Pruebas de Estrés en Sistemas Informáticos: Metodología y Aplicación en Entornos Virtuales de la UNAP

Sebastian Alberto Tapia Tito[1] and Gilmar Gutierrez Flores[2]

[1]Universidad Nacional del Altiplano

Octubre 2024

Abstract

Stress testing is important for measuring the robustness of a computer system under heavy load and traffic. With these tests, you will be able to identify errors, gaps, and improve the performance of the application. In this study, Apache JMeter was used to test different loads in a UNAP-like class, while Clumsy software was used to report packet loss and measure physical performance. The results provide important information about the system's ability to handle unexpected and damaging events on the network.

1 Introducción

Los sistemas informáticos, ya sean aplicaciones web o infraestructura empresarial, deben poder soportar diversas tensiones en situaciones del mundo real. Las pruebas de estrés simulan situaciones en las que el sistema se ve sometido a cargas que superan los umbrales de uso, con el objetivo de detectar posibles errores o cuellos de botella. El objetivo es garantizar que el sistema pueda utilizarse en periodos de intenso tráfico o alta demanda, con un mínimo de controles y restricciones.

2 Objetivos de las pruebas de estrés

Los principales objetivos de las pruebas de estrés son:.

- **Mostrar límites de trabajos**: Define los límites de rendimiento del programa según la cantidad de usuarios o trabajos.
- **Análisis de condición**: Analizar cómo se comporta un sistema bajo carga excesiva, buscando posibles fallas, degradación del rendimiento o fallas.
- **Identificar brechas**: Encuentre componentes críticos que no soportan suficiente carga y pueden requerir optimización o dimensionamiento.
- **Ver pruebas fallidas**: Asegúrese de que las fallas del sistema estén planificadas y no puedan causar efectos dañinos o pérdida de datos críticos.

3 Tipos de pruebas de estrés

Hay varios tipos de pruebas de estrés, cada uno diseñado para evaluar diferentes aspectos del sistema.

- **Prueba de carga máxima**: Mide la carga mínima a máxima. Carga esperada o excedida.
- **Prueba a largo plazo**: evalúa el comportamiento del sistema bajo una carga sostenida durante un largo tiempo.
- **Prueba de sobretensión**: mide los aumentos repentinos de tráfico para ver qué tan rápido se adapta el sistema a estos cambios.
- **Pruebas de estrés distribuidas**: Un análisis integral del comportamiento del sistema mediante la integración de múltiples componentes en una arquitectura distribuida.

.

4 Materiales para una pruebas de estrés

Una de las herramientas utilizadas para las pruebas de estrés es:

- **JMeter**: Varias pruebas de estrés de plataforma y carga de bases de datos y aplicaciones web. Capítulo
- **LoadRunner**: Herramienta para pruebas de carga y rendimiento.
- **Gadling**: Una solución moderna para aplicaciones web capaces de realizar simulaciones avanzadas.
- **Clumsy**: Para simular condiciones adversas en la red

.

beamer

5 Métodos

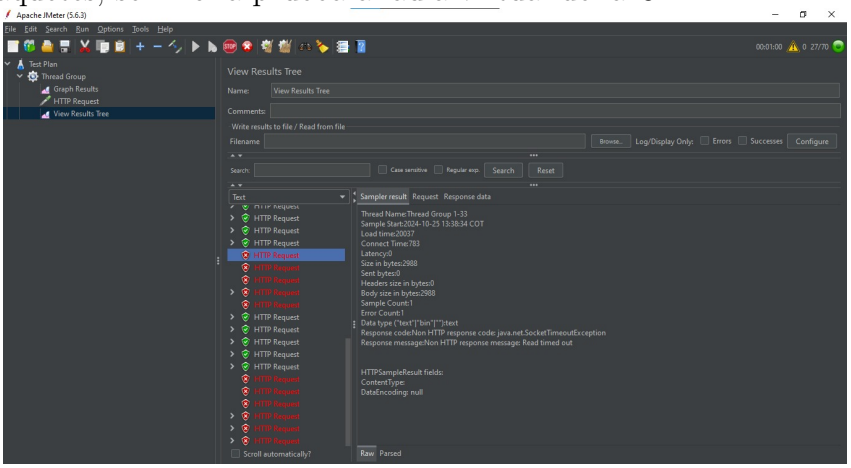
Se realizó una prueba de estrés usando Apache JMeter estableciendo el orden para número de subprocesos (usuarios simultáneos) y ajustando la carga para [tiempo de prueba]. Las métricas recopiladas incluyen el tiempo de respuesta, la tasa de error y la utilización de recursos.

