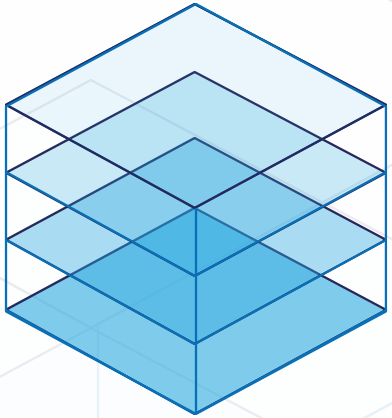


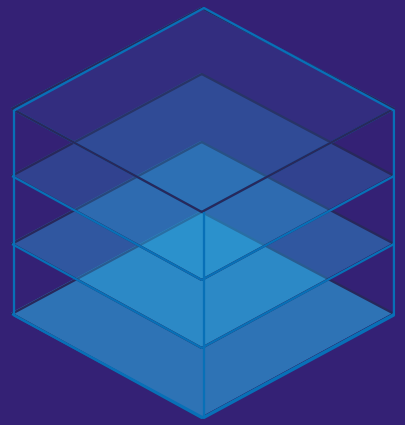
MISO

Maestría en Ingeniería de Software

Entrega 3 Sistema Conversión Cloud

Sistema de Conversión Cloud - Escalabilidad en la Capa Web

Camilo Ramírez Restrepo
Laura Daniela Molina Villar
Leidy Viviana Osorio Jiménez
Tim Ulf Pambor
Shadit Perez

