UNIVERSIDAD DE LAS FUERZAS ARMADAS

Computación Paralela

Departamento de Ciencias de la Computación

Load Balancing

Estudiantes:
Josué Merino, Angelo Sánchez, Justin Villarroel

Docente:
Ing. Carlos Andrés Pillajo Bolagay

Resultados de Aprendizaje

1. **Análisis de ingeniería.** La capacidad de identificar, formular y resolver problemas de ingeniería en su especialidad; elegir y aplicar de forma adecuada métodos analíticos, de cálculo y experimentales ya establecidos; reconocer la importancia de las restricciones sociales, de salud y de seguridad, ambientales, económicas e industriales.
2. **Proyectos de ingeniería.** Capacidad para proyectar, diseñar y desarrollar productos complejos (piezas, componentes, productos acabados, etc.) procesos y sistemas de su especialidad, que cumplen con los requisitos establecidos, incluyendo tener conciencia de los aspectos sociales, de salud y seguridad, ambientales, económicos e industriales; así como seleccionar y aplicar métodos de proyectos apropiados.
3. **Aplicación práctica de la ingeniería.** Comprensión de las técnicas aplicables y métodos de análisis, proyecto e investigación y sus lineamientos en el ámbito de su especialidad.
4. **Comunicación y trabajo en equipo.** Capacidad para comunicar eficazmente información, ideas, problemas y soluciones en el ámbito de ingeniería y con la sociedad en general.

Desarrollo

Round Robin

En Round Robin se observa una distribución igual a todos los servidores. Inicialmente, no se evidencia una pérdida de paquetes de manera abrupta; sin embargo, con el paso del tiempo se observan paquetes encolados. Por lo tanto, se puede decir que este algoritmo funciona de manera óptima al inicio, pero presenta encolamiento en los servidores con el transcurso del tiempo.

