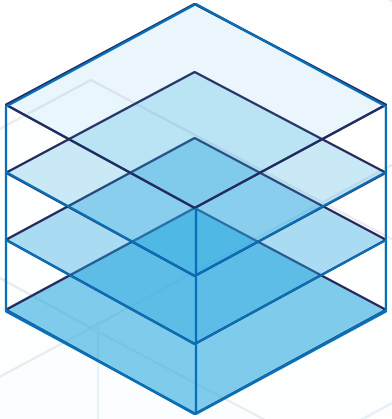


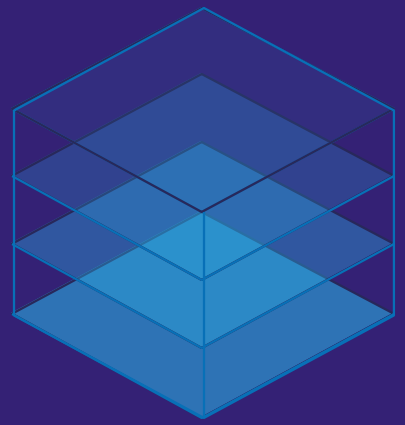
MISO

Maestría en Ingeniería de Software

Entrega 4 Sistema Conversión Cloud

Sistema de Conversión Cloud - Escalabilidad en el Backend

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

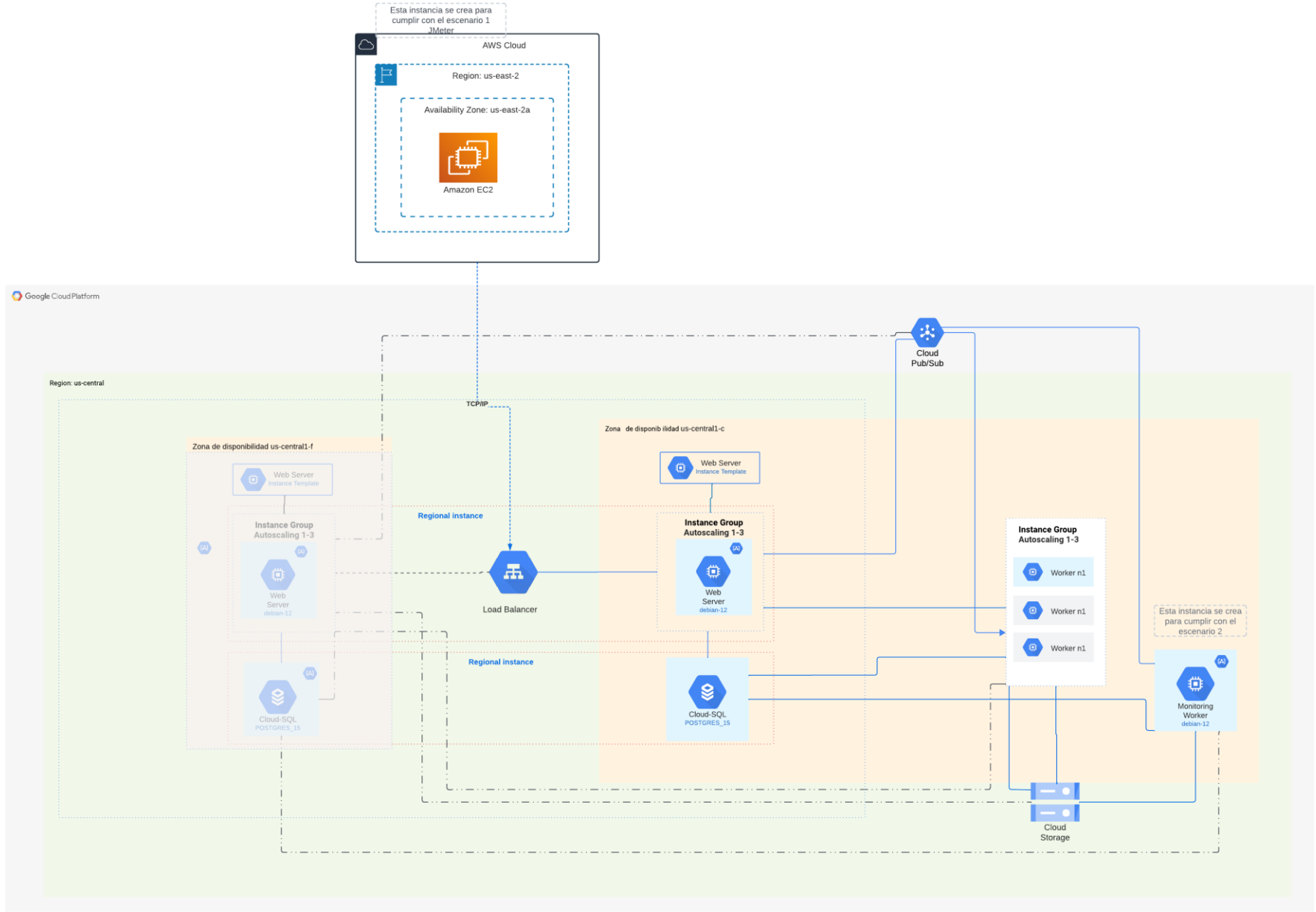


MISO

Maestría en Ingeniería de Software

Arquitectura

Arquitectura



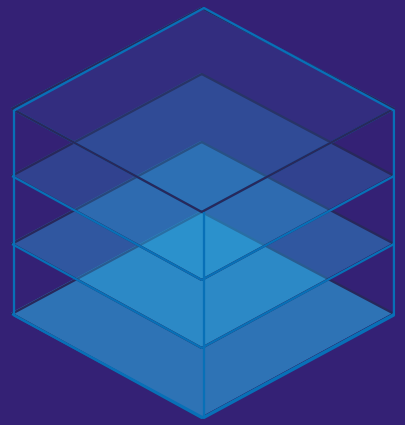
Configuración del sistema

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.
- Compute Engine: Se usan instancias n2-highcpu-2 para que ejecuta la conversión del video.
- 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. Trabaja en dos zonas de disponibilidad para garantizar una alta disponibilidad.
- 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 tanto para la capa web como para la capa worker. Las instancias del Managed Instance Group se basan en el Instance Template. Las instancias de la capa web están ubicadas en dos zonas de disponibilidad para garantizar una alta disponibilidad.
- Autoscaling: Se activa autoscaling para el Managed Instance group, así que crean/eliminan instancias de acuerdo con la carga. En el caso de la capa web, cuando se supera el objetivo del 60% de uso promedio de memoria o 80% de la CPU en caso de los workers, 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.
- Pub/Sub: Es un servicio de mensajería y publicación/suscripción en GCP. Puede estar involucrado en la comunicación entre componentes del sistema.
- 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 y alta disponibilidad 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:

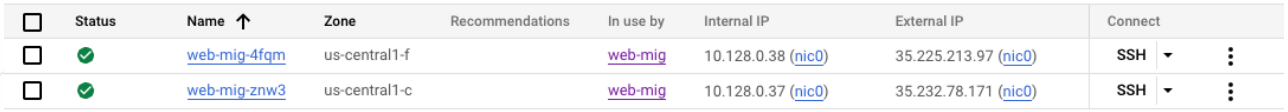
- El tiempo de respuesta de la aplicación en todos los escenarios de prueba no debe superar los 0.5 segundos en el 95% de las transacciones, por petición.
- El tiempo de respuesta en ningún escenario de prueba debe superar los 4 segundos en el 99% de las transacciones.

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 1:25pm con una instancia en cada zona de disponibilidad.



Status	Name	Zone	Recommendations	In use by	Internal IP	External IP	Connect
✓	web-mig-4fqm	us-central1-f		web-mig	10.128.0.38 (nic0)	35.225.213.97 (nic0)	SSH
✓	web-mig-znw3	us-central1-c		web-mig	10.128.0.37 (nic0)	35.232.78.171 (nic0)	SSH

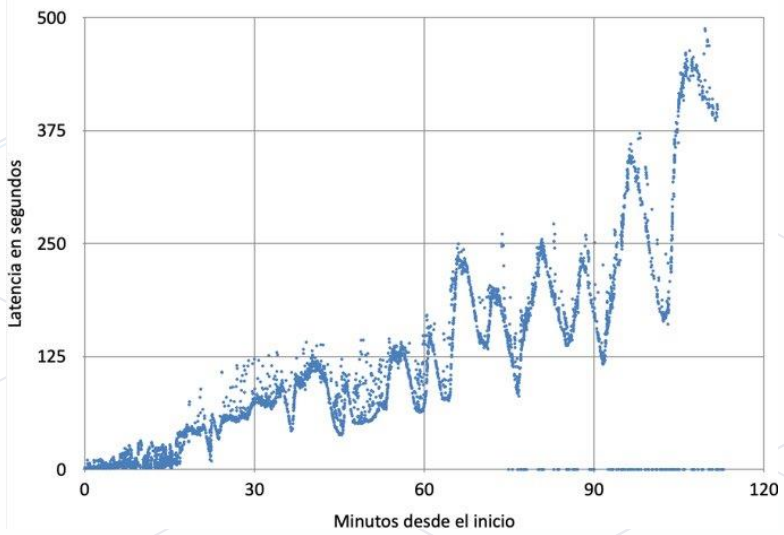
- **Escalamiento**
  - Primer Scale-out (2 instancia a 3 instancias con 2 instancias en zona us-central1-f y 1 instancia en us-central1-c) después de 52 minutos a las 2:17pm con 104 usuarios concurrentes. En el experimento pasado se observó el escalamiento de 2 instancias a 3 instancias con 96 usuarios concurrentes.

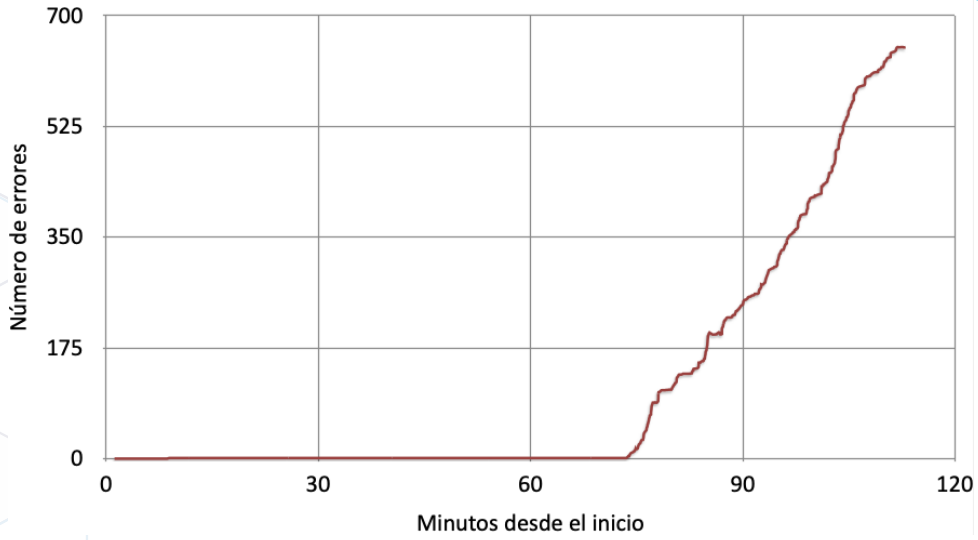
Status	Name	Zone	Recommendations	In use by	Internal IP	External IP	Connect
✓	web-mig-4fqm	us-central1-f		web-mig	10.128.0.38 (nic0)	35.225.213.97 (nic0)	SSH
✓	web-mig-b988	us-central1-f		web-mig	10.128.0.39 (nic0)	34.69.104.81 (nic0)	SSH
✓	web-mig-znw3	us-central1-c		web-mig	10.128.0.37 (nic0)	35.232.78.171 (nic0)	SSH

- Degradación: Se observó la generación de errores a partir de 146 usuarios concurrentes después a las 2:38pm.
- **Colapso Total:** El sistema colapsó completamente con 226 usuarios concurrentes a las 3:18pm.
- Fin de pruebas de estrés con JMeter a las 3:25pm.
- **Escalamiento Inverso (Scale-in):**
  - Primer Scale-in (3 instancias a 2 instancias, uno en cada zona) a las 3:42pm.

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.