Además se usó Clumsy para ajustar los parámetros de la red estableciendo parámetros como latencia, pérdida de paquetes y cambios de ancho de banda. Esto permite monitorear el sistema operativo según la conexión especificada.

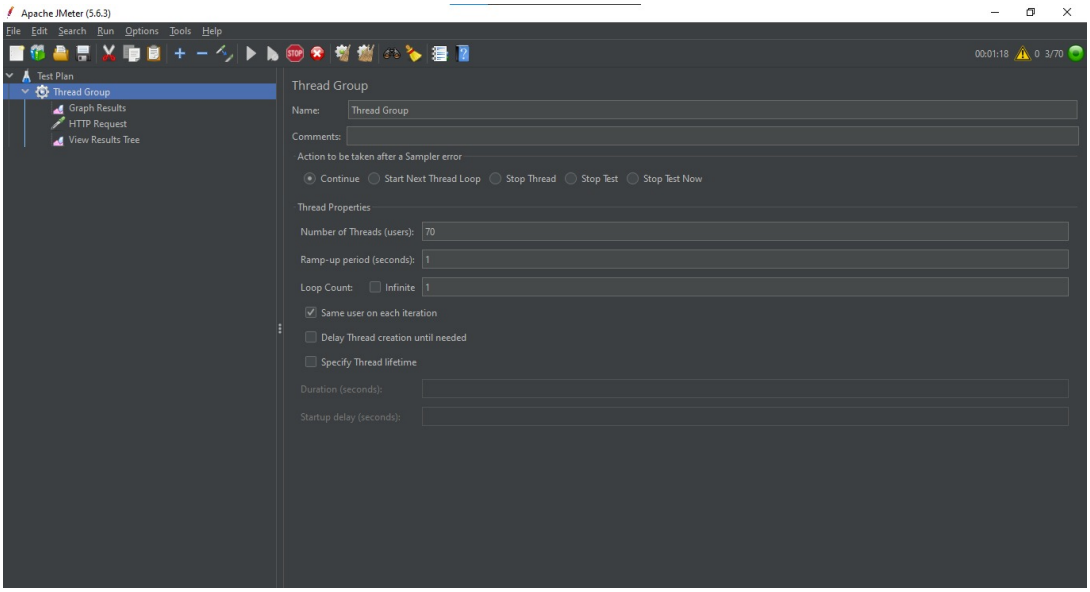
Los resultados se analizan para identificar puntos de error y estimar los parámetros del sistema.