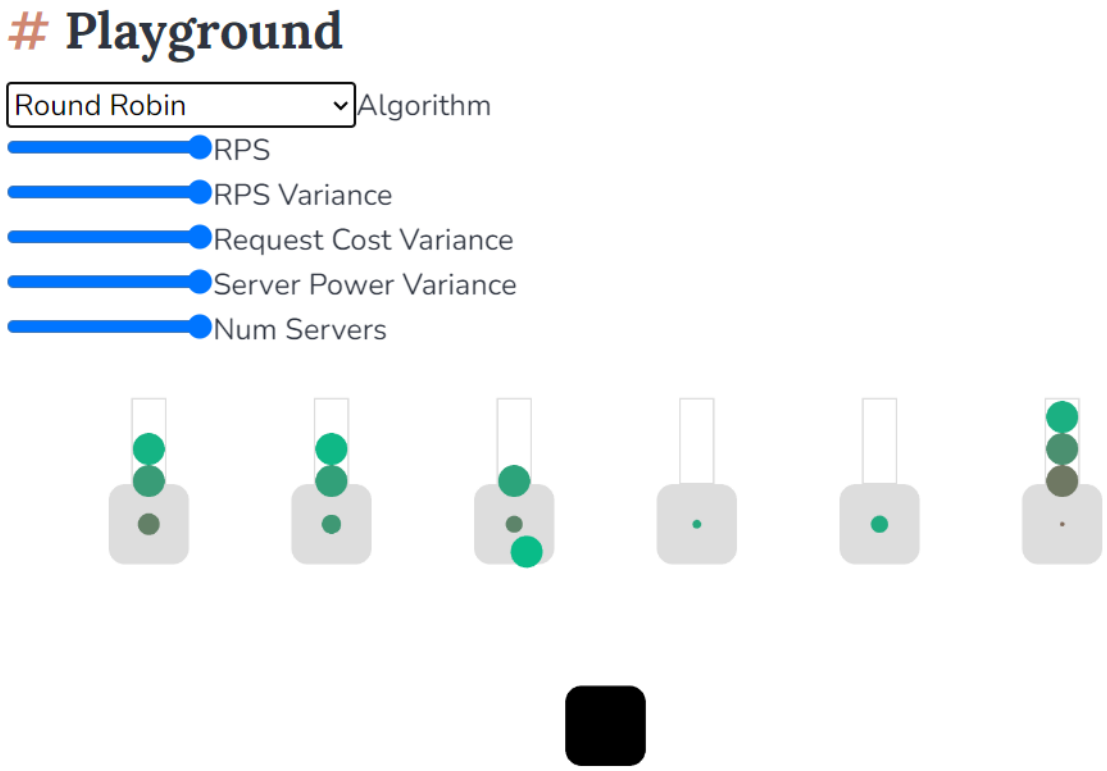

Figura 1: Round Robin al inicio

Figura 2: Round Robin con el correr del tiempo

Weighted Random

El algoritmo Weighted Random es un método para seleccionar servidores de esta colección de acuerdo con sus pesos, esencialmente dando a los servidores con pesos más altos una mayor probabilidad de ser elegidos.(5)

Funcionamiento

- Primero, calculamos el peso total sumando los pesos de todos los elementos en la colección.
- Luego, generamos un número aleatorio dentro del rango de 1 al peso total.
- Buscamos en qué área (elemento) cae el número aleatorio según los pesos.

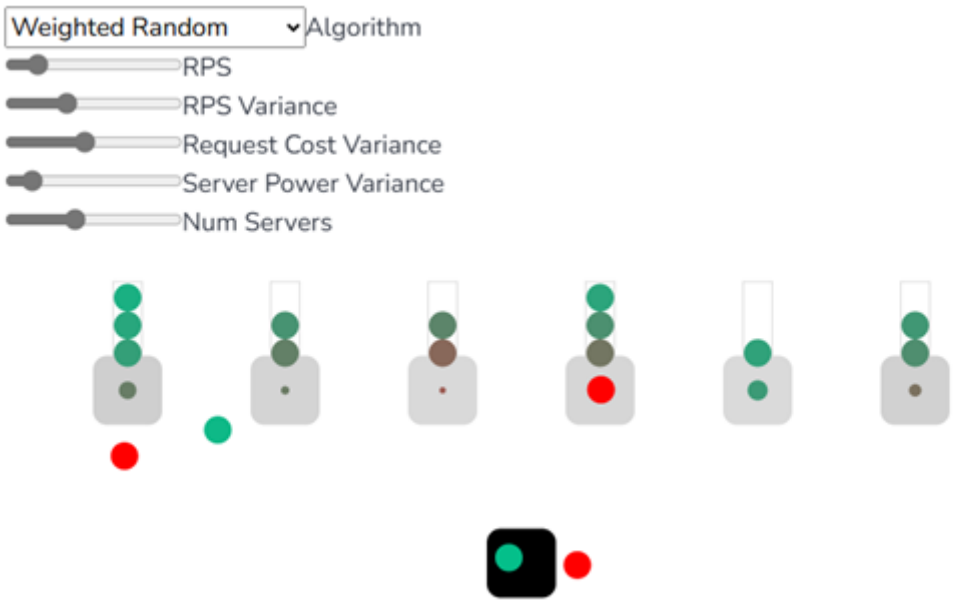

Figura 3: Weighted Random

Ventajas

- Permite manejar diferentes probabilidades de selección.
- Es útil en diversas áreas, desde publicidad hasta juegos y sistemas distribuidos.

Weighted Round Robin

A diferencia del Round Robin clásico, este método funciona en distribución ponderada: a cada servidor se le asigna por adelantado un valor en función de sus capacidades y su potencia.(4)

Funcionamiento

- Cada servidor o cola tiene un peso asociado.
- En lugar de enviar solicitudes de manera equitativa, el WRR ofrece a cada servidor un número fijo de oportunidades de servicio, según su peso configurado.
- Si todos los paquetes tienen el mismo tamaño, el WRR es una aproximación simple del "generalized processor sharing" (GPS).

