundo que el servidor es capaz de aguantar y que está fuertemente relacionado con el almacenamiento del mismo en el cual se almacenan las fotografías que sirven como cargas de prueba, el cual en ninguna prueba es capaz de llegar ni tan siquiera a la unidad, y eso es algo que los encargados de testear su funcionamiento hemos conocido de primera mano al tener que esperar grandes cantidades de tiempo a que todos los tests fueran completados varias veces con éxito. En cuanto a las peticiones por segundo, casi nunca superan el cuarto de unidad, hecho que no hace más que corroborar las afirmaciones anteriores.

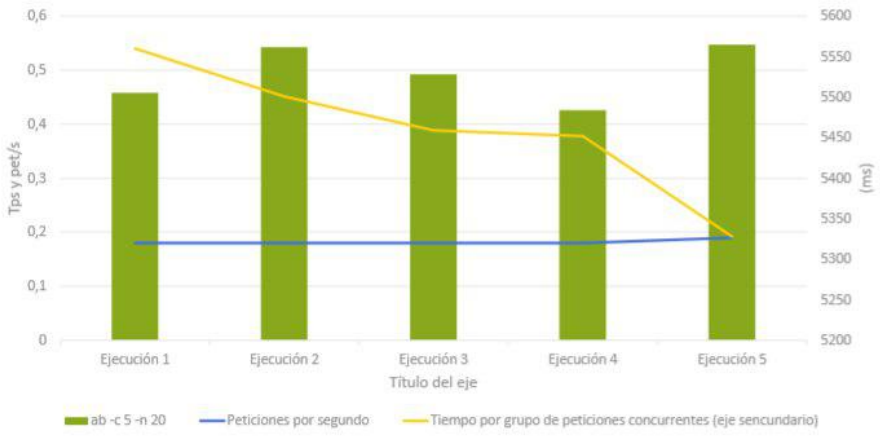

Figura 3.4: Transferencias por Segundo, peticiones por segundo y tiempo empleado en un grupo de peticiones concurrentes.

Si contemplamos el gráfico 3.4, nos damos cuenta de que el número de peticiones por segundo, al ser un parámetro íntimamente ligado al hardware sobre el que se esté ejecutando el banco de pruebas escogido, apenas varía, puesto que la CPU solo es capaz de ejecutar un número determinado de ellas, siendo todas iguales. Lo interesante de este gráfico, es ver como el tiempo empleado en servir un grupo de peticiones concurrentes decae a lo largo de las cinco ejecuciones del test. Tras indagar en los motivos de este aumento del rendimiento, nos damos cuenta de que Azure es un servicio de computación en la nube que alardea de proveer de una caché inteligente a sus máquinas virtuales que estén corriendo en sus servidores, como es nuestro caso, eso por ello por lo que se aprecia esta mejora en el rendimiento,

debido a que gran parte de los datos que son necesarios para la ejecución de la prueba, ya están cargados en la caché del servidor y son accesibles de manera mucho más rápida para la siguiente ejecución.

Si contemplamos las gráficas 3.5, 3.6 y 3.7, no hay gran diferencia entre ellas en cuanto al parámetro de medición, el número de transferencias por segundo del disco del servidor. Para analizar las gráficas, recordemos que estos datos han sido tomados con el monitor sar, mientras que ab era ejecutado, tras ello, se ha realizado la media aritmética de los valores del parámetro interesado que sar tomaba cada 10 segundos durante la ejecución del test. Entonces, podemos ver como apenas hay diferencias significativas entre las ejecuciones, por ejemplo, en la gráfica 3.5, vemos como normalmente el test de menor concurrencia y menos repeticiones (línea amarilla), tarda normalmente menos tiempo que el test con más concurrencia y el doble de repeticiones (línea verde), aunque veamos ejecuciones donde estas posiciones se inviertan, esto se debe más al margen de error que a otra cosa. En cuanto a la gráfica 3.7, la referida a los resultados del mosaico de imágenes pequeñas, observamos como la línea azul, la correspondiente con el uso de https, lo cual es curioso debido a que en las dos últimas pruebas se habían mantenido por debajo, sin embargo, no es de preocupación, ya que no se aprecian diferencias muy significativas en la mayoría de las ejecuciones, sin embargo, vemos como esa línea sube hasta una transferencia por segundo en la segunda ejecución, lo cual es muy significativo. La imagen general de estas pruebas es que mientras más grandes sean las imágenes del banco de pruebas, menos transferencias por segundo se llevarán a cabo, lo cual es lógico, puesto que son archivos más grandes.