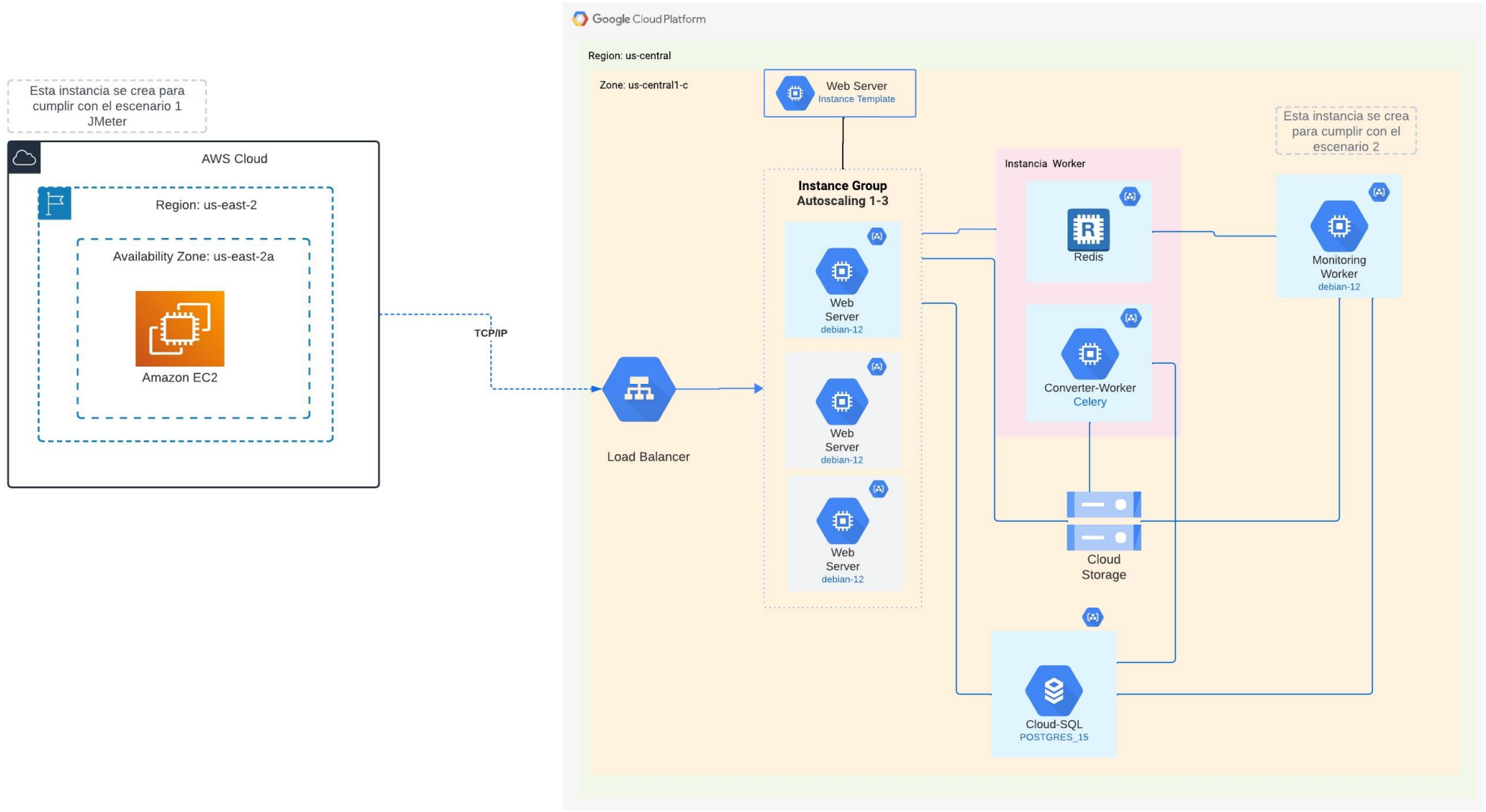


MISO

Maestría en Ingeniería de Software

Arquitectura

Arquitectura



Configuración del sistema

MISO
Maestría en Ingeniería de Software

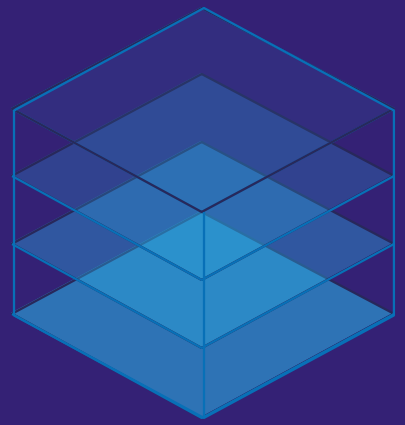
Entorno GCP

- Google Cloud Monitoring: Utilizamos Google Cloud Monitoring para supervisar y gestionar nuestro entorno en la nube.
- Compute Engine: Se ha implementado Compute Engine con una instancia e2-highcpu-2 que utiliza Debian 12. Esta máquina virtual cuenta con etiquetas para permitir el tráfico HTTP desde el Load Balancer y Health Check y actúa como receptor de información relacionada con la base de datos.
- Instancia worker: Se ha configurado una instancia worker que ejecuta dos contenedores: uno con Redis y otro con Celery. Esta configuración optimiza la gestión de tareas y procesos.
- Cloud Storage: Para el almacenamiento de videos, hemos integrado Cloud Storage, aprovechando su robusta capacidad de almacenamiento en la nube.
- Compute Engine: Se ha implementado otra instancia Compute Engine llamada "monitoring-worker", utilizando una instancia e2-highcpu-2, para ejecutar las pruebas de estrés del escenario 2.
- Cloud SQL Postgres: Para implementar una base de datos Postgres con 1 vCPU y 4 GB de memoria, garantizando un rendimiento eficiente y escalable.
- Load Balancer: Integramos un Regional external Application Load Balancer para distribuir equitativamente la carga de tráfico entre las instancias, optimizando así el desempeño y la disponibilidad del sistema.
- Managed Instance Group e Instance Template: Se ha configurado un Managed Instance Group y un Instance Template para gestionar y escalar fácilmente las instancias, proporcionando flexibilidad y alta disponibilidad en nuestro entorno en la nube. Las instancias del Managed Instance Group se basan en el Instance Template.
- Autoscaling: Se activa autoscaling para el Managed Instance group, así que crean/eliminan instancias de acuerdo con la carga. Cuando se supera el objetivo del 55% de uso promedio de memoria, se crea una nueva instancia hasta un máximo de 3 instancias. Cuando se cae por debajo de nuevo, se elimina una instancia hasta el mínimo de una instancia.

- **Aplicación Flask**

Despliegue Arquitectura GCP

Configuración GCP infraestructura como código: Documento de configuración



MISO

Maestría en Ingeniería de Software

Experimento 1 – Agregar tarea a la cola

Experimento 1

El objetivo de este plan es evaluar la capacidad de la aplicación Cloud conversión tool y su infraestructura de soporte en un entorno Cloud con autoscaling en la capa web, para determinar sus máximos aceptables. El objetivo es comprender cómo la aplicación responde a diferentes niveles de carga de usuarios y cuál es su capacidad máxima.

• Configuración AWS y JMeter: Documento

Esta arquitectura combina servicios de AWS y Google Cloud para crear un entorno distribuido y escalable que satisface los requisitos del Escenario 1, con pruebas de rendimiento realizadas desde una instancia de JMeter en AWS y la infraestructura principal en Google Cloud.

Configuración del sistema

Servidores EC2

- Capacidad de CPU: Cada instancia t2.medium ofrece 4 vCPUs, permitiendo la ejecución simultánea de múltiples instancias para simular usuarios concurrentes.
- Sistema Operativo: Utilizaremos Debian 12 en las instancias EC2 para llevar a cabo las pruebas.
- Configuración de Red: Las instancias EC2 proporcionan una conexión de red de hasta 10 Gbps para satisfacer los requisitos de ancho de banda necesarios para la transmisión de videos.
- Implementación sobre la instancia AWS: Se llevará a cabo la instalación de JMeter sobre la instancia para facilitar y gestionar las pruebas de rendimiento.

Experimento 1

Métricas

Throughput: cantidad de peticiones procesadas por minuto.

Tiempo de respuesta (P95): percentil 95% del tiempo máximo que tarda la aplicación en procesar una petición

Tiempo de respuesta (P99): percentil 99% del tiempo máximo que tarda la aplicación en procesar una petición

Utilización de recursos: monitoreo de la CPU, memoria y uso de red durante las pruebas.

Criterios de aceptación

Throughput: La aplicación debe ser capaz de procesar 100 peticiones por minuto.

Tiempo de respuesta:

- El tiempo de respuesta de la aplicación en todos los escenarios de prueba no debe superar los 0.5 segundos en el 95% de las transacciones, por petición.
- El tiempo de respuesta en ningún escenario de prueba debe superar los 4 segundos en el 99% de las transacciones.

Utilización de recursos: Durante las pruebas con 100 peticiones concurrentes, la CPU del servidor alcanza un pico de 80% y la memoria se mantendrá < 80% de uso.

Resultados

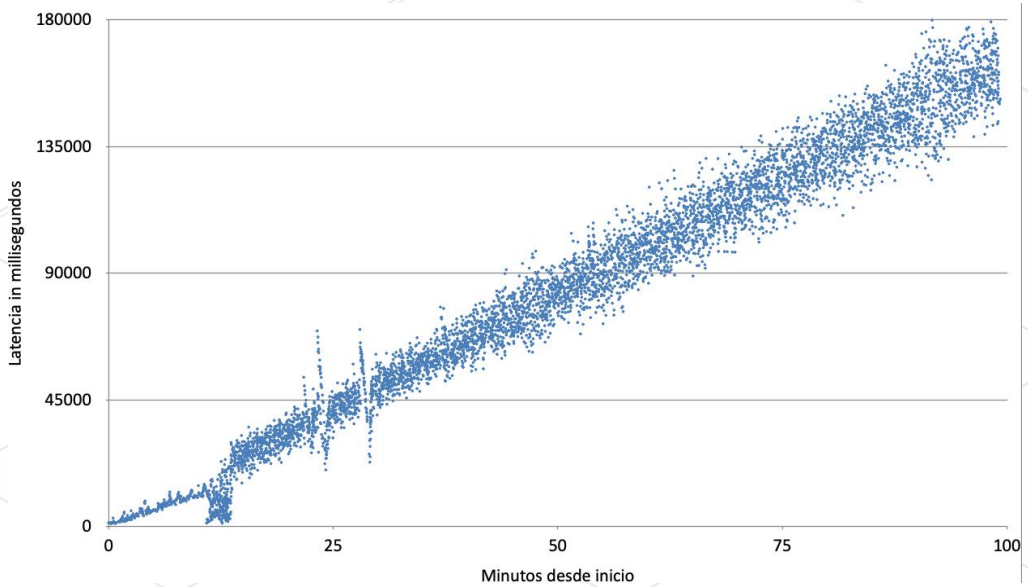
El experimento se inició con 1 usuario concurrente. Cada 30 segundos se agregó un nuevo usuario concurrente, y a lo largo del tiempo se observaron los siguientes eventos clave:

