Análisis de capacidad

Entrega 3 – Desarrollo de Software en la Nube

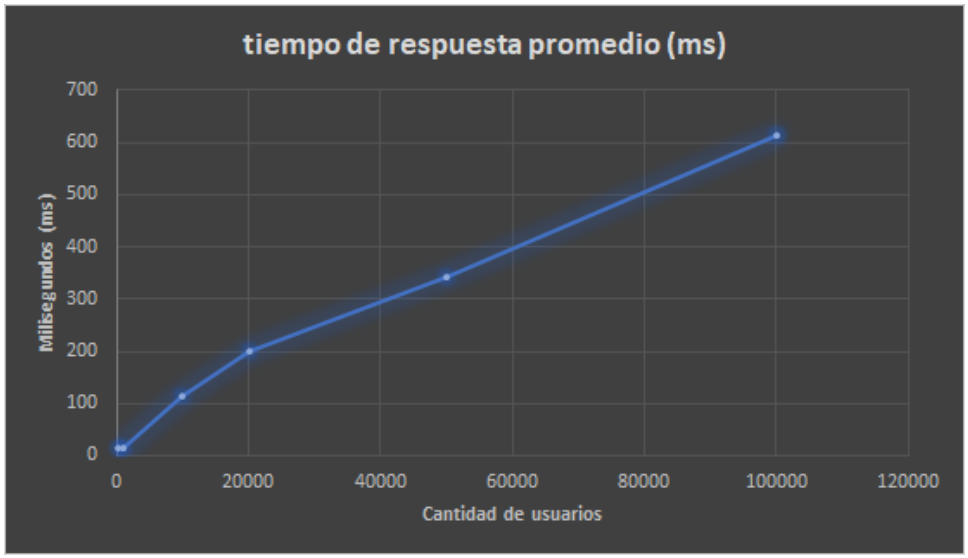
Diego Alejandro Camelo Giraldo, Juan Sebastián Alegría Zúñiga, María Camila Gómez Hernández, Andrés Felipe Lugo Saavedra

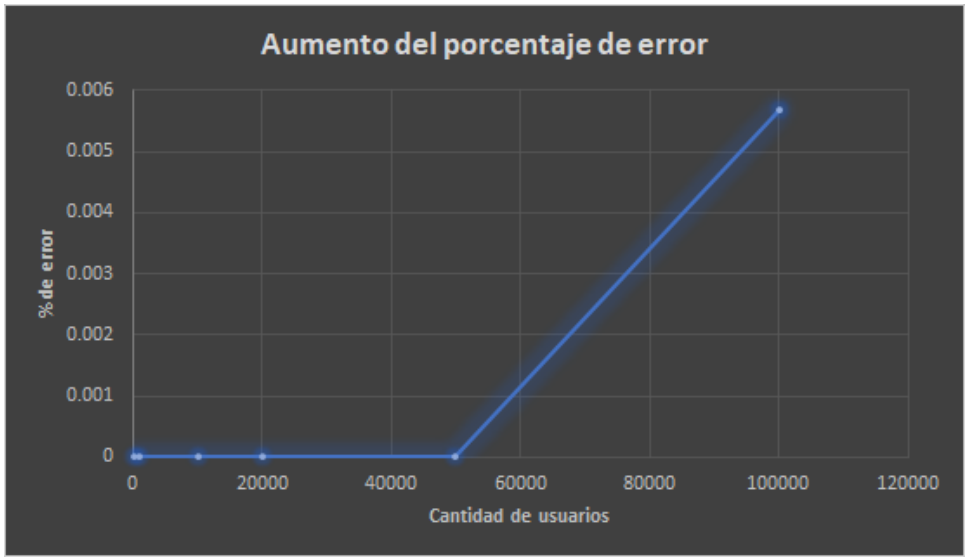
Grupo 6

Nota: Antes de ejecutar las pruebas de estrés, ajuste el documento del plan de análisis de capacidad con base a la retroalimentación realizada por los tutores del curso. 1. Pruebas de estrés sobre el Modelo de Despliegue.