Ejemplo Practico

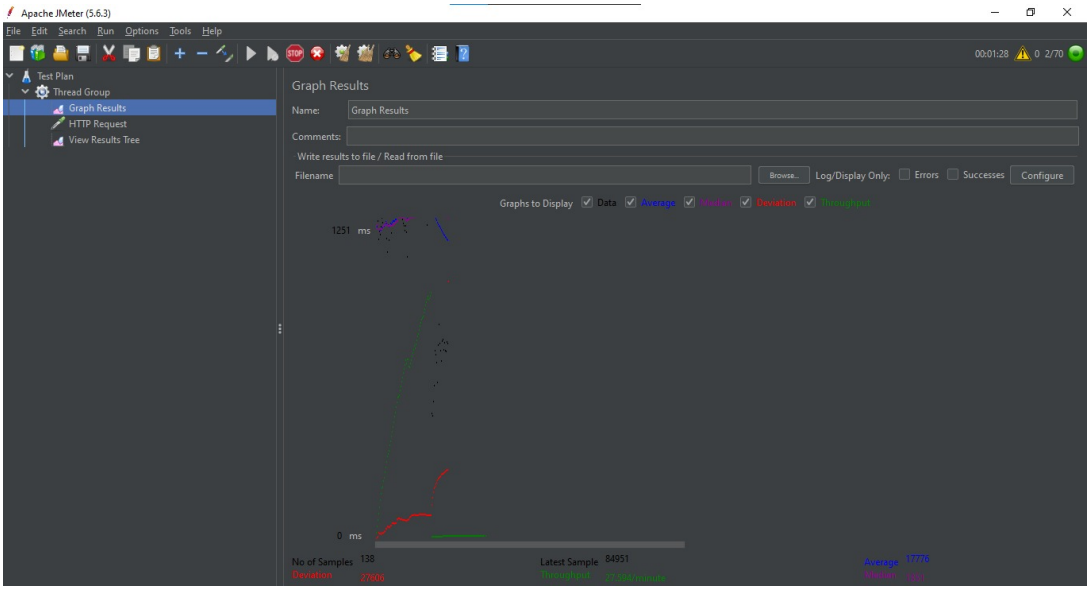
Se utilizó JMeter para simular las máquinas y el software Clumsy para simular la pérdida de paquetes, se hizo la prueba al aula virtual de la UNAP.



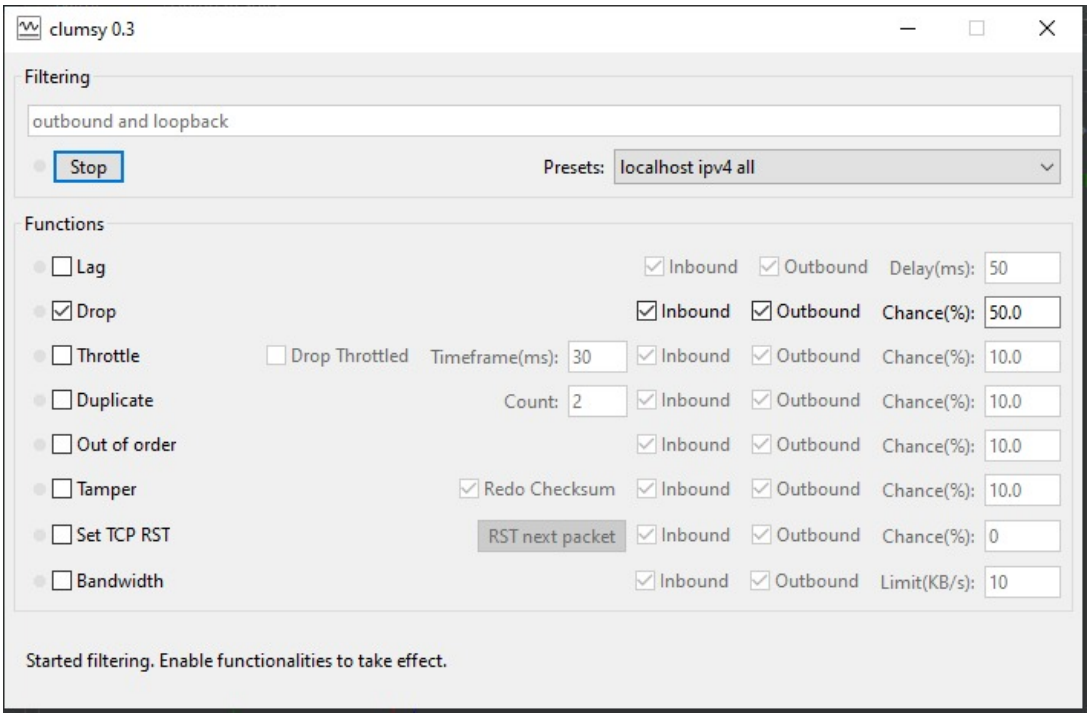
Fallos a la conexión por pérdida de paquetes



