UNIVERSIDAD TECNICA DE BABAHOYO

FACULTAD DE ADMINISTRACION FINANZAS E INFORMATICA

PROCESO DE TITULACION

MAYO 2023-SEPTIEMBRE 2023

EXAMEN COMPLEXIVO DE GRADO O DE FIN DE CARRERA

PRUEBA PRACTICA

PREVIO A LA OBTENCION DEL TITULO DE:

INGENIERA EN SISTEMAS DE INFORMACION

TEMA:

ANALISIS COMPARATIVO DE LOS ALGORITMOS DE BALANCEO DE CARGAS ROUND ROBIN Y LEAST CONNECTION DE HA-PROXY EN LOS SISTEMAS DE GESTION DE BASE DE DATOS

ESTUDIANTE:

MARIUXY TATIANA VASQUEZ FIGUEROA

TUTOR:

ING. FABIAN EDUARDO ALCOSER CANTUÑA

AÑO 2023

Índice

RESUMEN

El balanceo de cargas nace como una estrategia critica para optimizar el rendimiento y la escalabilidad de los sistemas de gestión de base de datos, distribuyendo el tráfico de manera equitativa entre los servidores, el objetivo de este estudio, es comparar el rendimiento de los algoritmos de balanceo de cargas Round Robin y Least Connection de ha-proxy en sistemas de gestión de base de datos para evaluar su eficiencia en diferentes escenarios, haciendo uso de la investigación experimental y comparativa, también se usó los métodos cualitativo y bibliográfico, para abordar experiencias, perspectivas de las personas y la corroboración de fuentes confiables para llevar a cabo la investigación, con la utilización de técnicas como la encuesta, población y muestra para sustentar el proyecto. Este estudio permito evaluar y comprar los algoritmos Round Robin y Least Connection, en donde fueron configurados en 4 escenarios diferentes, dando como resultado que el algoritmo Least Connection fue el que funciono de mejor manera en términos de tiempo transcurrido, tasa de transacciones y rendimiento, teniendo un mayor porcentaje de rapidez en respuesta y menor consumo de recursos en los servidores, porque las encuestas realizadas a los usuarios se sugiere tener un enfoque para mejorar los servicios de las y repuesta, mientras que los profesionales afirman mejorarlo con la implementación de algoritmos como los de ha-proxy.

Palabras claves: Ha-proxy, Round Robin, Least Connection, Disponibilidad, Comparación.

PLANTEAMIENTO DEL PROBLEMA

Actualmente, el internet desempeña un papel fundamental en la vida de las personas, revolucionando varios aspectos importantes, convirtiéndose en un medio global entre las comunicaciones de la vida cotidiana; ligándose estrechamente al uso de páginas web dinámicas que son altamente solicitadas en distintos sectores como organizaciones o negocios. Teniendo la función de simplificar diversos procesos que antes era necesario realizarlos de manera presencial, como solicitar información personal, realizar pagos de seguros, obtener datos sobre familiares, entre otras funciones en sitios web gubernamentales, también se menciona que los sitios web que se dedican a las ventas de productos (AMAZON, ALIBABA, SHOPIFY, SHEIN, ETC) son altamente utilizadas para realizar compras en línea sin necesidad de la presencia física en una tienda, facilitando los pagos con tarjetas de crédito o débito.

Las páginas web dinámicas y los sistemas de gestión de base de datos son esenciales en muchas aplicaciones y sitios web actuales, ya que estos softwares son utilizados principalmente para administrar y organizar grandes cantidades de información en una base de datos, permitiendo realizar inserciones, consultas, actualizaciones y eliminación de datos, garantizando la integridad y seguridad de la información.

La problemática de este caso de estudio está ligada explícitamente a las solicitudes que hacen los clientes a servicios públicos o privados que ayudan a optimizar los procesos presenciales o manuales de la vida diaria, teniendo acceso en línea y solicitando diferentes tipos de información desde la comodidad de sus hogares, pero en diversas ocasiones el acceso de esta información se colapsa, ocasionando la suspensión del servicio por periodos, debido a que cuando hay múltiples peticiones a la base de datos, hace que se reduzca la velocidad en emitir

una respuesta al solicitante o maquina cliente, dando paso a que se presente la sobrecarga y tráfico de información, ocasionando que el servidor no pueda administrar todas las solicitudes de forma eficiente, en la cual se presente en tiempos de respuesta lentos u ocasionar bloqueos de conexión y con ello, el consumo excesivo de recursos, al presentarse este inconveniente en las organizaciones que prestan servicios por medio de la web, ocasiona que tengan perdidas graves o molestias a nivel de sus clientes al no tener una respuesta de lo solicitado.

Entre los problemas que manifiestan los usuarios están los retrasos en la obtención de respuestas en donde esta problemática afecta en gran parte la experiencia de los usuarios y la calidad de los servicios que prestan las instituciones a nivel general, ya que también ocasiona que los usuarios abandonen el sitio web cuando estos problemas persisten frecuentemente, resultando en la insatisfacción de los clientes que requieren servicios en línea o de manera local, conllevando a la deserción de dichos sistemas debido a la ineficiencia de los servidores de base datos, conllevando a la perdida de datos , interrupción en los servicios en línea y retraso en la entrega de productos o servicios, teniendo un impacto negativo en la productividad de las organizaciones.

JUSTIFICACIÓN

Hoy en día, la gestión adecuada de grandes volúmenes de datos es fundamentales para distintos tipos de organizaciones que dependen de sistemas de base de datos cumpliendo un rol importante, que es el almacenar y dar acceso a información. A medida que los datos continúan creciendo exponencialmente al pasar el tiempo, los sistemas de base de datos deben ser capaces de distribuir la carga de trabajo de una manera óptima, para así poder garantizar un buen rendimiento y una alta disponibilidad de la información solicitada por los clientes ya sea en línea o de manera local.

