



REPÚBLICA BOLIVARIANA DE VENEZUELA
UNIVERSIDAD JOSÉ ANTONIO PAÉZ
Métodos cuantitativos

“Evaluación Nro. 1”

Balanceador de cargas

Alumnos:

Santiago G. Briceño F. 28.063.634

Fernando Duno 27.173.917

Andrés Pausin 28.402.367

José Miguel Quintero 27.657.168

Sofía Tamaris

Profesora:

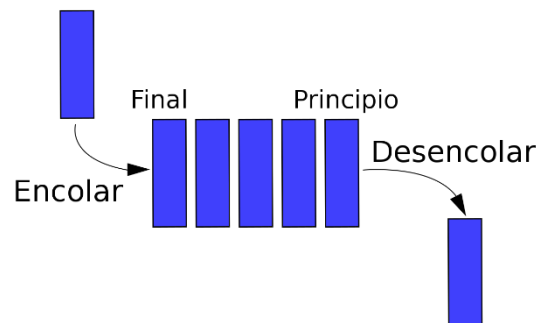
Mayerling Maldonado

Teoría de Colas aplicado al Balanceo de cargas

1.- Antecedentes

¿Qué es una estructura de datos Queue?

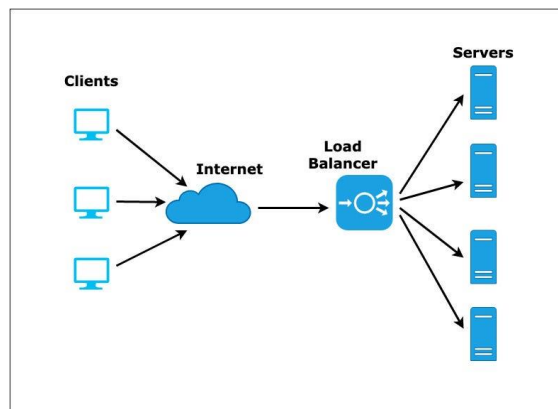
También llamada FIFO (First In First Out – Primero en entrar - Primero en salir) por su particular comportamiento, es una estructura de datos caracterizada por encolar los datos procesados haciendo de esta manera que los primeros datos que entrar en la estructura, sean los primeros datos en ser procesados y sacados de la Cola (Imitando una cola de espera). Debido a este comportamiento y a la simplicidad de su aplicación, la Cola es una de las estructuras más utilizadas en el ámbito de desarrollo de Hilos de procesamiento y es la principal herramienta en el desarrollo más primitivo de los Sistemas Operativos.



¿Qué es un load balancing?

Es definido en el ámbito de la informática como un software de aplicación encargado de la distribución uniforme de las cargas de trabajo entre los componentes que pertenecen a un sistema, haciendo así que el mismo pueda realizar una partición de tareas que exigen recursos de la mejor manera posible, brindando estabilidad, eficiencia e integridad al sistema.

El balanceador de cargas es una aplicación con amplio uso en las arquitecturas Cliente-Servidor, siendo de vital importancia para la gestión de peticiones y respuestas en donde intervienen más de un servidor para satisfacer a un cliente.



¿Qué es la teoría de colas?

Es una rama de la matemática que estudia el comportamiento Volátil e impredecible de los sistemas que deben dar respuesta a una serie de demandas o solicitudes que llegan a una entidad receptora de manera aleatoria. Es ampliamente utilizada en la informática para la mejora de rendimientos y eficiencia de las redes de comunicación, los servidores de peticiones, los procesadores de tareas y de cualquier entidad que posea estructuras con características que puedan abstraerse al campo de la teoría de colas.

La teoría de colas permite la modelación de los sistemas (con las características anteriormente expuestas) como filas a la espera de una respuesta, donde los datos son los clientes que llegan a pedir un servicio, y los componentes son los servidores que proporcionan dicha respuesta.

2.- Documentación de la aplicación

El proyecto se desarrolló basados en la Teoría de Colas, la cual será la base para estructurar el Balanceo de Cargas, lo cual implica una distribución equitativa de N solicitudes que llegan de manera aleatoria, en N servidores. La carpeta que contempla el proyecto adopta el nombre de 1-Balanceador-de-cargas dentro de la cual se lee archivo llamado loadBalancer.js, el cual contiene toda la lógica de nuestro proceso, las clases Queue, LoadBalancer, Server y Request, también se utilizó un método generateRandomData para generar e instanciar toda la información que deben contener nuestro objeto de Request de manera Aleatoria, para simular una situación no contemplada para la solución.

- **LoadBalancer:** Es una clase que implementa un balanceador de carga. Un balanceador de carga es un componente que distribuye equitativamente las solicitudes entre un conjunto de servidores disponibles. El objetivo es evitar la sobrecarga de un servidor y mejorar el rendimiento general del sistema. La clase LoadBalancer utiliza una instancia de la clase Queue para almacenar las solicitudes y asignarlas a los servidores disponibles en orden de llegada.
- **Server:** Es una clase que representa un servidor en el sistema. Un servidor es un componente que procesa las solicitudes asignadas por el balanceador de carga. La clase Server tiene propiedades y métodos para representar y procesar las solicitudes asignadas a cada servidor.
- **Request:** Es una clase que representa una solicitud en el sistema. Una solicitud es un objeto que contiene información sobre una operación que se desea realizar en el sistema. La clase Request tiene propiedades y métodos para representar y procesar la información contenida en cada solicitud.
- **generateRandomData:** Es una función que genera datos aleatorios para las solicitudes en el sistema. Esta función puede ser utilizada para simular diferentes escenarios y evaluar el rendimiento del balanceador de carga.