Escenario 1. Se deberá definir un escenario donde se pueda probar cuál es la máxima cantidad de peticiones HTTP por minuto que soporta el API con 30 archivos publicados. Para hacer pruebas de estrés se sugiere utilizar la herramienta Apache Bench (ab) o JMeter que debe instalarse en el ambiente de nube (para no sesgar los resultados de la prueba con la latencia que introduce la red, la máquina de pruebas debe ubicarse en el mismo segmento de red de la aplicación). El escenario y los resultados de las pruebas de estrés deberán ser documentados con gráficas que ilustran cómo se comporta el sistema a medida que el número de usuarios virtuales accediendo a la aplicación se incrementa hasta llegar al punto de degradar completamente el rendimiento del sistema.

Restricciones del escenario. En las pruebas de stress el tiempo de respuesta promedio de la página debe ser de máximo 1.500 ms, si este tiempo no se cumple, se concluye que el sistema NO soporta la cantidad de peticiones de la prueba. En caso de que durante una prueba se generen más de un 1% de errores en las peticiones de la prueba, se concluye que el sistema NO soporta la cantidad de peticiones.

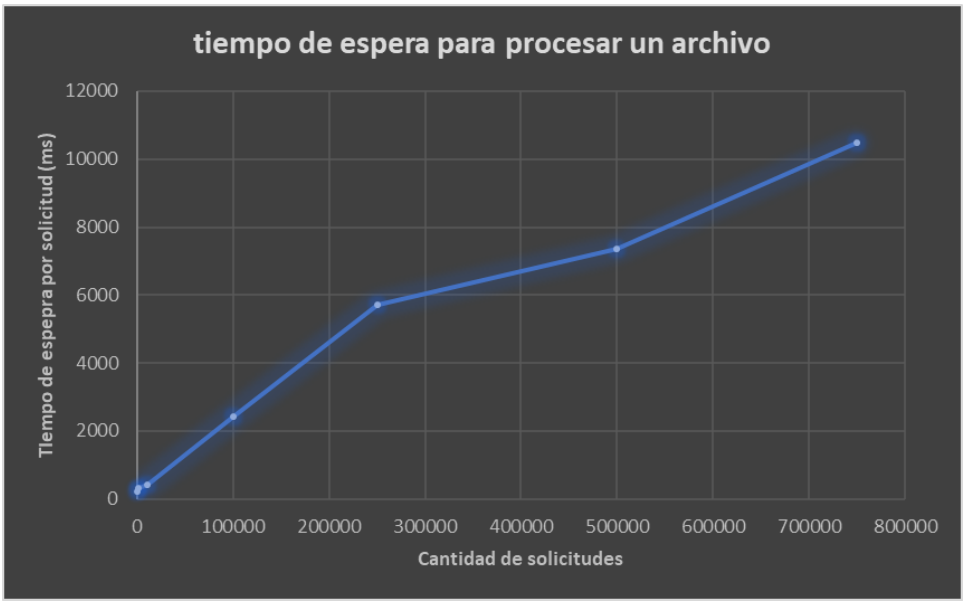




En este escenario, llevamos a cabo una prueba de carga en la que fuimos aumentando gradualmente el número de usuarios que enviaban solicitudes, mientras registramos los tiempos de respuesta esperados y el porcentaje de error en cada solicitud. Durante toda la prueba, vimos que el comportamiento de la prueba se mantenía muy estable conforme al aumento de solicitudes, dando como resultado un aumento lineal en el tiempo de respuesta por solicitud. Esto muestra una gran mejoria con respecto a los resultados de las pruebas anteriores donde este escenario mostraba un comportamiento exponencial y un límite en donde claramente el tiempo de espera crecía más allá de lo esperado. Estos resultados se vieron constantes hasta llegar a las 100000 solicitudes por minuto, en donde los resultados

seguían sin romper ninguna de las 2 limitaciones del caso, aunque una pequeña parte de las solicitudes presenta errores. Al tratar de probar más allá de este número de solicitudes, el programa en la máquina virtual empezó a congelarse por lo que fue imposible seguir con las pruebas, sin embargo, consideramos que los resultados son más que satisfactorios para probar el rendimiento actual del programa.