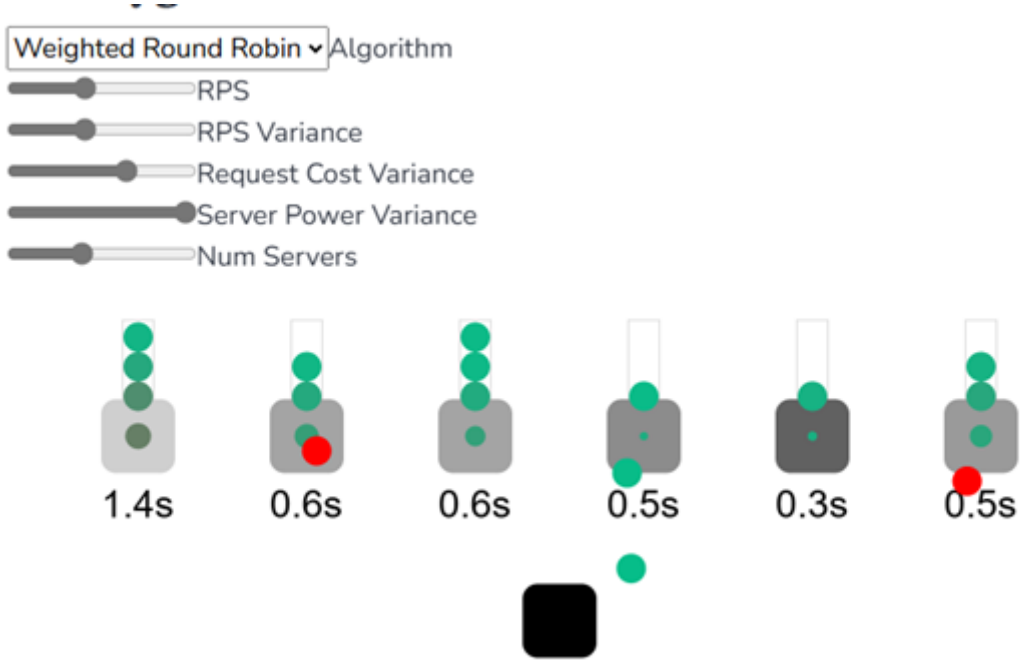

Figura 4: Weighted Round Robin

Ventajas

- Maneja mejor la variabilidad en la potencia de los servidores en comparación con el round robin simple.
- Útil para balancear cargas en redes y sistemas.

Least Connections

El algoritmo Least Connections asigna solicitudes al servidor con menor número de conexiones activas en un momento dado. Su principal ventaja radica en su capacidad para distribuir las solicitudes de manera equitativa, lo que lo hace ideal para manejar variaciones significativas en la carga de trabajo, como diferencias en el poder de los servidores o en la complejidad de las solicitudes. Este algoritmo sobresale por su simplicidad y eficiencia.

Figura 5: Least Connections

Peak EWMA

El algoritmo Peak EWMA es una técnica de balanceo de carga que utiliza una media móvil ponderada exponencialmente para estimar la latencia de los servidores. Al dar mayor peso a las observaciones recientes, este algoritmo puede adaptarse rápidamente a cambios en el rendimiento del servidor, lo que lo hace ideal para entornos dinámicos donde la latencia puede variar frecuentemente. (2)

Funcionamiento

- Este algoritmo utiliza una media móvil exponencialmente ponderada (EWMA) para estimar la latencia de los servidores y toma decisiones de balanceo de carga en función de estas estimaciones.
- El algoritmo se enfoca en los picos de latencia, ponderando más las últimas observaciones. De esta manera, es capaz de reaccionar rápidamente a cambios en la latencia de los servidores.

- A medida que se reciben las solicitudes, el algoritmo asigna la carga al servidor con la latencia estimada más baja, según la EWMA.

Caracteristicas principales

- Respuesta rápida a cambios en la latencia.
- Mayor precisión en entornos donde la latencia varía frecuentemente.
- Es sensible a picos de carga y ajusta el balanceo para evitar saturar servidores.

Pérdidas de requerimientos

- Las pérdidas pueden ocurrir si los servidores seleccionados tienen latencias fluctuantes o si el algoritmo responde a picos temporales de latencia de manera demasiado agresiva, sobrecargando otros servidores.
- Es posible que se presenten pérdidas si el algoritmo sobreestima la capacidad de un servidor debido a un cálculo incorrecto en la media móvil, especialmente si hay picos de carga imprevistos.