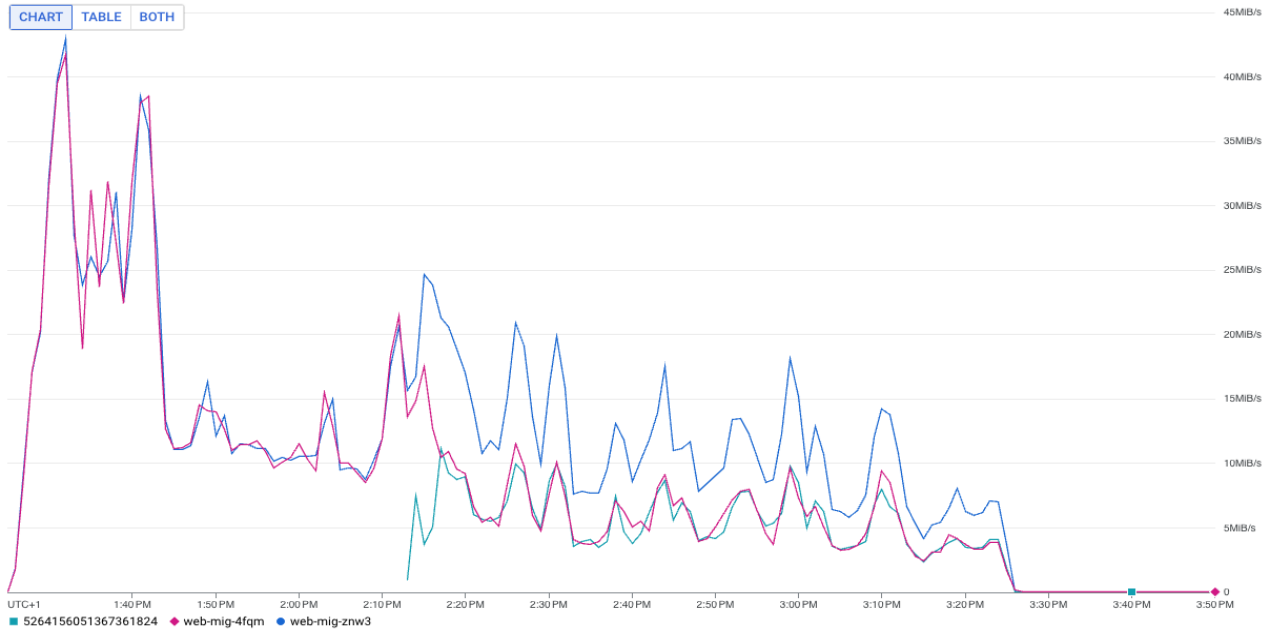
Resultados - Latencia



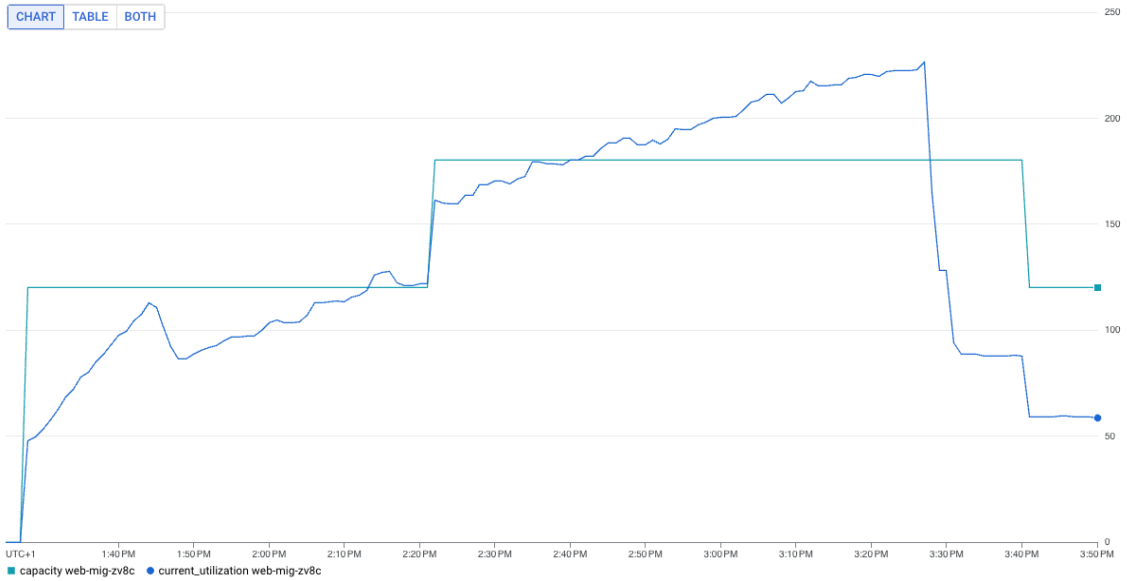


Resultados - Recursos utilizados Servidor Web - Red Recibido

La latencia promedio experimenta un incremento proporcional al aumento en el número de usuarios, es fundamentar resaltar que las instancias no experimentan una distribución equitativa de peticiones, esto se genera una latencia mayor en comparación con otras instancias, dado que una instancia puede recibir mas peticiones que las otras, esto aumenta la latencia debido a la carga de trabajo y afecta negativamente su capacidad de respuesta.