- Inicio: Inicio del experimento a las 7:17pm.
- Escalamiento
  - Primer Scale-out (1 instancia a 2 instancias) después de 18 minutos a las 7:35pm con 36 usuarios concurrentes.
  - Segundo Scale-out (2 instancias a 3 instancias) después de 48 minutos a las 8:05pm con 96 usuarios concurrentes.
- Degradación: Se observó una degradación a partir de 186 usuarios concurrentes después a las 8:50pm.
- **Colapso Total:** El sistema colapsó completamente con 200 usuarios concurrentes a las 8:57pm.
- Fin de pruebas de estrés con JMeter a las 9:03pm.
- **Escalamiento Inverso (Scale-in):**
  - Primer Scale-in (3 instancias a 2 instancias) a las 9:16pm.
  - Segundo Scale-in (2 instancias a 1 instancia) a las 9:53pm.

Adicionalmente, las peticiones fueron enviadas desde AWS utilizando JMeter, y los resultados del escenario 1 indicaron un throughput de 1.25 req/s: Resultados escenario 1

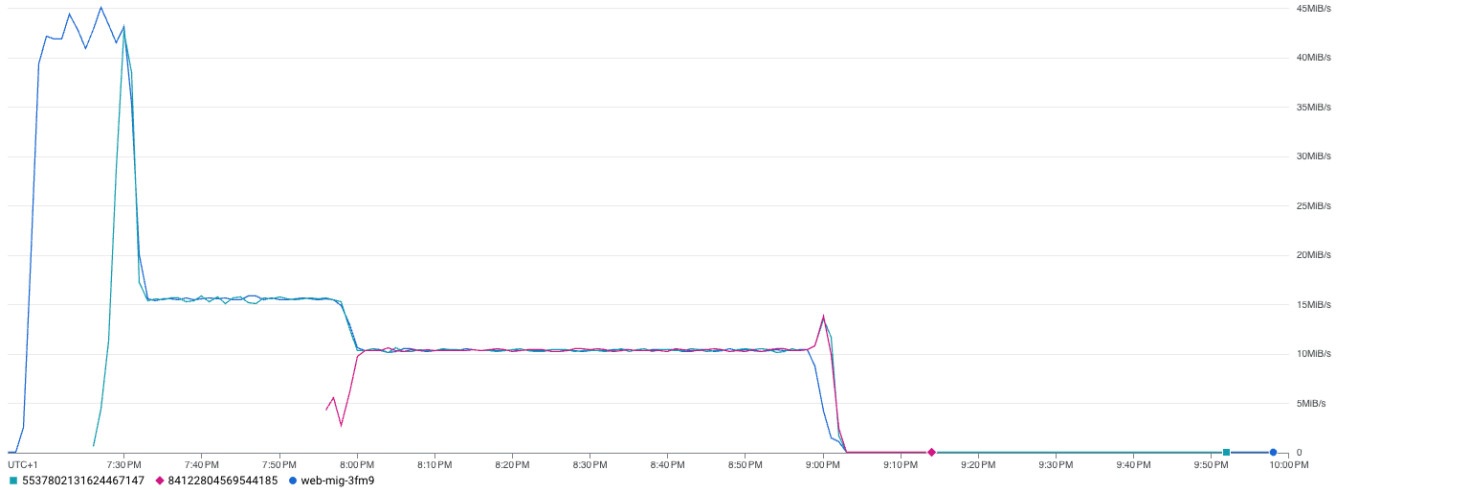
Se generaron gráficas para visualizar diferentes aspectos del sistema, incluyendo el uso de CPU, uso de RAM, el comportamiento del Load Balancer, la latencia y el tráfico de red.

