Determinación de instancias requeridas para un servidor de aplicación web.

Matias Walter Orieta, Juan Pablo Castiglione

Universidad Tecnológica Nacional, Facultad Regional Buenos Aires

Abstract

El presente trabajo tiene como objetivo la optimización de los recursos necesarios para el correcto funcionamiento de un servidor de aplicaciones web, en términos de cantidad de instancias e hilos de procesamiento. El problema principal radica en determinar la cantidad óptima de instancias mínimas y máximas, así como la cantidad de hilos por instancia, con el propósito de optimizar el tiempo de respuesta promedio de las peticiones, manteniendo una holgura adecuada para evitar sobrecargas o infrautilización de recursos. Para ello, se realizó una simulación de la aplicación web bajo condiciones de máxima demanda, utilizando funciones de densidad de probabilidad para modelar los intervalos entre llegadas y los tiempos de atención de las peticiones. Durante el desarrollo de la simulación y el análisis de los resultados, se emplearon diversas herramientas de software y técnicas de programación, con el objetivo de evaluar métricas como el tiempo de respuesta promedio, el porcentaje de tiempo ocioso, y el uso de instancias. Como conclusión, se determinaron los valores óptimos para los parámetros mencionados, logrando mejorar tanto la eficiencia del servidor como la experiencia del usuario final.

Introducción

La presente investigación se enfoca en el análisis de la gestión de recursos en servidores de aplicaciones web, tomando como caso de estudio un sistema de balanceo de carga basado en instancias virtualizadas. Este trabajo se fundamenta en la creciente demanda de aplicaciones web con altos requerimientos de disponibilidad y rendimiento, así como en la necesidad de administrar eficientemente los recursos disponibles para responder adecuadamente a las solicitudes de los usuarios.

La aplicación web estudiada enfrenta diversos desafíos operativos, tales como la correcta asignación de instancias y hilos para atender el flujo de peticiones recibidas. Entre las estrategias utilizadas para abordar estas necesidades se encuentran la creación dinámica de instancias según la demanda, la distribución equitativa de las solicitudes y el apagado de instancias cuando no son necesarias.

El objetivo principal de esta investigación es evaluar y determinar las cantidades óptimas de instancias mínimas y máximas, así como la cantidad de hilos por instancia, con el propósito de lograr un mejor tiempo de respuesta promedio y mantener una holgura adecuada en la gestión de recursos. A través de la simulación y el análisis de este sistema, se busca contribuir al entendimiento teórico y práctico de la gestión eficiente de servidores de aplicaciones web bajo escenarios de alta demanda.

Elementos del trabajo y metodología

Para llevar a cabo este trabajo de simulación, cuyo objetivo es optimizar la cantidad de instancias e hilos de un servidor de aplicaciones web, se ha utilizado Python junto con sus bibliotecas especializadas para simulación y análisis de datos, como numpy, pandas y scipy.

Se empleó la metodología de simulación Evento a Evento, conocida por su capacidad de modelar sistemas dinámicos en los que los eventos ocurren en momentos discretos. En este caso, se simuló el comportamiento de las instancias y la ocupación de hilos a medida que las peticiones llegaban y eran atendidas, registrando el tiempo comprometido de cada hilo y la respuesta total al usuario. La metodología Evento a Evento es ampliamente utilizada en disciplinas como la ingeniería y la ciencia de la computación para analizar sistemas complejos y dinámicos, especialmente cuando el uso de métodos analíticos tradicionales resulta difícil. Esta técnica ha permitido una comprensión detallada del comportamiento del servidor en diferentes condiciones de carga, ayudando a

determinar configuraciones óptimas para mejorar el rendimiento y la experiencia del usuario.

Desarrollo

Para proceder con la simulación utilizando la metodología de simulación Evento a Evento, se comenzó por definir y registrar las variables específicas necesarias para modelar el comportamiento del servidor de aplicaciones web. Estas variables se clasificaron en dos categorías principales: exógenas, que comprenden los datos y las variables de control, y endógenas, que incluyen las variables de estado y de resultado.