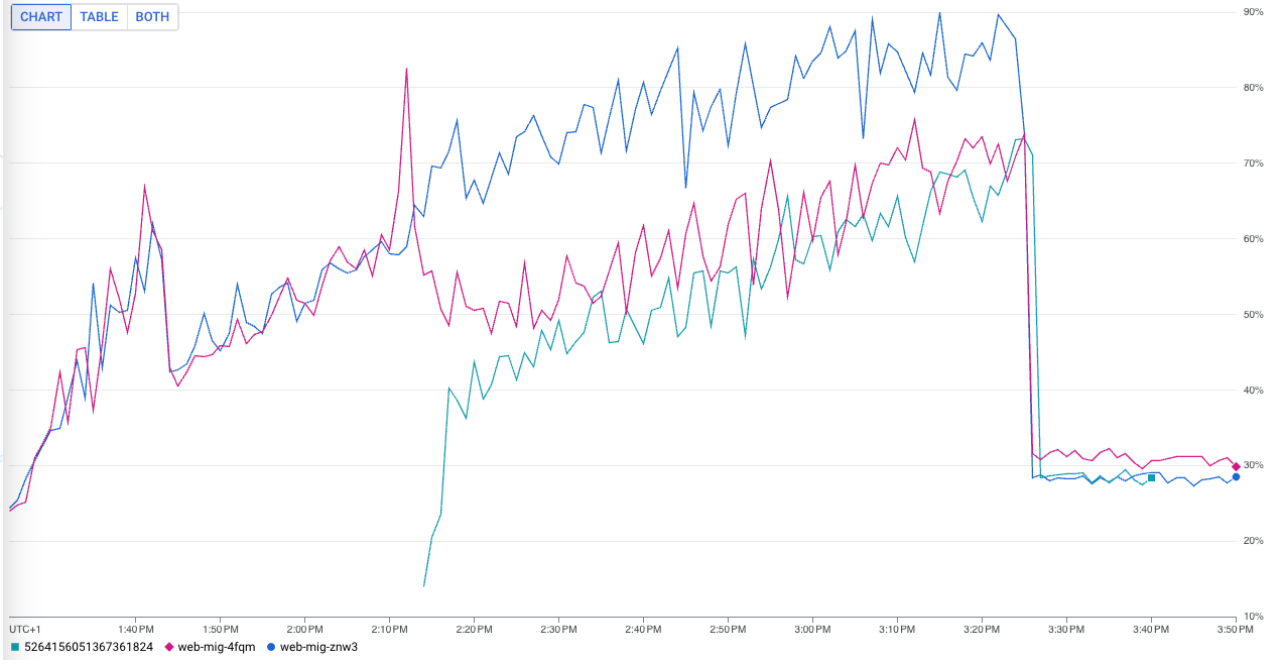


Autoescalamiento

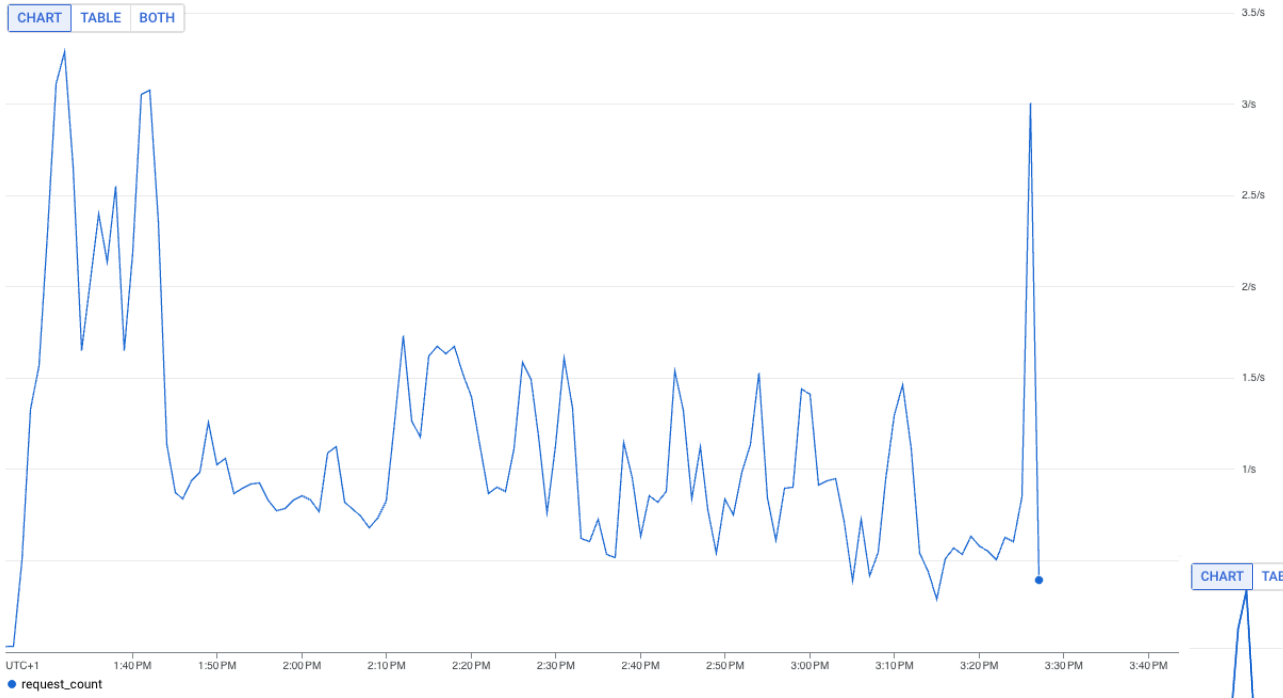


Se implementa la creación de nuevas instancias cuando se excede el umbral del 60% de uso de memoria. En caso de que el uso de memoria disminuya por debajo de este umbral, se procede a eliminar una instancia, manteniendo un mínimo de una instancia en funcionamiento en cada zona.

Resultados - Recursos utilizados Servidor Web - Memoria



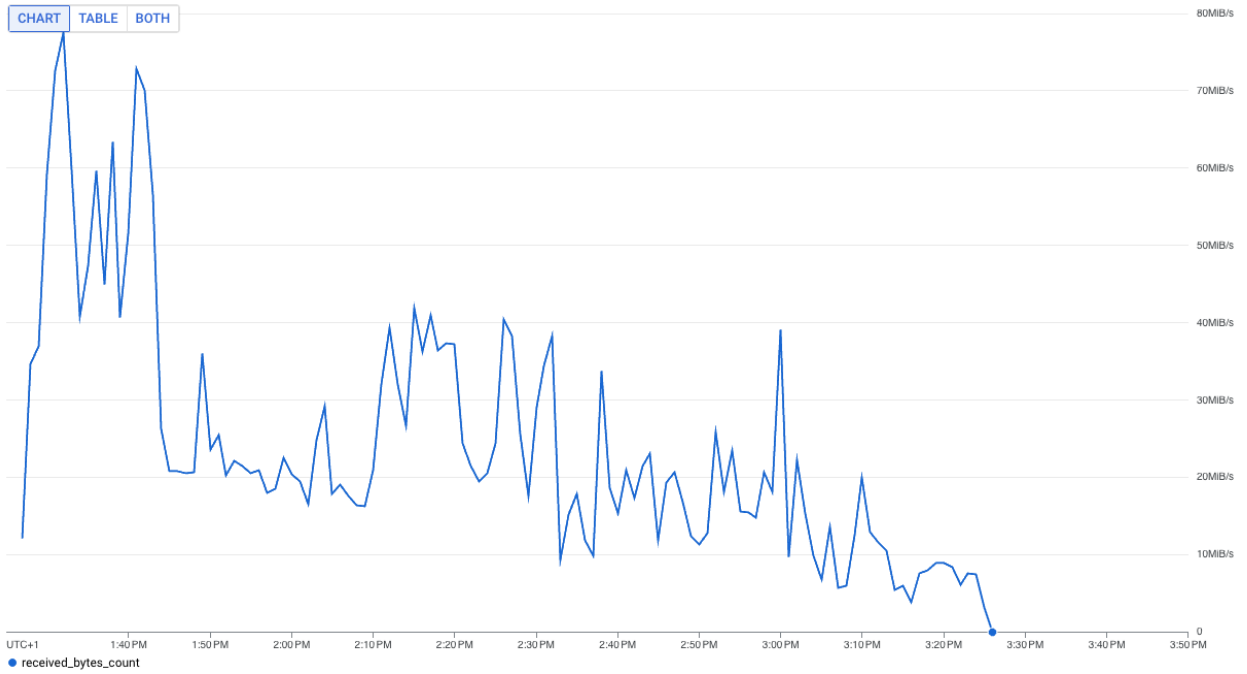
Resultados – Peticiones por segundo - Load Balancer