Resultados - Latencia

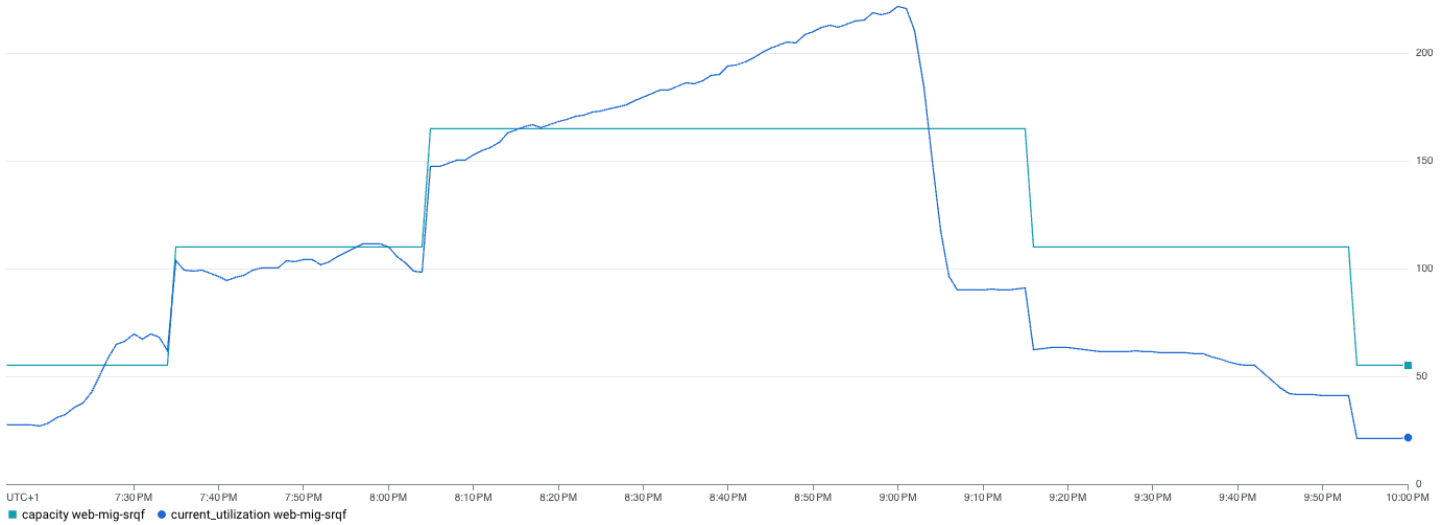


La latencia media aumenta linealmente con el número de usuarios. Esto se puede atribuirse al hecho de que el ancho de banda de red entre la máquina en AWS y GCP es fijo y no aumenta con los usuarios, lo que resulta en menos ancho de banda por usuario con un número creciente de usuarios. Este efecto también se puede evidenciar en el tráfico de red recibido por cada instancia, que se reduce al agregar más instancias en lugar de mantenerse al mismo nivel.