Las variables exógenas se desglosan en:

Datos:

Intervalo entre arribos (IA): Tiempo entre la llegada de dos peticiones consecutivas, registrado durante un momento pico del día.

Tiempo de atención por petición (TA): Tiempo necesario para procesar y responder cada petición recibida por el servidor.

Variables de control:

Cantidad máxima de instancias (CImax): Límite superior de instancias que pueden ejecutarse simultáneamente.

Cantidad mínima de instancias (CImin): Límite inferior de instancias que deben estar activas en todo momento.

Cantidad de hilos por instancia (CH): Número de hilos que cada instancia tiene disponibles para atender las peticiones.

Las variables endógenas incluyen:

Variables de resultado:

Tiempo de respuesta promedio (TRP): Tiempo total que toma responder una petición, incluyendo el tiempo de espera y el tiempo efectivo de atención.

Cantidad máxima de instancias levantadas (CILmáx): Máximo número de instancias activas durante la simulación.

Porcentaje de tiempo ocioso (PTO): Proporción del tiempo en que cada hilo o instancia se encuentra sin atender peticiones.

Variable de estado:

Tiempo comprometido por hilo (TC(ij)): Tiempo durante el cual cada hilo de una instancia está ocupado atendiendo una petición. Por ejemplo, TC(2,3) representa el tiempo comprometido del hilo 3 en la instancia 2.

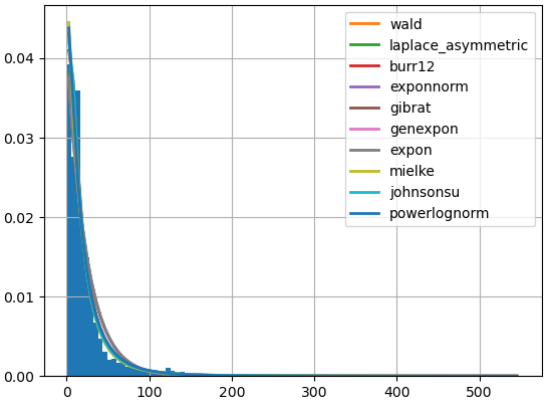
Es fundamental resaltar que las funciones de densidad de probabilidad desempeñan un papel central en el trabajo desarrollado. En el ámbito de la simulación, estas funciones matemáticas describen la probabilidad de que una variable aleatoria adquiera un valor específico dentro de un rango definido. Constituyen una herramienta esencial en la modelización y análisis de sistemas estocásticos durante el proceso de simulación.

Las variables aleatorias se emplean para capturar la incertidumbre o variabilidad que existe naturalmente en un sistema o proceso. Estas variables pueden abarcar aspectos como los tiempos de llegada o los tiempos de atención, entre otros. La función de densidad de probabilidad (FDP) se utiliza para describir la distribución de probabilidad asociada a estas variables aleatorias.

En el contexto de nuestra investigación sobre la gestión de recursos de servidores de aplicaciones web, hemos analizado el intervalo entre arribos (IA) y el tiempo de atención (TA) de las peticiones. Durante la recolección de datos, descubrimos que no existe un registro sistemático del tiempo preciso en que se completa cada solicitud. No obstante, basándonos en el comportamiento observado en sistemas

similares y en la experiencia del sector, se ha estimado que el tiempo de atención suele oscilar entre ciertos valores dependiendo de la complejidad de la petición.

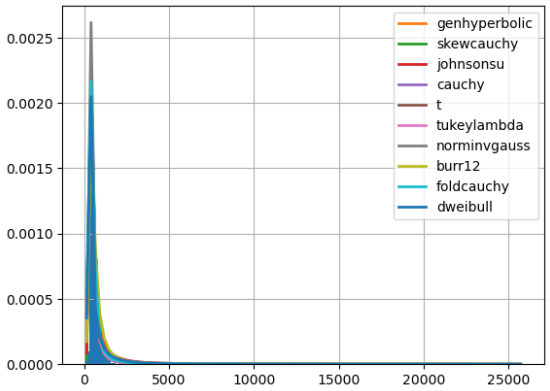
Para obtener una representación más precisa del intervalo entre arribos durante el momento de mayor carga del día, alrededor de las 14:30 horas, se utilizó un dataset que registraba las marcas de tiempo (timestamp) de las llegadas de las peticiones, expresadas en milisegundos. A partir de estos datos, se determinó la función de densidad de probabilidad (fdp) correspondiente mediante el uso de las bibliotecas fitter y scipy de Python, que permitieron ajustar diferentes distribuciones y seleccionar la más adecuada. En este caso, fitter indicó que la distribución de Wald ofrecía el mejor ajuste para modelar los intervalos entre las llegadas de las peticiones.