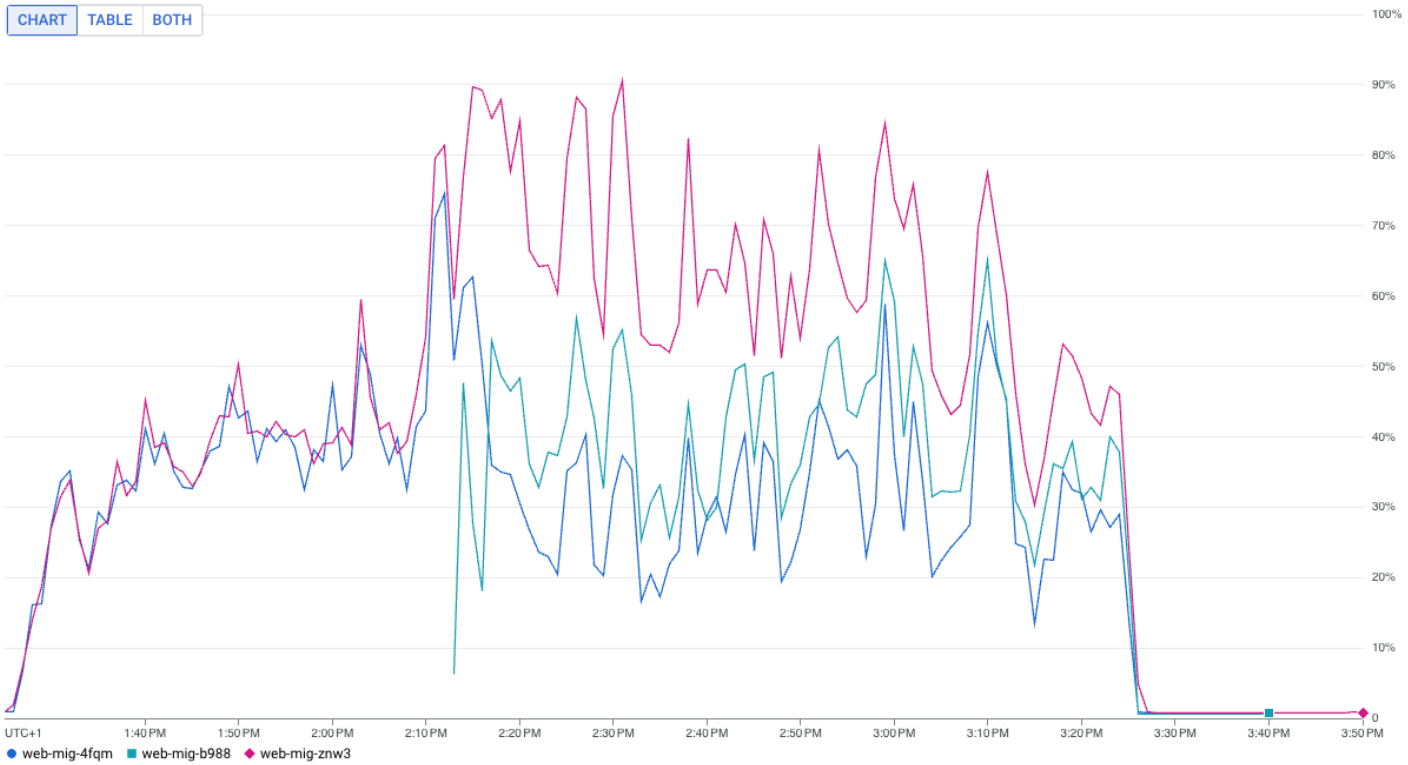
El rendimiento (throughput) se mantiene estable y constante a lo largo de todo el experimento. La adición o eliminación de instancias puede generar picos temporales, ya que el sistema se reajusta para restablecer el equilibrio.

La estrategia del balanceador de carga se orienta a minimizar la comunicación entre estas zonas. Este enfoque garantiza una asignación eficiente de recursos al adaptarse a la variabilidad en la demanda, optimizando así el rendimiento del sistema y mitigando la latencia asociada con la comunicación interzonal.

Bucket Recibido



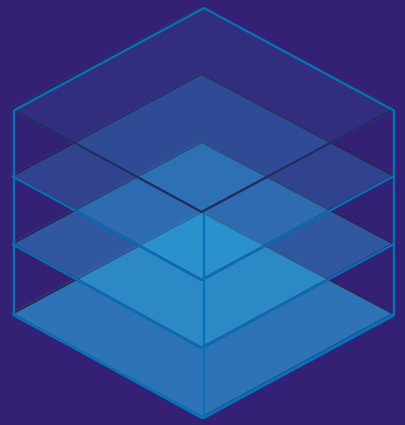
Resultados - Recursos utilizados CPU



El incremento en el uso de la CPU se observa al escalar el sistema. En el experimento pasado se observó el escalamiento de 2 instancias a 3 instancias con 96 usuarios concurrentes. Este fenómeno se atribuye a la configuración del balanceador de carga, el cual está ajustado para equilibrar según la utilización, con un objetivo del 55% de uso de RAM. La estrategia del balanceador de carga está diseñada para minimizar la comunicación entre estas zonas. Sin embargo, es esencial tener en cuenta que esta optimización puede llevar a que una instancia individual reciba más peticiones que las otras dos, lo que potencialmente genera errores en el sistema.