Resultados - Recursos utilizados Servidor Web - Red Recibido

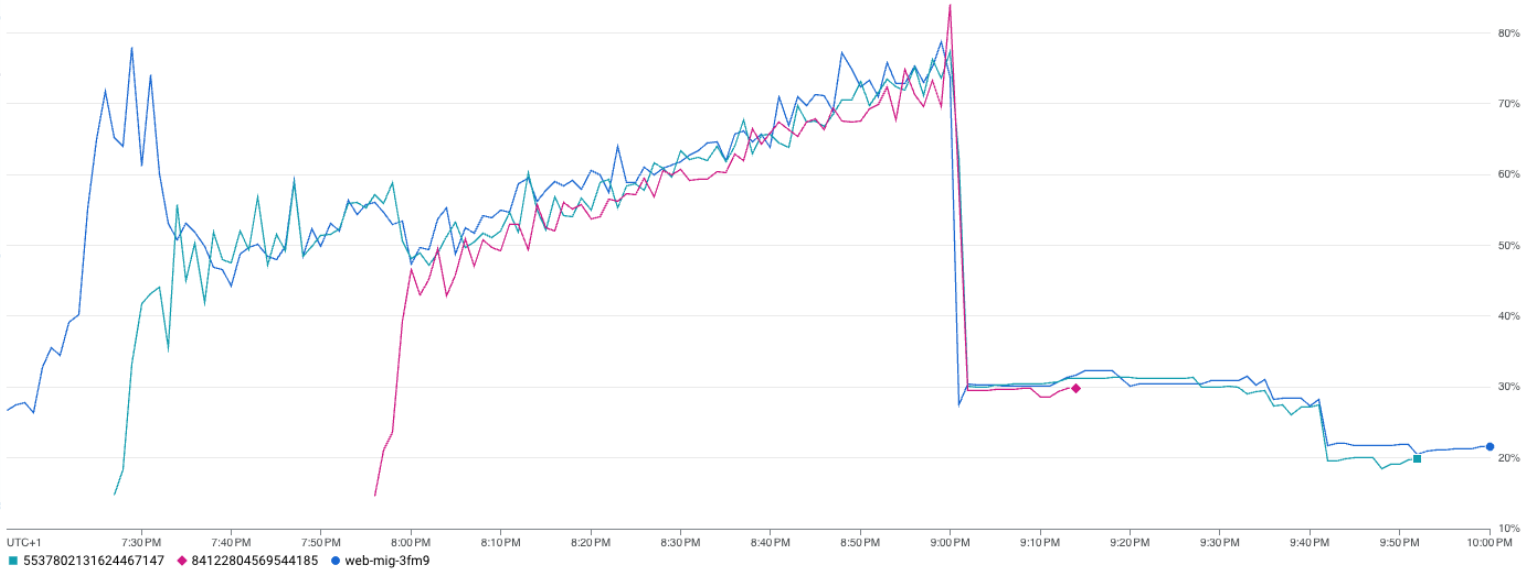


Autoescalamiento

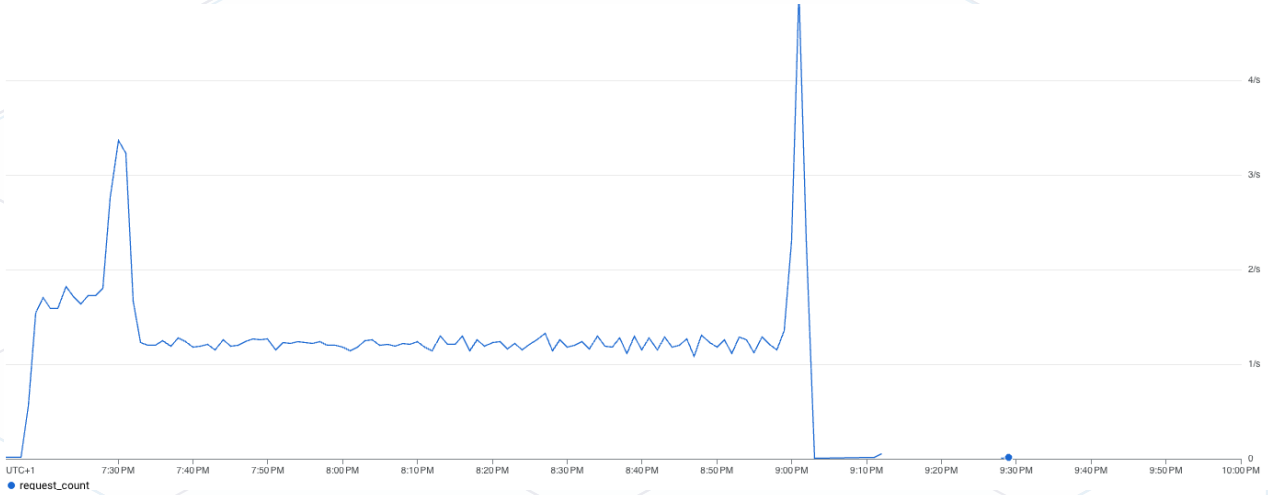


Cuando se supera el objetivo del 55% de uso de memoria, se crea una nueva instancia hasta un máximo de 3 instancias. Cuando se cae por debajo de nuevo, se elimina una instancia hasta el mínimo de una instancia.

Resultados - Recursos utilizados Servidor Web - Memoria



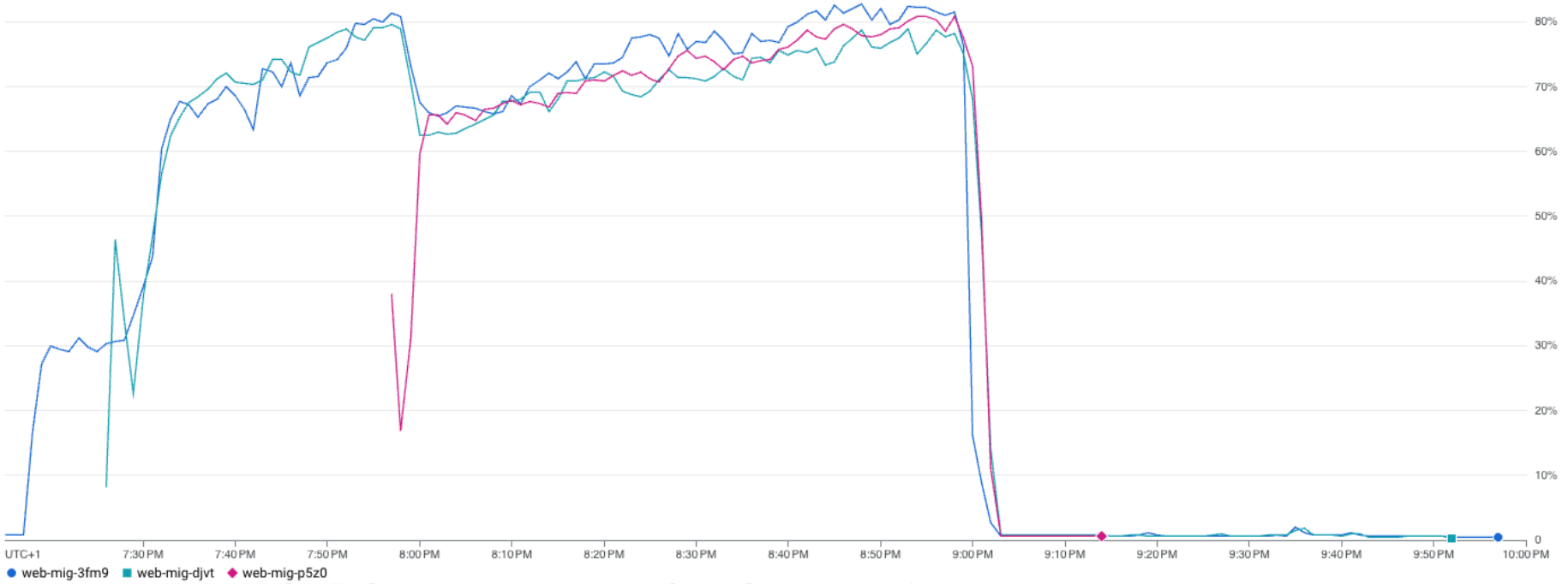
Resultados – Peticiones por segundo - Load Balancer



El throughput es estable y constante a lo largo del experimento. Agregar o eliminar instancias puede resultar en picos cortos, mientras el sistema se reequilibra.

Resultados - Recursos utilizados - Bucket Recibido

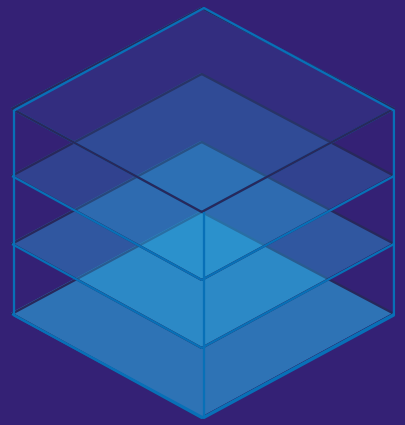
Resultados - Recursos utilizados CPU



El uso de CPU aumenta cuando el sistema escala de una instancia a dos instancias. Esto se debe a que el load balancer está configura de balancear por utilización con un objetivo de uso de RAM de 55%. Como el procesamiento de una petición tarda varios segundos y el uso de memoria no es constante durante este tiempo, con esta estrategia inteligente de balanceo se puede hacer un mejor uso de los recursos de las instancias.

Reflexiones y conclusiones

- El experimento evidenció la capacidad de escalabilidad del sistema al pasar de 1 a 200 usuarios concurrentes, en el experimento anterior sin escalamiento en la capa web el sistema colapsó con 40 usuarios concurrentes.
- El colapso total con 200 usuarios sugiere la necesidad de revisar y optimizar la capacidad de la aplicación o la infraestructura.
- A partir de 186 usuarios, se observó una degradación en el rendimiento, indicando posibles limitaciones en la capacidad del sistema.
- A pesar de lograr el throughput requerido, el tiempo de respuesta se degradó de forma linear con la carga subiendo.
- Con un throughput de 1.25 req/s, se recomienda evaluar y ajustar la capacidad del sistema para cumplir con los criterios de aceptación establecidos en términos de tiempo de respuesta y utilización de recursos, dado que el objetivo es procesar 100 peticiones por minuto, se observa una discrepancia significativa que indica la necesidad de ajustar la capacidad para alcanzar este requisito.
- La utilización de recursos, especialmente la CPU, alcanzó su pico de 80% durante la carga máxima.
- Las estrategias de escalamiento fueron efectivas para manejar la demanda, pero la degradación y el colapso sugieren posibles limitaciones de la aplicación o de la infraestructura.
- La capacidad de escalamiento se demostró efectiva hasta cierto punto, evidenciado por los eventos de escalamiento exitosos.