[Figura 1]

Este hallazgo es significativo, ya que la distribución de Wald es conocida por su capacidad para modelar tiempos de espera y de entrega en sistemas de colas, donde la variabilidad y la asimetría son comunes. Al capturar la naturaleza estocástica de los procesos de llegada, esta distribución resulta crucial para simular con precisión el comportamiento del servidor bajo condiciones de alta demanda. Esta estimación nos permitió modelar la situación en la que el servidor enfrenta su mayor carga y evaluar su comportamiento bajo dichas condiciones, buscando optimizar la gestión de recursos para reducir los tiempos de espera y maximizar la eficiencia en el uso de las instancias y los hilos.

Además, se utilizó otro conjunto de datos que contenía los tiempos de atención de las peticiones. Al aplicar el mismo proceso de ajuste de distribuciones, se determinó que la distribución genhyperbolic proporcionaba el mejor ajuste para modelar estos tiempos.



[Figura 2]

Esta distribución es particularmente adecuada para capturar la variabilidad observada en los tiempos de atención, lo que permite una representación más precisa del rendimiento del servidor.

Posteriormente, se utilizó la función rvs de la biblioteca scipy.stats para generar muestras aleatorias a partir de las distribuciones ajustadas. Esta función se destaca por su capacidad para generar números aleatorios de manera eficiente, permitiendo simular datos sin la necesidad de cálculos complejos por parte del usuario.

Una de las principales ventajas de rvs es su flexibilidad, ya que admite diferentes tipos de distribuciones, lo que facilita la adaptación a diversas situaciones y contextos. Además, al manejar internamente el proceso de muestreo, se simplifica el flujo de trabajo y se reduce el riesgo de errores en la implementación. Esto permite a los investigadores centrarse en el análisis y la interpretación de los resultados sin

preocuparse por la complejidad matemática detrás del muestreo.

Tabla de Eventos Independientes

Eventos	EFNC	EFC	Condiciones
Llegada de petición	Llegada de petición	-	-

[Tabla 1]

La Tabla de Eventos Independientes (TEI) en la simulación del servidor web es una estructura clave para rastrear y gestionar los eventos que tendrán lugar durante la ejecución del sistema. Es esencial en simulaciones de eventos discretos, donde eventos específicos impulsan la evolución del sistema en momentos determinados, a diferencia de cambios continuos. Esto permite modelar y analizar el comportamiento del servidor en función de eventos críticos, facilitando la comprensión y la toma de decisiones.

En el contexto de nuestro servidor web, el evento principal en la TEI es la llegada de una nueva petición. Cada vez que se recibe una solicitud, se desencadena el procesamiento de esa petición, lo que afecta directamente a la variable de estado, como el tiempo comprometido de los hilos disponibles en las instancias.

Dado que disponemos de datos sobre el intervalo entre arribos de pedidos, podemos anticipar que, tras una llegada, se producirá una nueva solicitud en un tiempo predecible.

Además, implementamos un modelo de tiempo comprometido, ya que, en el contexto del servidor, la planificación de la atención a los pedidos se realiza de manera estructurada para optimizar recursos y tiempos. Por lo tanto, no contamos con eventos condicionados en nuestro modelo, lo que simplifica el flujo de la simulación y permite una evaluación más directa del rendimiento del servidor bajo distintas cargas de trabajo.