Reflexiones y conclusiones

- El experimento evidenció la capacidad de escalabilidad del sistema al pasar de 1 a 226 usuarios concurrentes, en el experimento anterior sin el uso de GCP PubSub el sistema colapsó con 200 usuarios concurrentes.
- La observación de la degradación a partir de 146 usuarios concurrentes y el colapso total a 226 usuarios subraya la necesidad de identificar y establecer umbrales críticos para gestionar la capacidad del sistema.
- La implementación del primer scale-out destaca la importancia de contar con mecanismos automáticos de escalabilidad para hacer frente a aumentos repentinos en la carga de usuarios.
- El escalamiento inverso demostró ser una táctica eficaz para estabilizar el sistema después de un pico de demanda, pero también resalta la importancia de optimizar la gestión de recursos durante períodos de baja carga.
- Las visualizaciones proporcionan información valiosa para afinar la configuración del sistema y mejorar su capacidad para manejar cargas pico.



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:

- El tiempo de respuesta no debe superar los 40 segundos.
- No debe superar los 120 segundos en el 95% de las transacciones, por petición.

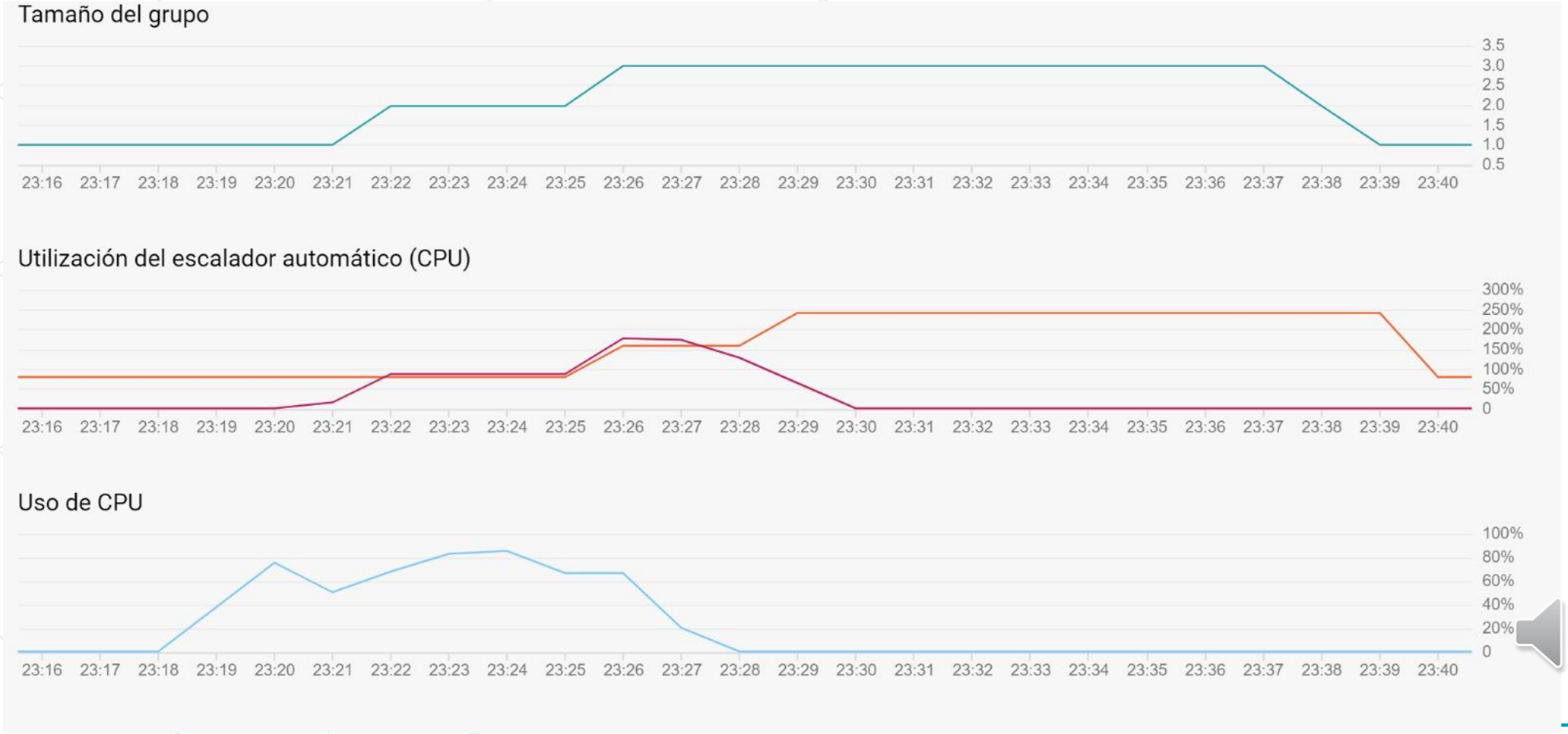
Utilización de recursos: Durante las pruebas con 100 peticiones concurrentes, la CPU del servidor alcanza un pico de 80% y la memoria se mantendrá < 80% de uso.