Escenario 2. Se deberá definir un escenario donde se pueda probar cuál es la máxima cantidad de archivos que pueden ser procesados por minuto en la aplicación. Para hacer pruebas de estrés se debe utilizar la herramienta Apache Bench (ab) o JMeter que debe instalarse en el ambiente de nube (para no sesgar los resultados de la prueba con la latencia que introduce la red, la máquina de pruebas debe ubicarse en el mismo segmento de red de la aplicación). Las pruebas de estrés deberán realizarse desde otros equipos diferentes a los utilizados para ejecutar el servidor web y el servidor de base de datos. El escenario y los resultados de las pruebas de estrés deberán ser documentados con gráficas que ilustran cómo se comporta el sistema a medida que el número de usuarios convirtiendo archivos se incrementa, hasta llegar al punto en que el tiempo para iniciar el procesamiento de un archivo enviado por un usuario supere los 10 minutos (600 segundos).



En este escenario podemos ver como el cambio en el tiempo de espera entre solicitudes es casi imperceptible antes de los primeros 100000 archivos en lista para procesar. Luego de estos el tiempo de espera parece crecer de forma lineal y los tiempos de espera empiezan a superar los 2 minutos por archivo, lo cual nos puede mostrar un cuello de botella en donde la aplicación empieza a sufrir por la cantidad de solicitudes simultaneas. Luego, entre las 250000 y las 500000 su crecimiento parece ser menor, lo cual puede deberse al auto escalado de las máquinas virtuales. Finalmente, el crecimiento en el tiempo entre solicitudes vuelve a crecer entre las 500000 y las 750000 solicitudes, mostrandonos el cuello de botella tras el escalado y que, como se ve en las gráficas, logro superar la barrera de los 10 minutos por solicitud. Se tiene que tener en cuenta 2 factores importantes para entender mejor los resultados conseguidos. El primero es que la falta de recursos en la máquina virtual pudo tener que ver con el comportamiento de las pruebas, pues a veces jmeter puede dar ciertos fallos. Lo segundo es que no se superó el 2% de errores en ninguno de los casos anteriores, mostrando que todas las solicitudes se atendían sin importar que tanto se esperaban.

Cuellos de botella:

Al probar los servicios de la aplicación con la herramienta jmeter, logramos descubrir la cantidad de solicitudes que estas son capaces de soportar antes de empezar a presentar fallas y que el rendimiento de la aplicación comience a fallar.

Servicio de Sign-Up: Soporta poco más de 100000 solicitudes en un minuto, con un rendimiento aproximado de 671 ms por solicitud. El cuello de botella no es muy claro en este servicio, no obstante, se puede suponer que luego de las 2000000 empiezan a existir problemas para atender y aparecen errores del 0.65%.

Servicio de Login: Soporta un aproximado de 95000 solicitudes en un minuto, con un rendimiento aproximado de 751 ms por solicitud. El cuello de botella comienza cerca de las 100000 solicitudes subiendo los tiempos de respuesta a 1147 ms y creciendo los porcentajes de error a 0,87 %

Servicio de carga de archivos: Soporta un aproximado de 100000 solicitudes en un minuto, con un rendimiento aproximado de 225 ms por solicitud. El cuello de botella en esta situación comienza cuando se sobrepasa el numero anterior, no obstante, el auto escalado permite que se regule por un periodo de tiempo, hasta que se sobrepasan las 500000 solicitudes y el cuello de botella vuelve a aparecer. Porcentaje de errores entre 0.81% y el 1.74%

Servicio de acceso a archivos: Soporta un aproximado de 100000 solicitudes en un minuto, con un rendimiento aproximado de 611 ms por solicitud. El cuello de botella comienza cerca de las 150000 solicitudes subiendo los tiempos de respuesta a 1233 ms y creciendo los porcentajes de error a 1,05 %

Conclusiones:

Estas pruebas de carga muestran que la aplicación está en capacidad de recibir cientos de miles de solicitudes sin sufrir en el proceso de atenderlas constantemente. Lo cual demuestra

lo efectivo que es la aplicación del auto escalado para situaciones de alta demanda en recepcion de multiples solicitudes.