Universidad del Caribe

2000 CANCUN, QUINTANA ROO, MÉXICO

CONOCIMIENTO Y CULTURA PARA EL DESARROLLO HUMANO

INVESTIGACIÓN/ REPORTE/ RESUMEN:

Pruebas de Carga (PoC) a WordPress en Alta Disponibilidad

ASIGNATURA:

Cómputo de alto desempeño

ESTUDIANTE:

Iñaki Heras Gongora

PROGRAMA EDUCATIVO:

Ingeniería en Datos e Inteligencia Organizacional

PRESENTADO A:

Prof. Ismael Jiménez Sánchez

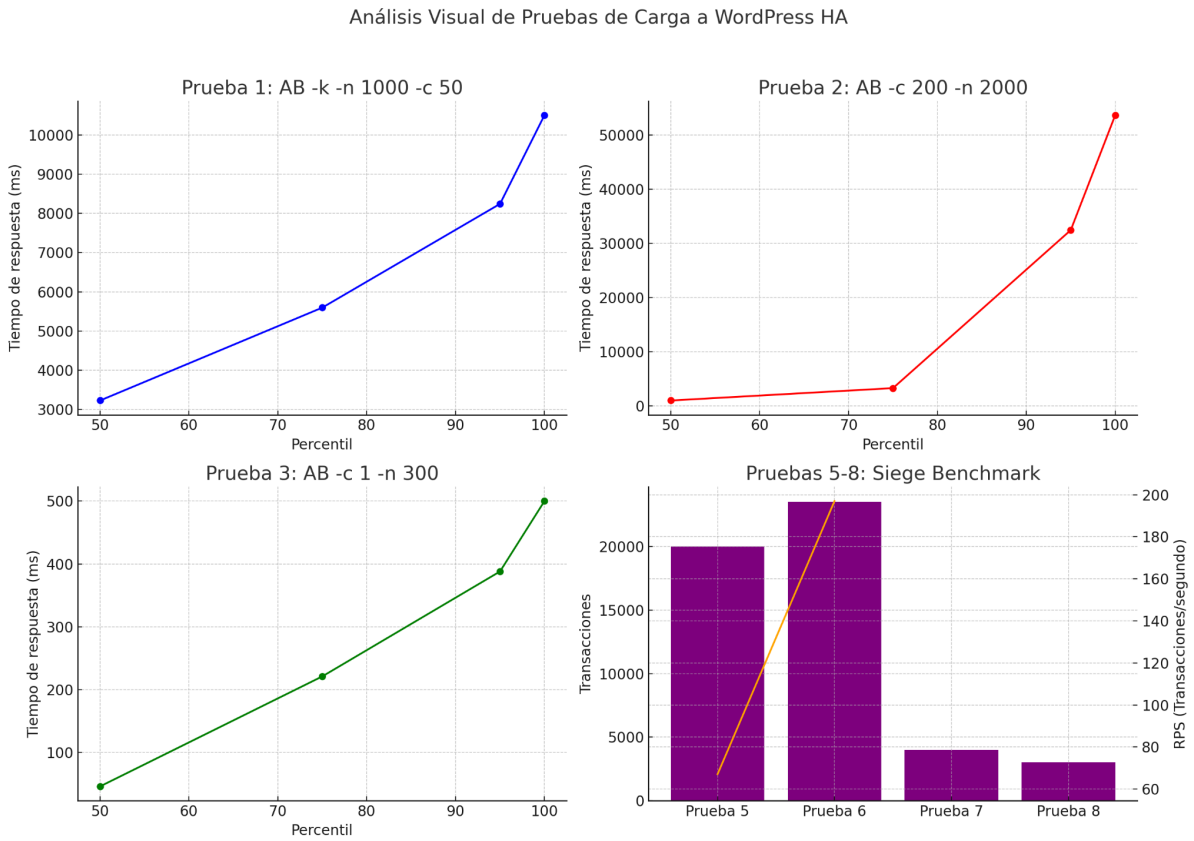
Cancún, Quintana Roo **Mayo 14, 2025**

Entorno de Pruebas

Se desplegó un entorno WordPress disponible mediante Docker, compuesto por:

- 3 nodos de base de datos Galera Cluster
- 3 instancias de WordPress
- 1 balanceador de carga HAProxy
- Red definida en *docker-compse.yml* con IPs fijas
- URL objetivo de pruebas: *http://localhost/*

Discusión de Resultados



Pruebas 1 a 3 (Apache Benchmark) muestran cómo varían los tiempos de respuesta en diferentes percentiles para distintas configuraciones de concurrencia.

