Determinación de recursos requeridos 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 solicitudes que debieron esperar, y el uso máximo 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.

Palabras Clave

Servidor de aplicaciones web, Cargas de trabajo, Balanceo de carga, Rendimiento de aplicaciones web, Parámetros de servidor, Gestión de servidores, Virtualización de instancias.

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 e 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, y la distribución equitativa de las solicitudes.

La gestión de peticiones funciona de la siguiente manera: la instancia con menos hilos ocupados será la encargada de procesar y responder la petición entrante. Si una de las instancias llega a una ocupación del 50% de sus hilos disponibles, una nueva instancia se inicializa. Si en el momento de llegar una petición, todos los hilos de todas las instancias se encuentran ocupados, esperará a que se libere alguno. Para el usuario este tiempo de espera forma parte del tiempo de respuesta total. Los administradores pueden elegir cuál será el número de hilos por instancia, y las cantidades mínimas y máximas de instancias a ejecutar por vez.

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 el tiempo de respuesta total al usuario en milisegundos.

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. Expresado en milisegundos.

Tiempo de atención por petición (TA):
Tiempo necesario para procesar cada petición recibida por el servidor. Expresado en milisegundos.

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 Solicitudes que Esperaron (PTE): la proporción de solicitudes que al llegar no encontrar un hilo libre para ser atendidas.

Variable de estado:

Tiempo comprometido por hilo (TC(i,j)): 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. Siendo $0 <= i <= CImax - 2$, y $0 <= j <= CH - 1$.

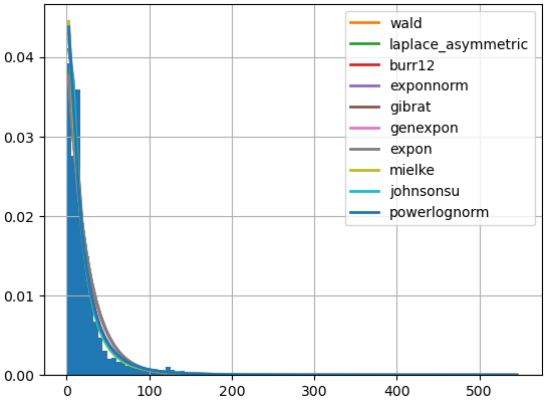
Es fundamental resaltar que las funciones de densidad de probabilidad desempeñan un papel central en el trabajo desarrollado. En el ámbito de la simulación, estas funciones matemáticas describen la probabilidad de que una variable aleatoria adquiera un valor específico dentro de un rango definido. Constituyen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

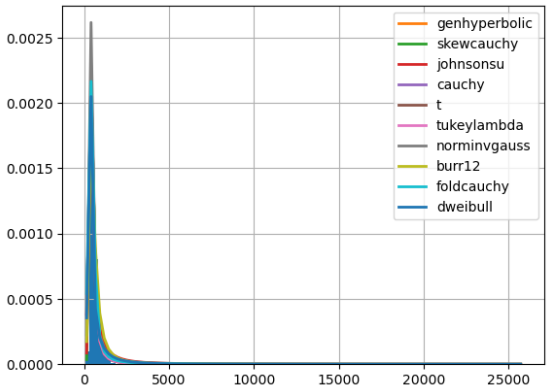
En el contexto de nuestra investigación sobre la gestión de recursos de servidores de aplicaciones web, hemos analizado el intervalo entre arribos (IA) y el tiempo de atención (TA) de las peticiones.

Para obtener una representación más precisa del intervalo entre arribos durante el momento de mayor carga del día, alrededor de las 14:30 horas para esta aplicación, se utilizó un dataset de 3000 registros que presenta 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]

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 nos permitió centrarnos en el análisis y la interpretación de los resultados sin preocuparse por la complejidad matemática detrás del muestreo.

Tabla de Eventos Independientes

[Tabla 1]

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

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 la carga de trabajo del horario estudiado.

Tabla de Eventos Futuros.

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 e 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): 1
- Cantidad máxima de instancias (CImax): 5
- Cantidad de hilos por instancia (CH): 10

Se simularon aproximadamente 600.000 milisegundos. Para este escenario los resultados fueron los siguientes:

- Tiempo de respuesta promedio: 1220181.41 milisegundos
- Porcentaje de solicitudes que tuvieron que esperar (PTE): 62.39%
- Máxima cantidad de instancias levantadas en simultáneo: 5

Escenario 2

En este segundo escenario, decidimos aumentar la cantidad de hilos disponibles por instancia a 50, ya que nuestra infraestructura nos lo permitiría.

- Cantidad de instancias (CImin): 1
- Cantidad máxima de instancias (CImax): 5
- Cantidad de hilos por instancia (CH): 50

Se obtuvieron los siguientes resultados:

- Tiempo de respuesta promedio: 914.61 milisegundos
- Porcentaje de solicitudes que tuvieron que esperar (PTE): 0,00%
- Máxima cantidad de instancias levantadas en simultáneo: 3

Escenario 3

En este tercer escenario, se incrementa la cantidad mínima de instancias a 5, la máxima a 10 y la cantidad de hilos por instancia a 100 para evaluar el rendimiento del servidor en condiciones de alta capacidad.

- Cantidad mínima de instancias (CImin): 5
- Cantidad máxima de instancias (CImax): 10
- Cantidad de hilos por instancia (CH): 100
-

Una vez realizada la corrida obtenemos los siguientes resultados:

- Tiempo de respuesta promedio: 1058.96 milisegundos
- Porcentaje de solicitudes que tuvieron que esperar (PTE): 0,00%
- Máxima cantidad de instancias levantadas en simultáneo: 5

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.

El análisis muestra que un menor número de instancias mínimas (escenario 1) resulta en tiempos de respuesta significativamente mayores, evidenciando insuficiencia para cubrir los picos de demanda, esta primera simulación arrojó resultados absurdos e inaceptables. Por otro lado, el escenario 3 resulta en tiempos de respuesta mejores, pero observando la cantidad máxima de instancias levantadas, y el porcentaje nulo de solicitudes que tuvieron que esperar, determinamos que esta configuración sería un desperdicio de recursos que se podrían asignar a otros servicios.

Finalmente, el escenario 2 presentó un excelente tiempo de respuesta promedio, contando con cierta holgura respecto al tope de instancias, y mostrando también que el piso de instancias y la cantidad de hilos por instancia no eran demasiado altos de modo que representaran un desperdicio de recursos.

Por tanto, se concluye que una configuración de 1 instancia mínima, 5 máximas y 50 hilos por instancia hará un uso eficiente de los recursos disponibles manteniendo una calidad de servicio ideal.