Configuración de la simulación



Gráfico



Parámetros del software Clumsy

6 Beneficios

El análisis de pruebas de estrés tiene numerosos beneficios:

- **Resiliencia mejorada**: El sistema es más capaz de responder a cargas inesperadas al encontrar y corregir deficiencias.
- **Mayor confiabilidad**: Exponer los sistemas a condiciones adversas les ayuda a funcionar correctamente en situaciones críticas.
- **Optimización de recursos**: Estas pruebas pueden revelar puntos débiles y permitir ajustar el código o componentes para mejorar la escalabilidad.

7 Conclusión

Las pruebas de estrés son una parte importante del ciclo de desarrollo de software, ya que garantizan que el sistema pueda sobrevivir a los peores escenarios. La detección temprana de problemas de rendimiento permite a los desarrolladores de software y administradores de sistemas mejorar los sistemas operativos en términos de experiencia del usuario final y también mejorar la estabilidad de los entornos de producción. Las pruebas de estrés realizadas con herramientas probadas aumentan la confiabilidad y estabilidad del software. Este es un hecho importante a considerar si se imponen altas exigencias al sistema en cuestión. .

References

Aggarwal, A., & Singh, V. (2024). Migration aspects from monolith to distributed systems using software code build and deployment time and latency perspective [Cited by: 0]. *Telkomnika (Telecommunication Computing Electronics and Control)*, *22*(4), 854–860. https://doi.org/10.12928/TELKOMNIKA.v22i4.25655

Aguilar Castro, J. L., & Kloul, L. (1997). Study of the task assignment problem in the distributed systems: Cost functions and resolution methods; [estudio del problema de asignación de tareas en los sistemas distribuidos: Funciones de costo y métodos de resolución] [Cited by: 2]. *Revista Tecnica de la Facultad de Ingenieria Universidad del Zulia*, *20*(3), 203–213. https://www.scopus.com/inward/record.uri?eid=2-s2.0-0031327066&partnerID=40&md5=a7347c4dff554d47a1521eb82165f44f

Azketa, E., Mendialdua, X., Ibarguren, I., & Solís, A. (2021). Synchronization method for distributed systems with functional safety; [método de sincronización para sistemas distribuidos con seguridad funcional] [Cited by: 0; All Open Access, Gold Open Access, Green Open Access]. *RIAI - Revista Iberoamericana de Automatica e Informatica Industrial*, *18*(2), 113–118. https://doi.org/10.4995/RIAI.2020.14022

Caram, F., Proto, A., Merlino, H., & García-Martínez, R. Study of the processing capacity of distributed systems using the ising model; [estudio de capacidad de procesamiento de sistemas distribuidos utilizando el modelo de ising] [Cited by: 0]. In: Cited by: 0. 2012, 119–126. https://www.scopus.com/inward/record.uri?eid=2-s2.0-84883635608&partnerID=40&md5=6324677363d8770558a1e730211a200a

Castro, J. L. A., & Matos, R. L. S. (2007). Cache memory coherence protocol for distributed systems; [protocolo de coherencia de memoria cache para sistemas distribuidos] [Cited by: 0]. *Revista Tecnica de la Facultad de Ingenieria Universidad del Zulia*, *30*(2), 170–178. https://www.scopus.com/inward/record.uri?eid=2-s2.0-47749154791&partnerID=40&md5=dfb61f3e417ee16c5a8338206b92f35a

Dzulkifly, S., Seng, W. Y., Yeh, L. H., Ibrahim, A. B., & Zhiqiang, S. (2025). Pantas.io: Game-based learning to cultivate programming skills for primary school students [Cited by: 0; All Open Access, Hybrid Gold Open Access]. *Journal of Advanced Research in Applied Sciences and Engineering Technology*, *50*(1), 178–190. https://doi.org/10.37934/araset.50.1.178190