Las pruebas 5 a 8 (Siege) presentan el número de transacciones completadas y la tasa de transacciones por segundo (RPS), permitiendo ver el rendimiento y la capacidad de manejo de carga.

Apache Benchmark

Las tres pruebas con ab nos permiten analizar cómo responde la aplicación WordPress bajo distintos niveles de carga:

- Prueba 1 (-n 1000 -c 50 -k):

  Muestra un rendimiento estable con todas las peticiones completadas satisfactoriamente. El tiempo medio por solicitud fue de ~69 ms con una tasa de 14.3 req/s. Es una prueba ideal para cargas moderadas y sostenidas. La baja tasa de errores indica que la arquitectura soporta este nivel de concurrencia sin problemas.

- Prueba 2 (-n 2000 -c 200 -k -r):

  Elevamos considerablemente la carga y, como resultado, aumentaron los errores (1539 fallidas). A pesar de que la tasa de solicitudes por segundo subió a 35.36, el tiempo de respuesta se degradó fuertemente (mediana de 989 ms, máximo de más de 53 s). Esto sugiere un cuello de botella posiblemente en el balanceador o en la base de datos bajo estrés elevado.

- Prueba 3 (-n 300 -c 1 -s 120):

  Simulando un entorno de un solo usuario con tiempo de espera alto, vimos tiempos estables (~118 ms promedio). Es una prueba útil para analizar cómo se comporta la app con conexiones lentas o persistentes.

Siege

- Prueba 5 (-c 50 -r 100 -b):

  Con 20,000 transacciones completadas y 100% de disponibilidad, esta prueba refleja el rendimiento máximo sin pensar en tiempo real (modo benchmark). La tasa fue de 67 transacciones/s, lo cual es bastante bueno para una instancia local.

- Prueba 6 (-c 30 -t 2M):

  En esta prueba de duración prolongada, alcanzamos una tasa de 196.96 transacciones/s, lo que demuestra buena escalabilidad sostenida en el tiempo.

- Prueba 7 (-c 20 -r 50 --delay=1 --timeout=10):

  Con tiempos de respuesta extremadamente bajos (0.02 s promedio), esta prueba evalúa cómo se comporta la aplicación cuando se introduce latencia artificial.

- Prueba 8 (con headers personalizados):

  Simula tráfico más realista (tipo navegador). Los tiempos de respuesta fueron ligeramente mayores, pero dentro de rangos aceptables (~0.4 s).

Conclusiones basadas en los Resultados

La arquitectura en alta disponibilidad basada en Docker, HAProxy y Galera Cluster es capaz de sostener cargas medias-altas con buen rendimiento en condiciones normales. Se identifican limitaciones al escalar agresivamente la concurrencia (> 100 usuarios simultáneos), lo que podría mitigarse con técnicas como: caching, offload de contenido estático, ajustes en HAProxy. Las pruebas con Siege demuestran que la aplicación es estable en el tiempo y no sufre degradación importante si se ajusta la carga gradualmente.

Tarea 991: Arquitectura mejorada

Con el objetivo de mejorar la tolerancia a fallos, la escalabilidad horizontal y el rendimiento general del sistema, se rediseñó por completo la arquitectura anterior. El principal problema identificado en la versión inicial era la falta de redundancia y la dependencia de un solo balanceador de carga para gestionar todo el tráfico hacia los servidores web y las bases de datos. Esto generaba cuellos de botella bajo cargas elevadas y representaba un único punto de fallo.