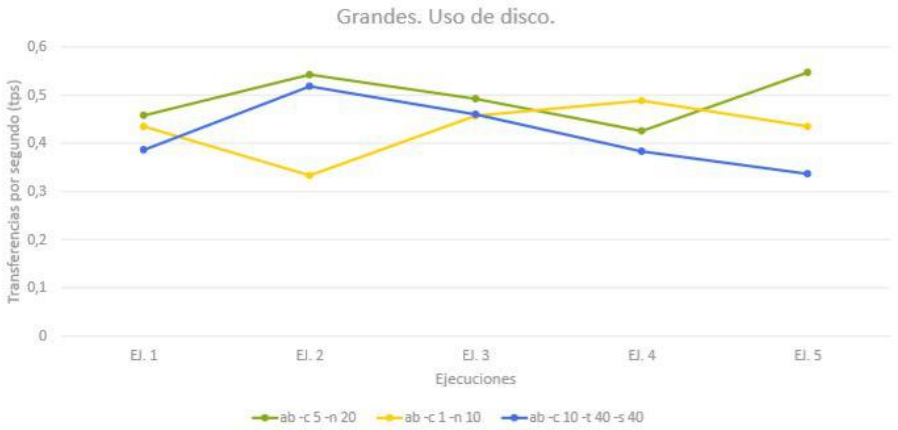

Figura 3.5: Uso de disco en las cinco ejecuciones con imágenes grandes

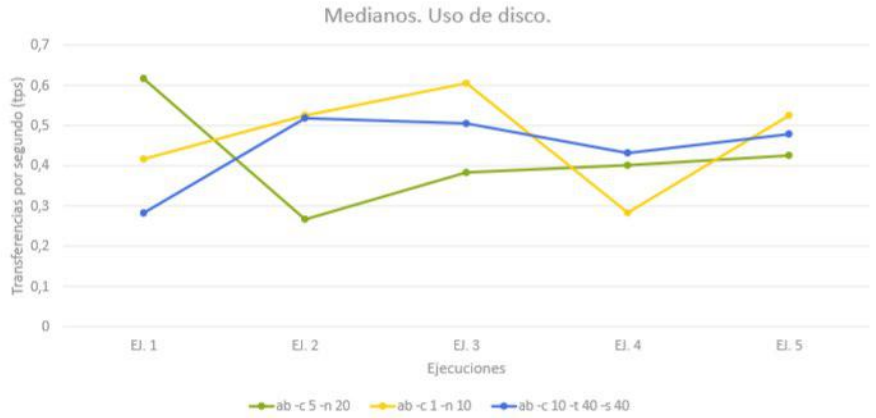

Figura 3.6: Uso de disco en las cinco ejecuciones con imágenes medianas

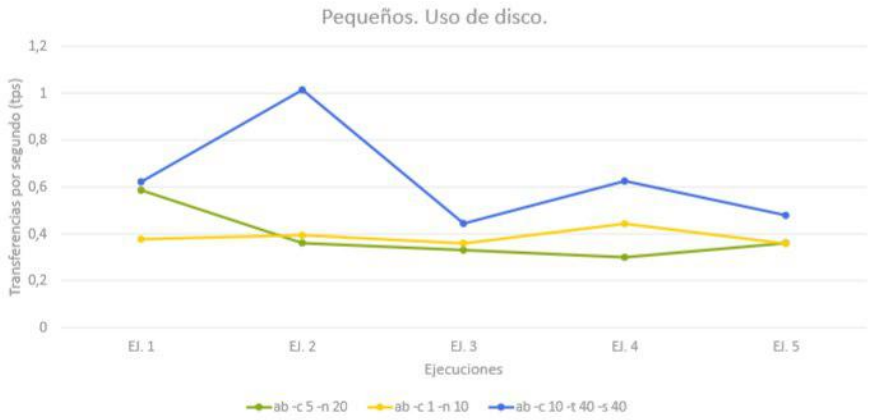

Figura 3.7: Uso de disco en las cinco ejecuciones con imágenes pequeñas

Si analizamos ahora el uso de la red del servidor, aquí no tenemos ningún problema de cuello de botella, Azure nos proporciona una velocidad de conexión veloz y estable y no es algo que un administrador de un servidor pequeño como este tenga que preocuparse, por supuesto, mientras más exigente sea el test con la CPU, lo será con el uso de red que éste requerirá, puesto que estamos hablando de un servidor web.