MISO

Maestría en Ingeniería de Software

Experimento 2 – Conversión de videos

Experimento 2

Análisis de desempeño del componente encargado de convertir videos, actualizar el estado en la base de datos y guardar el video convertido, de manera que el video quede listo para ser solicitado por el usuario.

• Configuración: Documento

Experimento 2

Métricas

Throughput: Cantidad de peticiones por minuto en la conversión de videos.

Tiempo de respuesta promedio: Tiempo promedio que tarda la aplicación en procesar las tareas de conversión**.**

Tiempo de respuesta (P95): Percentil 95% de tiempo máximo que tarda la aplicación en procesar una tarea.

Criterios de aceptación

Throughput: Capacidad de procesar 100 peticiones por minuto.

Tiempo de respuesta

- No debe superar los 0.5 segundos en el 95% de las transacciones, por petición.
- No debe superar los 4 segundos en el 99% de las transacciones.

Utilización de recursos: Durante las pruebas con 100 peticiones concurrentes, la CPU del servidor alcanza un pico de 80% y la memoria se mantendrá < 80% de uso.

Resultados

El experimento constó de 3 casos, que fueron ejecutados 5 veces cada uno:

- **Caso 1:** 5 peticiones concurrentes, 10 peticiones en total
- **Caso 2:** 10 peticiones concurrentes, 20 peticiones en total
- **Caso 3:** 20 peticiones concurrentes, 40 en total

Las métricas arrojadas en cada iteración se pueden observar en el siguiente link: Resultados escenario 2.xlsx

Igualmente, por cada iteración y caso se generaron 3 gráficas:

- Petición vs Tiempo de respuesta
- Uso CPU
- Uso RAM

Las gráficas respectivas de cada iteración y caso se pueden ver en el siguiente enlace: Escenario 2 con gráficas.docx

Resultados promedio – Caso 1

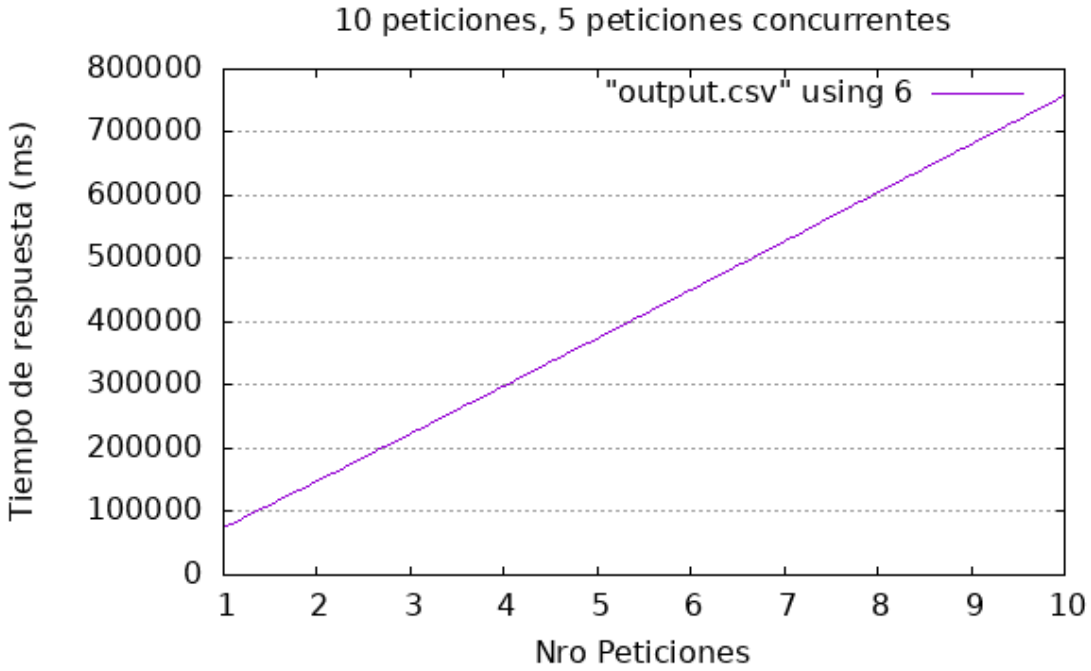
Tiempo respuesta promedio por petición (sec): 410,49342