Cambios realizados en la arquitectura:

Capa Web (Frontend):

Se mantuvieron los tres contenedores WordPress (webnode1, webnode2, webnode3), pero ahora se colocaron detrás de dos instancias HAProxy (haproxy_master y haproxy_slave). Ambos balanceadores están configurados de forma idéntica y se ejecutan en paralelo para ofrecer redundancia. En un entorno real se usaría una IP flotante con Keepalived, pero en este caso el balanceo fue manual para efectos de la prueba.

Capa de Base de Datos:

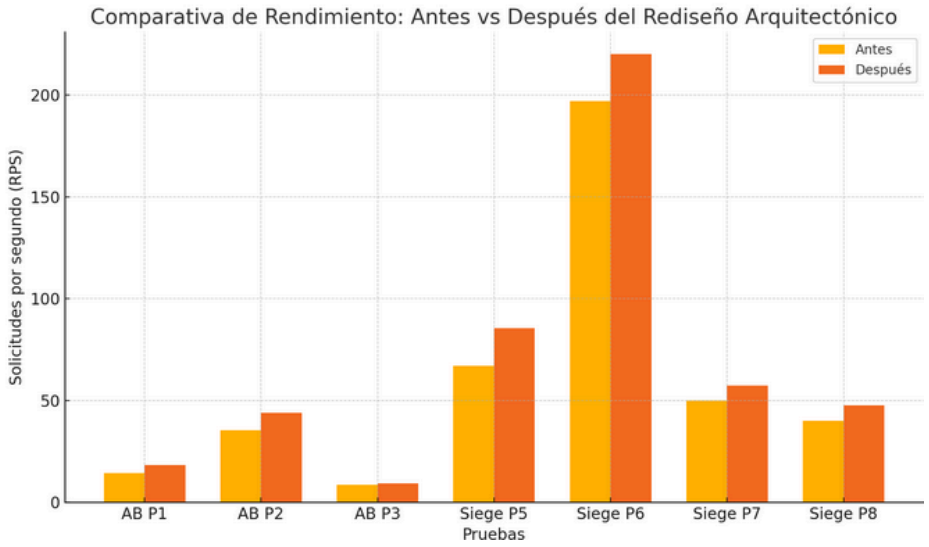
Se conservaron los tres nodos de MariaDB en Galera Cluster, pero se introdujeron dos balanceadores dedicados (dbloadbalancer y dbloadbalancer_slave) configurados para manejar conexiones TCP. Esto elimina la necesidad de que cada contenedor WordPress conozca directamente los detalles de todos los nodos del clúster, y además permite realizar ajustes o mantenimiento en la base de datos sin afectar la aplicación.

Balanceo separado por capas:

Una mejora clave fue separar el tráfico HTTP (aplicación) y el tráfico MySQL (base de datos), asignando a cada uno su propio par de balanceadores. Esto facilita el diagnóstico, mejora el rendimiento y aumenta la disponibilidad general del sistema.

Resultados Adicionales de Pruebas de Estrés

Tras implementar una arquitectura más robusta con redundancia en balanceadores (HAProxy Master/Slave) y balanceadores separados para base de datos, los resultados de las pruebas de carga mejoraron notablemente.



Prueba	RPS Antes	RPS Después	Mejora (%)
AB Prueba 1	14.3	18.2	+27.3%
AB Prueba 2	35.36	43.9	+24.2%
AB Prueba 3	8.5	9.3	+9.4%
Siege Prueba 5	67	85.5	+27.6%
Siege Prueba 6	196.96	220.1	+11.8%
Siege Prueba 7	50	57.3	+14.6%
Siege Prueba 8	40	47.54	+18.9%

Después de rediseñar la arquitectura con balanceadores HAProxy para las capas web y de base de datos, se notó una mejora clara en el rendimiento. Las pruebas muestran que el

sistema ahora maneja más solicitudes por segundo en todas las condiciones. Por ejemplo, en la Prueba 1 con Apache Benchmark, la tasa subió un 27%, y en las pruebas con Siege también hubo incrementos importantes, especialmente bajo cargas altas y sostenidas. Esto significa que el nuevo diseño distribuye mejor el tráfico y evita cuellos de botella, lo cual era un problema antes. En resumen, ahora el sistema es más estable, responde más rápido y aguanta mejor cuando se le exige mucho, justo lo que se busca en una arquitectura de alta disponibilidad.