Gómez-Baryolo, O., Machado-Sosa, A., & Cabrera-Mondeja, A. (2017). Integration architecture for distributed systems based on socket; [arquitectura de integración basada en socket para sistemas distribuidos] [Cited by: 0]. *Espacios*, *38*(59). https://www.scopus.com/inward/record.uri?eid=2-s2.0-85038101436&partnerID=40&md5=6fe5130530f523a510a367d77c590164

Huang, T., Xie, R., Ren, Y., Yu, F. R., Zou, Z., Han, L., Liu, Y., Cheng, D., Li, Y., & Liu, T. (2024). Dtais: Distributed trusted active identity resolution systems for the industrial internet [Cited by: 1; All Open Access, Gold Open Access]. *Digital Communications and Networks*, *10*(4), 853–862. https://doi.org/10.1016/j.dcan.2023.06.006

Li, J., Moeini, B., Nejati, S., Sabetzadeh, M., & McCallen, M. (2024). A lean simulation framework for stress testing iot cloud systems [Cited by: 0; All Open Access, Green Open Access]. *IEEE Transactions on Software Engineering*, *50*(7), 1827–1851. https://doi.org/10.1109/TSE.2024.3402157

Martínez, J. T. F., Agostini, F., & La Red Martínez, D. L. (2021). Decision model for process and resource management in distributed systems with workload balancing;

[modelo de decisión para gestión de procesos y recursos en sistemas distribuidos con balanceo de carga de trabajo] [Cited by: 0]. *RISTI - Revista Iberica de Sistemas e Tecnologias de Informacao*, *2021*(E42), 343–356. https://www.scopus.com/inward/record.uri?eid=2-s2.0-85106993759&partnerID=40&md5=66d427982b1767dccb8462a50075a5c5

Rao, J., & Han, X. (2024). Analysis on the application of ecological environment protection system under blockchain technology [Cited by: 0; All Open Access, Hybrid Gold Open Access]. *Frontiers in Artificial Intelligence and Applications*, *384*, 1026–1035. https://doi.org/10.3233/FAIA240105

Saini, S. S., & Sharma, L. S. (2024). Investigation of the http live streaming media protocol's (hls) adaptability and performance [Cited by: 0]. *Journal of The Institution of Engineers (India): Series B*. https://doi.org/10.1007/s40031-024-01132-w

Terencio, N., Wella, & Sony, A. (2023). Enhancing mysql database security with mysql enterprise transparent data encryption [Cited by: 0; All Open Access, Bronze Open Access]. *Journal of Logistics, Informatics and Service Science*, *10*(4), 154–173. https://doi.org/10.33168/JLISS.2023.0411

Xavier, K. F., Kabirdoss, D., Seetharaman, C., Mani, K., Veeramani, G., & Thiyagarajan, V. K. M. (2025). Monitoring the manufacturing operation process data through the cloud database [Cited by: 0; All Open Access, Hybrid Gold Open Access]. *Journal of Advanced Research in Applied Sciences and Engineering Technology*, *50*(1), 276–285. https://doi.org/10.37934/araset.50.1.276285

Zou, Y., Yang, L., Jing, G., Zhang, R., Xie, Z., Li, H., & Yu, D. (2024). A survey of fault tolerant consensus in wireless networks [Cited by: 2; All Open Access, Gold Open Access]. *High-Confidence Computing*, *4*(2). https://doi.org/10.1016/j.hcc.2024.100202

Aggarwal and Singh, 2024 Aguilar Castro and Kloul, 1997 Azketa et al., 2021 Caram et al., 2012 Castro and Matos, 2007 Dzulkifly et al., 2025 Gómez-Baryolo et al., 2017 Huang et al., 2024 Martínez et al., 2021 Rao and Han, 2024 Saini and Sharma, 2024 Xavier et al., 2025 Zou et al., 2024 Li et al., 2024 Terencio et al., 2023