Métricas obtenidas

Datos	Caso 1	Caso 2	Caso 3	Caso 4	Caso 5
Peticiones concurrentes	5	10	20	40	80
Total peticiones	10	20	40	80	160
Tiempo respuesta promedio por petición (sec)	300,0921733	464,9051167	716,09512	1061,752083	2238,72931
Tiempo respuesta P95 (sec)	451,6734433	707,7307167	1161,92874	2139,21888	4066,3088
Peticiones por minuto (*Throughput*):	1,3	1,64	1,946666667	2,15	2,3

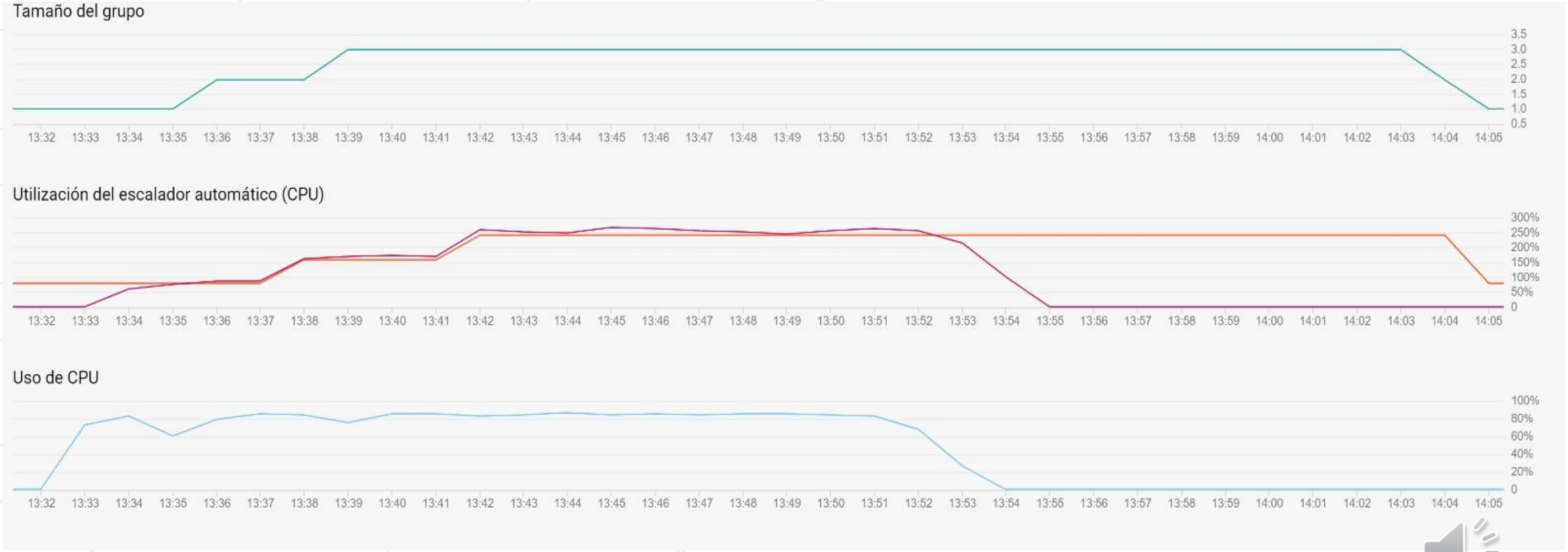
Revisar resultados obtenidos: Resultados