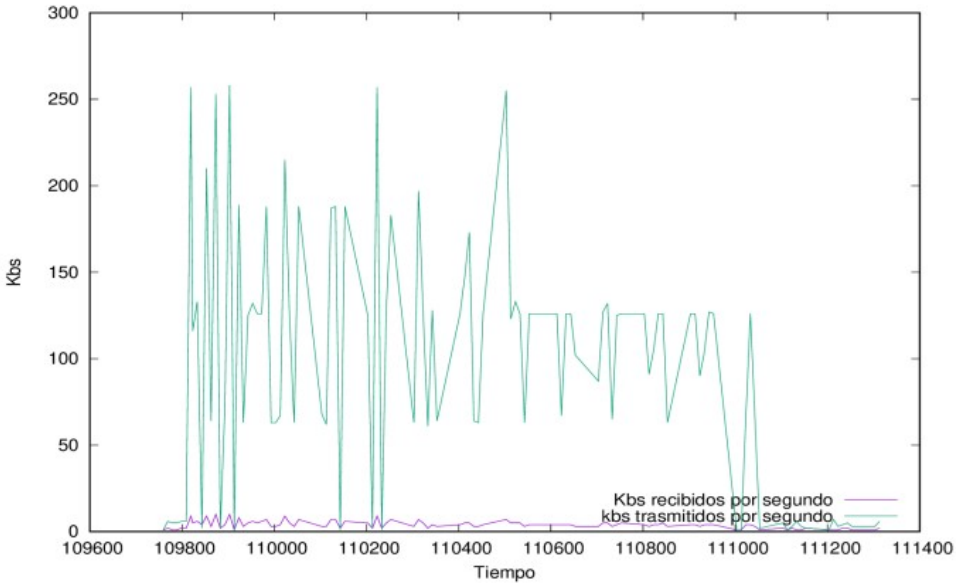

Figura 3.8: Uso de Red con imágenes grandes

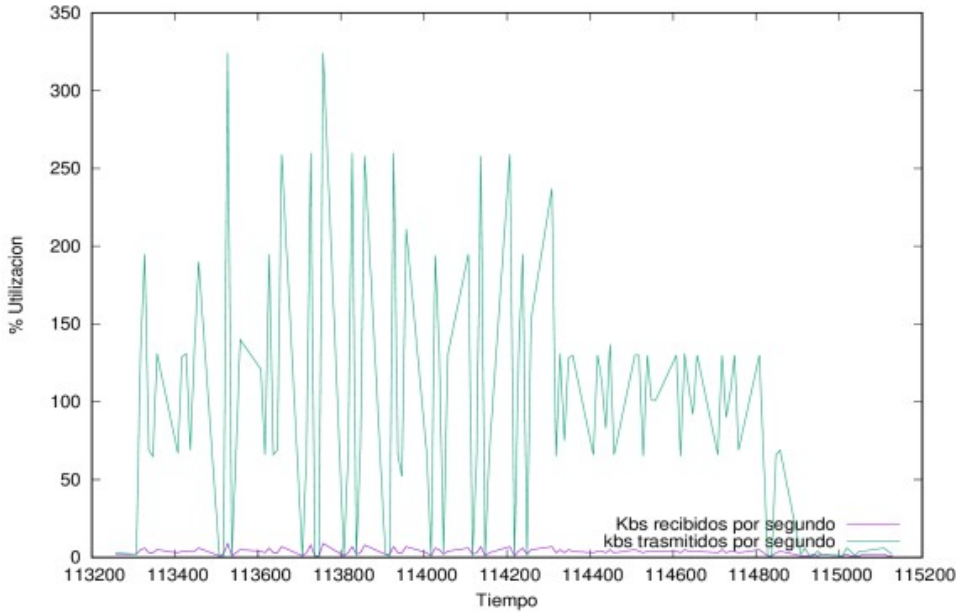

Figura 3.9: Uso de Red con imágenes medianas

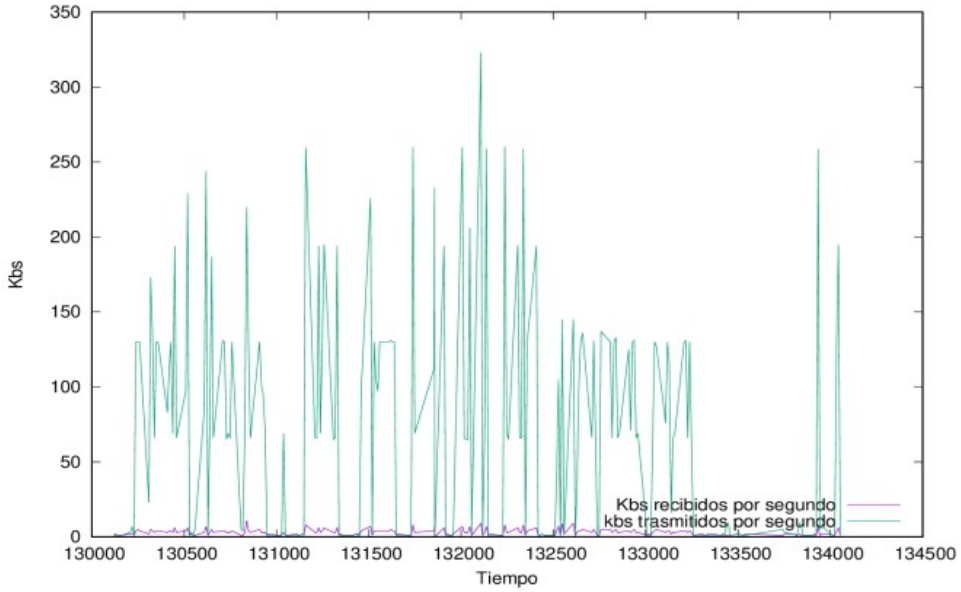

Figura 3.10: Uso de Red con imágenes pequeñas

En conclusión, hemos detectado que nuestro cuello de botella es una CPU que a pesar de ser muy potente en el conjunto de todos sus núcleos, apenas

aguantaría un servidor web pequeño como este ante un importante número de peticiones al servidor, esto es debido a que la cuenta de estudiante de Microsoft Azure que estamos empleando para este trabajo solo ofrece un núcleo de procesador, potente, si, pero que flaquea en tareas que involucren una alta concurrencia de peticiones. Y es que ese es el argumento por el cual los procesadores empleados para su uso en servidores de cualquier tipo suelen contar con un gran número de núcleos tanto físicos como lógicos y poder procesar rápidamente todas las peticiones que reciba. En cuanto a la referenciación, sabemos que los resultados aquí mostrados corresponden con los de un servidor de especificaciones limitadas, útil para este propósito, pero de limitado uso, y es que, tras haber ejecutado numerosos bancos de pruebas centrados en multitud de aspectos de un sistema informático, entre ellos Apache Benchmark, hemos encontrado que fácilmente, un ordenador doméstico con todos sus núcleos activados puede sobrepasar fácilmente el rendimiento de este procesador, organizaciones como SPEC tienen precisamente esa función, la de examinar este tipo de equipos informáticos para poder realizar comparativas, lo cual nos es muy útil a la hora de sacar conclusiones sobre el desempeño de este hardware. Por último, hemos de decir, que Microsoft Azure es una útil herramienta para alojar cualquier tipo de servidor o aplicación web y que nos ha sido muy útil para la realización de este trabajo y que, previo pago de la capacidad necesaria, se puede obtener muchísima potencia de cómputo de ella.

Bibliografía

[1] Kimball and Merz. *The data webhouse toolkit.* John Wiley, 2000.

[1]