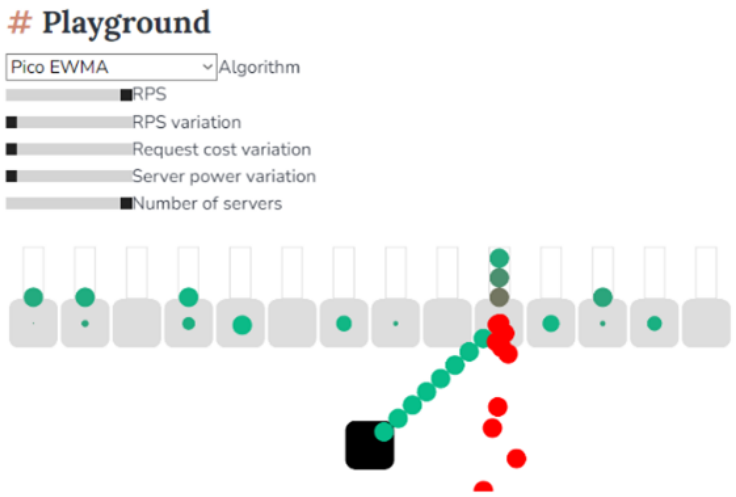

Figura 6: Peak EWMA

Interpretación de Pérdidas

Se puede aumentar la carga de solicitudes o introducir cambios bruscos en la latencia de algunos servidores para observar cómo el algoritmo reacciona y si conduce a pérdidas.

Random

El algoritmo Random distribuye las solicitudes entrantes de manera aleatoria entre los servidores disponibles. Es un método extremadamente simple y rápido de implementar, ya que no considera métricas de rendimiento ni el estado actual de los servidores. Su eficacia depende de la uniformidad de la carga a lo largo del tiempo, pero puede llevar a una distribución desigual y a sobrecargas si la aleatoriedad no favorece el equilibrio. (3)

Funcionamiento

- Este algoritmo asigna las solicitudes entrantes a un servidor de manera aleatoria, sin considerar la carga actual o la latencia.
- No hace suposiciones sobre el estado de los servidores; simplemente distribuye la carga de manera equitativa basada en la aleatoriedad.

Características Principales

- Simplicidad en la implementación.
- Distribución equitativa de las solicitudes en un promedio, pero sin optimización.
- No considera la capacidad actual ni el estado de los servidores Perdidas de requerimientos

Pérdidas de Requerimientos

- Las pérdidas pueden ocurrir si el algoritmo asigna demasiadas solicitudes a un servidor ya saturado, dado que no se consideran métricas como la latencia o la carga actual.
- La aleatoriedad puede causar una distribución desigual temporal, donde algunos servidores pueden ser sobrecargados mientras otros permanecen subutilizados, generando así pérdidas de solicitudes en los servidores saturados.

Figura 7: Random

Interpretación de Pérdidas

Incrementa la cantidad de solicitudes para todos los servidores hasta un punto donde algunos servidores empiezan a saturarse, lo que provocará pérdidas debido a la asignación aleatoria que podría sobrecargar servidores específicos.

Referencias

[1] Load balancing. (s.f.). `https://samwho.dev/load-balancing/`

[2] Doe, J., Smith, J. (2023). Adaptive Load Balancing Using Peak EWMA in Distributed Systems. Journal of Distributed Computing, 35(4), 567-578. `https://example.com/peak_ewma_article`

[3] Johnson, A., Lee, B. (2022). Load Balancing Algorithms and Techniques (2nd ed.). Tech Publishing. `https://example.com/random_balancing_book`

[4] ¿Qué es el load balancing? (s/f). OVHcloud. Recuperado el 16 de agosto de 2024, de `https://www.ovhcloud.com/es/public-cloud/what-load-balancing/`

[5] Understanding the weighted random algorithm. (2023, octubre 24). DEV Community. `https://dev.to/jacktt/understanding-the-weighted-random-algorithm-581p`