Tabla de Eventos Futuros

La Tabla de Eventos Futuros (TEF) es una estructura utilizada en simulaciones para rastrear y gestionar eventos que no dependen del estado o resultados de otros eventos en el sistema. En el contexto de nuestro servidor web, la TEF se centra en anticipar los momentos en los que se producirán nuevas llegadas de pedidos, lo que es fundamental para la planificación y gestión de recursos.

La TEF, en este caso, está formada por:

- Tiempo de la próxima llegada de petición (TPLL)

Resultados

Para poder realizar la simulación y obtener conclusiones que mejoren la eficiencia del servidor web, es necesario plantear varios escenarios factibles y hacer las comparaciones pertinentes.

En este caso, exploraremos tres escenarios diferentes para determinar la cantidad óptima de instancias y hilos, así como el impacto en el tiempo de respuesta promedio.

Escenario 1

Se plantea un primer escenario de control para realizar la simulación teniendo en cuenta el escenario actual del servidor con 2 instancias virtualizadas.

- Cantidad de instancias (CImin): 2
- Cantidad máxima de instancias (CImax): 5
- Cantidad de hilos por instancia (CH): 4

Se simularon aproximadamente 100.000 milisegundos. Para este escenario los resultados fueron los siguientes:

- Tiempo de respuesta promedio: 348604.06 milisegundos

- Porcentaje de tiempo ocioso (PTO): 13.82%
- Máxima cantidad de instancias levantadas en simultáneo: 3

Escenario 2

En este segundo escenario, se reduce la cantidad mínima de instancias a 1 para evaluar la capacidad del servidor en condiciones de baja disponibilidad.

- Cantidad de instancias (CImin): 1
- Cantidad máxima de instancias (CImax): 5
- Cantidad de hilos por instancia (CH): 4

Se obtuvieron los siguientes resultados:

- Tiempo de respuesta promedio: 478693.10 milisegundos
- Porcentaje de tiempo ocioso (PTO): 0.26%
- Máxima cantidad de instancias levantadas en simultáneo: 2

Escenario 3

En este tercer escenario, se incrementa la cantidad mínima de instancias a 3 para evaluar el rendimiento del servidor en condiciones de alta capacidad.

- Cantidad de instancias (CImin): 3
- Cantidad máxima de instancias (CImax): 6
- Cantidad de hilos por instancia (CH): 4
-

Una vez realizada la corrida obtenemos los siguientes resultados:

- Tiempo de respuesta promedio: 244976.80 milisegundos
- Porcentaje de tiempo ocioso (PTO): 26.13%
- Máxima cantidad de instancias levantadas en simultáneo: 4

Conclusión

En este estudio, se buscó optimizar la cantidad de instancias de un servidor web para mejorar tanto los costos operativos como los tiempos de respuesta de los usuarios. Mediante una simulación exhaustiva, se evaluaron diferentes escenarios y se analizaron los resultados obtenidos.

Los hallazgos indican que, para mantener tiempos de respuesta eficientes, una configuración de dos instancias mínimas logra un buen balance entre la capacidad de respuesta y el uso eficiente de recursos. Este equilibrio es esencial para evitar tanto la saturación de recursos, que incrementa el tiempo de espera para los usuarios, como el subuso de las instancias, que aumenta los costos sin generar un beneficio adicional.

El análisis muestra que un menor número de instancias mínimas (Escenario 2) resulta en tiempos de respuesta significativamente mayores, evidenciando insuficiencia para cubrir los picos de demanda, mientras que un mayor número de instancias (Escenario 3) resulta en tiempos de respuesta mejores, aunque con un aumento considerable en el porcentaje de tiempo ocioso. Esto resalta la necesidad de evitar sobreprovisionamiento para no desperdiciar recursos.

Por tanto, se concluye que, para alcanzar una mayor eficiencia en el servicio, se debe encontrar un equilibrio adecuado en la cantidad de instancias disponibles, ajustando de manera dinámica según la carga esperada. Asimismo, el uso de simulaciones de eventos discretos, apoyadas por herramientas de programación y análisis estadístico, permite obtener insights cruciales para mejorar la gestión operativa y optimizar el rendimiento del servidor, contribuyendo a una mejor calidad de servicio y satisfacción del usuario.