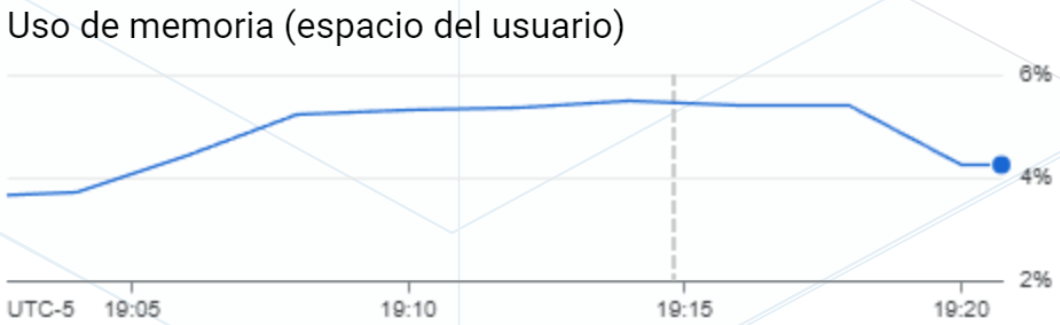
Tiempo respuesta P95 (sec): 713,68513

Peticiones por minuto (*Throughtput*): 0,802

Utilización máxima CPU: 88,51%

Utilización máxima RAM: 5,41%





Resultados promedio – Caso 2

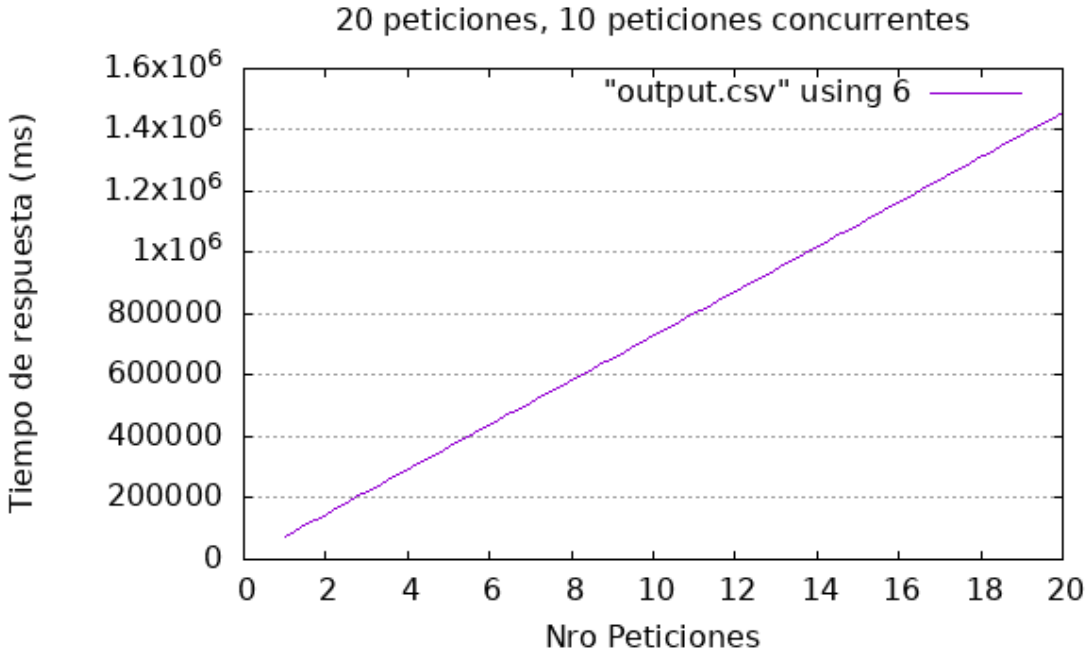
Tiempo respuesta promedio por petición (sec): 776,08173

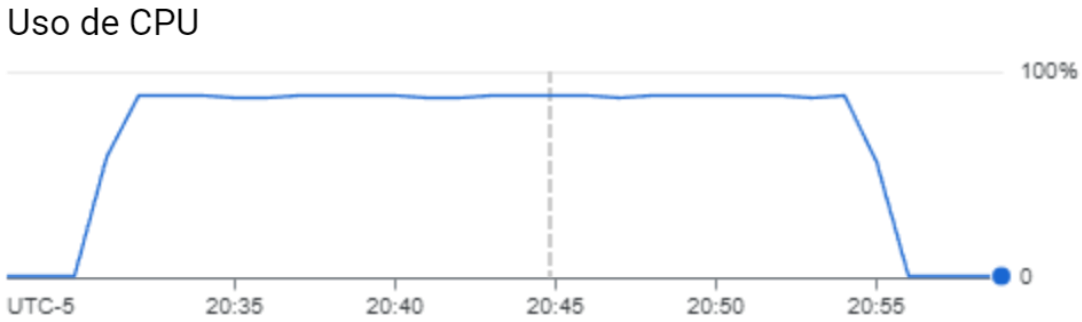
Tiempo respuesta P95 (sec): 1406,7539

Peticiones por minuto (*Throughtput*): 0,812

Utilización máxima CPU: 88,82%

Utilización máxima RAM: 5,54%





Resultados promedio – Caso 3

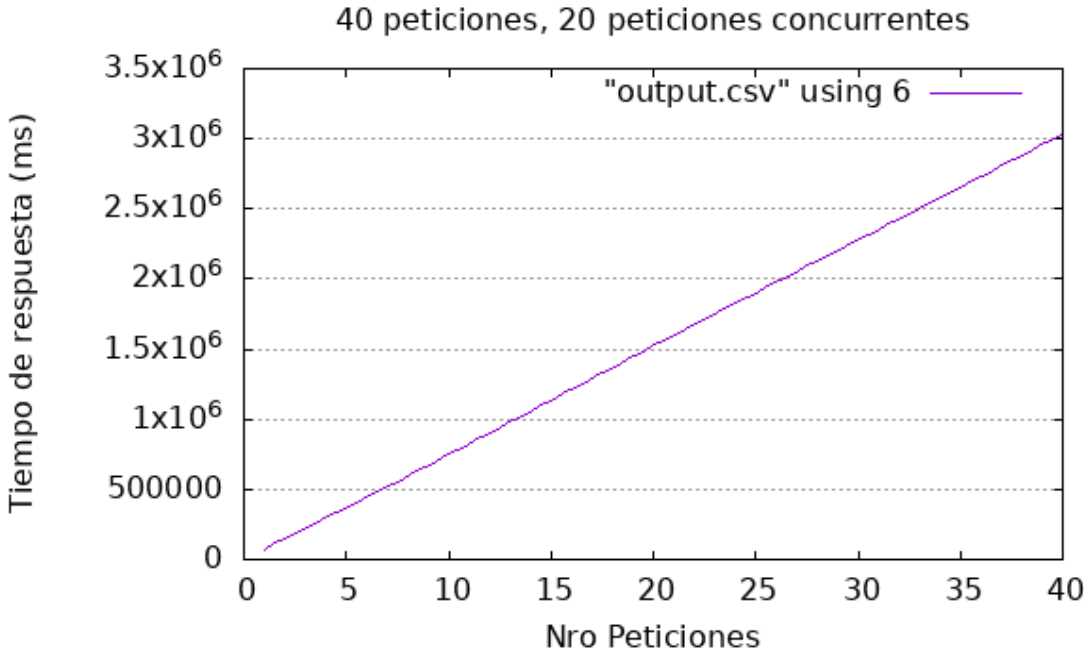
Tiempo respuesta promedio por petición (sec): 1549,6144