Resultados autoescalamiento – Caso 1

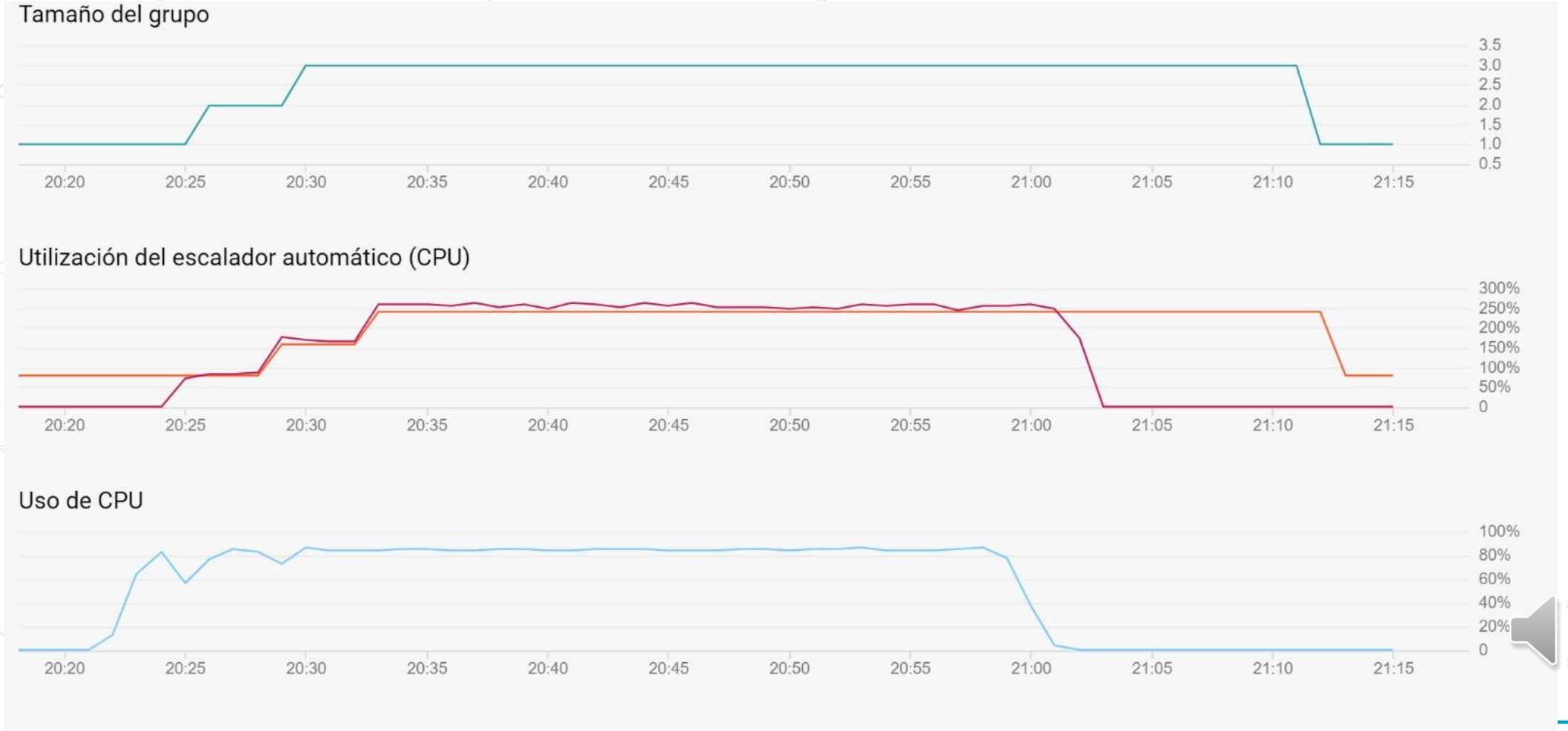


Resultados autoescalamiento – Caso 2

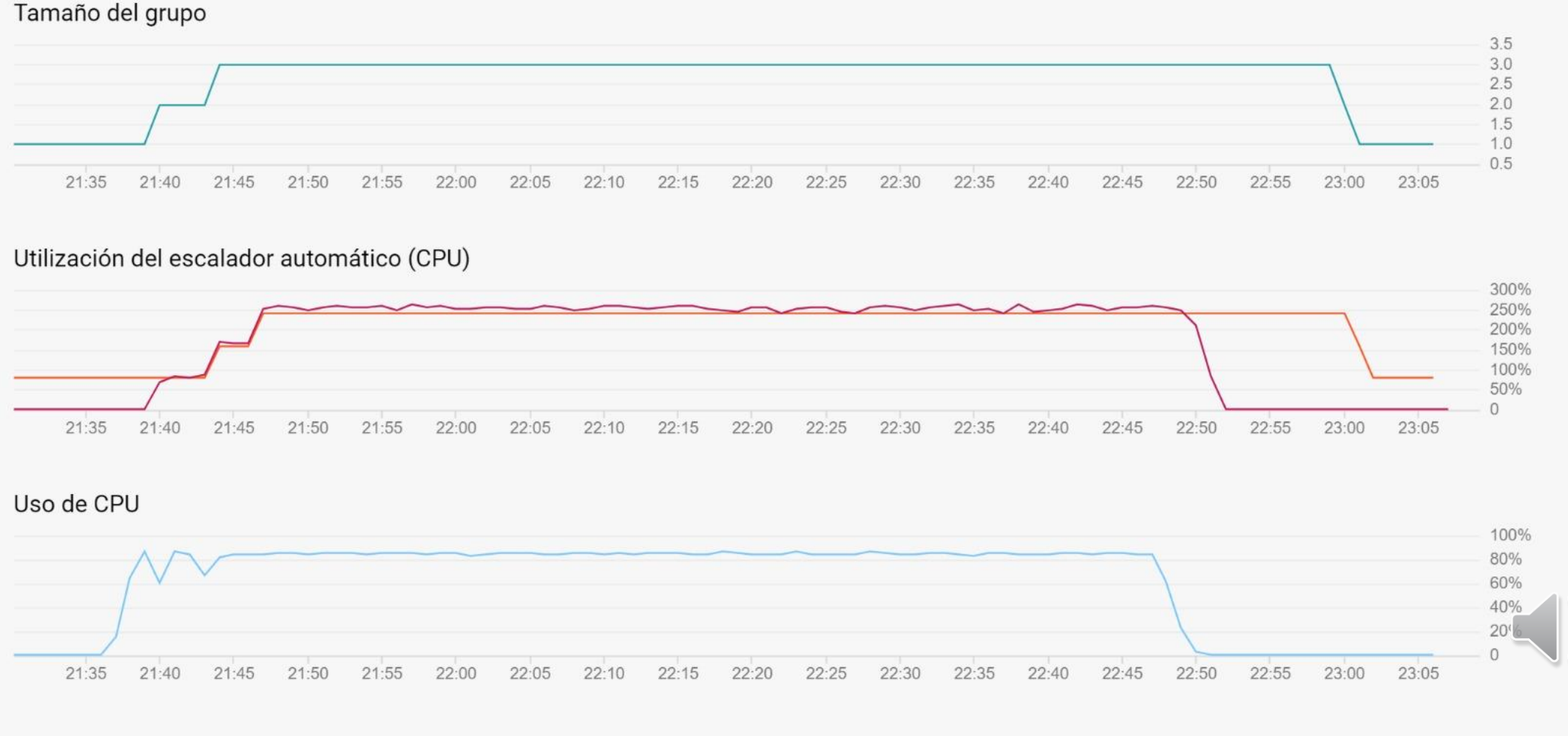
Resultados autoescalamiento – Caso 3



Resultados autoescalamiento – Caso 4



Resultados autoescalamiento – Caso 5



Comparación con resultados anteriores

Caso	Métrica	Primera entrega	Segunda entrega	Tercera Entrega
Caso 1	Tiempo respuesta promedio por petición (sec)	408,6708	410,49342	300,0921733
	Tiempo respuesta P95 (sec)	707,3242	713,68513	451,6734433
	Peticiones por minuto (*Throughtput*):	0,81	0,81	1,3
Caso 2	Tiempo respuesta promedio por petición (sec)	803,567	776,08173	464,9051167
	Tiempo respuesta P95 (sec)	1462,7432	1406,7539	707,7307167
	Peticiones por minuto (*Throughtput*):	0,798	0,812	1,64
Caso 2	Tiempo respuesta promedio por petición (sec)	1521,9354	1549,6144	716,09512
	Tiempo respuesta P95 (sec)	2825,5324	2869,7137	1161,92874
	Peticiones por minuto (*Throughtput*):	0,82	0,794	1,946666667

Análisis de resultados

- **La utilización máxima de CPU superó el 80% en todos los casos,** aunque se puede ver que la utilización varia del caso 1 al caso 5, pues en el caso uno el uso de CPU superior al 80% no es tanto como en los casos posteriores.

- **En ninguno de los 5 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, pero en definitiva disminuyó mucho esta métrica con respecto a las entregas anteriores.

- **El número de peticiones atendidas por minuto fue aumentando a medida que se agregaban más peticiones,** pasando de un constante 0.79 – 0.81, a 1.6 – 2.3, aunque dicho aumento fue decreciendo en los últimos casos, pues del caso 1 al 3 creció en 0.3 aproximadamente, pero del caso 3 al 4 fue de 0.2 y del caso 4 al 5 de 0.15, por lo que seguramente llegará aún límite máximo menor a 3 peticiones por minuto, lo que aún está lejos de las 100 peticiones por minuto.

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 todas las instancias alcanzó niveles superiores al 80% de manera constante durante todo el experimento, independientemente del caso probado, por lo que 3 instancias todavía siguen siendo pocas para alcanzar las metas prometidas

- De igual forma agregar más de una instancia fue una mejora positiva para el worker y los tiempos de respuesta disminuyeron considerablemente.