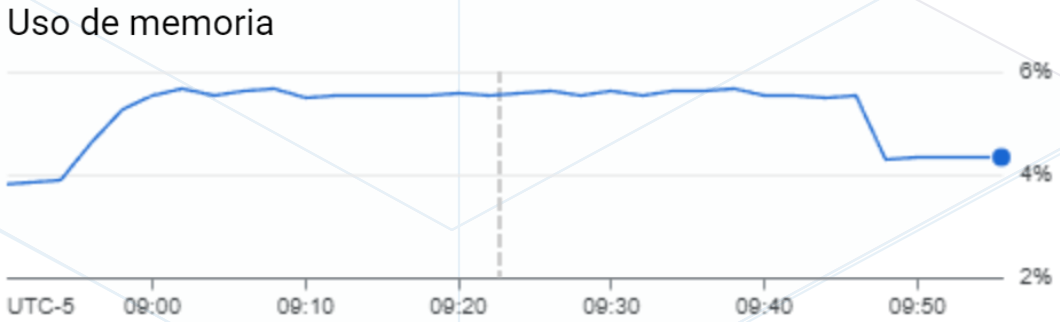
Tiempo respuesta P95 (sec): 2869,7137

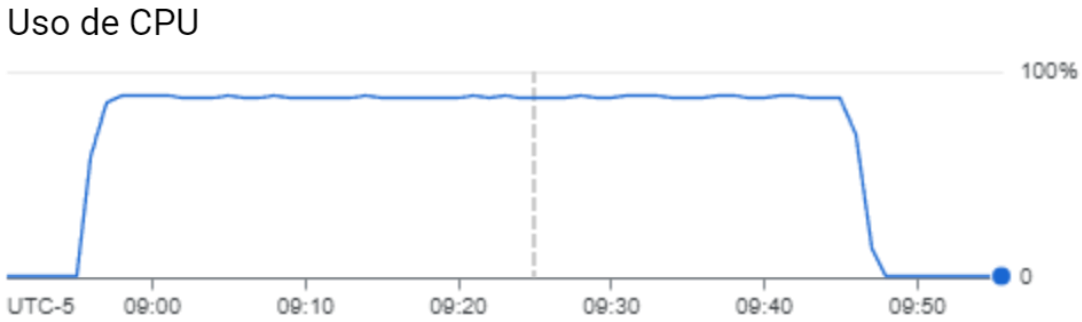
Peticiones por minuto (*Throughtput*): 0,794

Utilización máxima CPU: 88,64%

Utilización máxima RAM: 5,71%







Comparación con resultados anteriores

Caso	Métrica	Primera entrega	Segunda entrega
Caso 1	Tiempo respuesta promedio por petición (sec)	408,6708	410,49342
	Tiempo respuesta P95 (sec)	707,3242	713,68513
	Peticiones por minuto (*Throughtput*):	0,81	0,81
Caso 2	Tiempo respuesta promedio por petición (sec)	803,567	776,08173
	Tiempo respuesta P95 (sec)	1462,7432	1406,7539
	Peticiones por minuto (*Throughtput*):	0,798	0,812
Caso 2	Tiempo respuesta promedio por petición (sec)	1521,9354	1549,6144
	Tiempo respuesta P95 (sec)	2825,5324	2869,7137
	Peticiones por minuto (*Throughtput*):	0,82	0,794

Análisis de resultados

- Para todos los casos evaluados, **el uso de la memoria RAM cumple con el criterio de aceptación planteado** en la formulación del escenario (no exceder el 80% de su capacidad) ya que el pico máximo presentado en los 3 casos no fue superior al 6%, indicando que este no es un cuello de botella del sistema.

- **El uso de CPU se mantuvo alto en todos los casos (es decir, desde 5 usuarios concurrentes a 20 usuarios concurrentes)**, oscilando entre 88% y el 89% de su capacidad, por lo tanto, no se cumplió con el criterio de aceptación planteado en la formulación del escenario (no exceder el 80% de capacidad cuando haya 100 usuarios concurrentes)

- **En ninguno de los 3 casos se cumplió con el criterio de aceptación planteado para el tiempo de respuesta de cada petición** (menor o igual a 40 segundos), por el contrario, aumentó más a medida que se realizaban más peticiones. Este comportamiento se debe a que la cola de tareas se iba llenando de peticiones, resultando en un tiempo de espera mayor.

- **El número de peticiones atendidas por minuto se mantuvo constante entre 0.79 y 0.81** en cada uno de los tres casos, dando con el incumplimiento del criterio de aceptación planteado para esta métrica (mínimo 100 peticiones por minuto) y reflejando la limitación de procesamiento para manejar peticiones adicionales.

Reflexiones y conclusiones

- El componente de conversión de videos en su estado actual continúa siendo un punto crítico en el sistema, ya que la capacidad de procesamiento de la máquina virtual que lo aloja alcanzó niveles máximos de manera constante durante todo el experimento, independientemente del caso probado. Por tanto, es fundamental que se considere como un área de mejora para el futuro.

- Desde el inicio del experimento, los tiempos de respuesta de las solicitudes fueron altos. Esto se debe a que la cola de tareas recibe dichas peticiones de acuerdo con sus capacidades, lo que provoca un embotellamiento cada vez que se agregan más tareas a la cola.