El balanceo de cargas emerge como una estrategia crítica para optimizar el rendimiento y la escalabilidad de los sistemas de gestión de base de datos, siendo una técnica esencial que permite distribuir el tráfico de las solicitudes de manera equitativa entre los diferentes nodos o servidores que componen un clúster de base de datos, para asegurar que ningún servidor este sobrecargado.

El análisis comparativo de los algoritmos de balanceo de cargas como Round Robin y Least Connection se convierte en un tema de investigación crucial, ya que la selección de un algoritmo adecuado puede tener un impacto positivo en la distribución equitativa de las solicitudes.

La comparación y análisis exhaustivo de estos dos algoritmos en entornos de sistemas de gestión de base de datos (SGBD) permite una evaluación objetiva en términos de rendimiento, eficiencia y capacidad de manejo de cargas variables, permitiendo identificar las situaciones en las que cada algoritmo puede sobresalir y brindar recomendaciones para la implementación de estos algoritmos que proporciona el balanceador Ha-proxy para que los servidores tengan disponibilidad en todo momento y puedan solventar todos las solicitudes de manera exitosa.

OBJETIVOS DEL ESTUDIO

Objetivo General

- Comparar el rendimiento de los algoritmos de balanceo de cargas Round Robin y Least Connection de ha-proxy en sistemas de gestión de base de datos para evaluar su eficiencia en diferentes escenarios.

Objetivos específicos

- Analizar la implementación de ha-proxy y los algoritmos de balanceo de cargas para realizar un laboratorio.
- Establecer los algoritmos de balanceo de cargas Round Robin y Least connection de HA-proxy en escenarios diferentes para realizar pruebas y medir el rendimiento de los sistemas de gestión de base de datos.
- Evaluar el uso de recursos como CPU y memoria en los servidores de bases de datos al implementar ambos algoritmos de balanceo de cargas.

LÍNEAS DE INVESTIGACIÓN

La carrera de Ingeniería en Sistemas de Información de la Universidad Técnica de Babahoyo en la facultad de Administración Finanzas e Informática aprobó el tema de: Análisis comparativo de los algoritmos de balanceo de cargas Round Robin y Least Connection de Ha-proxy en los sistemas de gestión de base de datos, en correlación con la siguiente línea y sublínea de investigación;

Línea de investigación

Sistemas de información y comunicación e innovación.

Sublínea de investigación

Redes y tecnología inteligentes de software y hardware.

La línea de investigación se relaciona con el análisis de los algoritmos de balanceo de cargas con ha-proxy, enmarcándose en la innovación y comunicación, basándose en el análisis de la distribución de solicitudes a los servidores de bases de datos, haciendo una comunicación más efectiva entre los usuarios y servidores.

La sublínea de investigación se alinea perfectamente con el tema propuesto ya que proporciona enfoques en el tiempo de respuesta y la capacidad de carga de los sistemas de base de datos para evaluar la eficiencia de los algoritmos en la red mediante el uso de tecnologías inteligentes para ser más eficientes en la gestión de comunicación y sus recursos.

ARTICULACIÓN DEL TEMA

En el desarrollo de este caso de estudio, se procederá a hacer un análisis de los algoritmos de balanceo de cargas Round Robin y Least Connection de Ha-proxy en los sistemas de gestión de base de datos, con el fin de mejorar el rendimiento de respuesta de los servicios de la web que se conectan con sistemas de gestión de base de datos, para evitar la sobrecarga de los servidores, en el cual se explicara la relevancia de los balanceadores y por ende la forma en que se pueden aplicar dependiendo de los requisitos de los servidores, para así poder lograr una forma más distribuida para que puedan resolver las solicitudes de los usuarios.

Como tema clave tenemos: Balanceo de cargas con Ha-proxy; como antes mencionado, el enfoque principal de este tema, es analizar los algoritmos Round Robin y Least Connections que actúan distribuyendo las cargas, haciendo que el tiempo de respuesta sea óptimo para el usuario o maquina cliente.

Como tema de prácticas preprofesionales tenemos; aplicación de tecnologías en el sector público y privado; el balanceo de cargas Ha-proxy tiene una estrecha relación con la tecnología, ya que, en este punto se destaca la relevancia de las aplicaciones tecnológicas avanzadas como los balanceadores de cargas, siendo aplicados tanto en el sector público o privado, en la cual se analiza como esta tecnología puede mejorar la disponibilidad y rendimiento de los sistemas de bases de datos en diferentes organizaciones.

MARCO CONCEPTUAL

Hoy en día, estamos en plena era digital, haciendo que el internet este estrechamente relacionado con el uso de las bases de datos, transformando la forma en que interactuamos con la información, donde las bases de datos son sistemas diseñados para almacenar, organizar y gestionar datos de manera estructurada, mientras que internet es una red global que conecta miles de dispositivos permitiendo la comunicación e intercambio de información que se puede acceder desde cualquier lugar del mundo, dando lugar a aplicaciones web y servicios en línea que almacenan miles de datos que son accesibles a través de un navegador.

¿Que son los sistemas de gestión de base de datos (SGBD)?

Un SGBD es un software diseñado para administrar eficientemente grandes volúmenes de datos, en la cual permite a los usuarios almacenar, manipular y consultar información de manera segura y organizada.

Estos sistemas brindan una interfaz entre los usuarios y la base de datos, facilitando la creación, modificación, eliminación y búsqueda de datos.

Según (Tesone et al., 2021) Los sistemas de gestión de bases de datos (DBMS, por su sigla en inglés correspondiente a Database Management. System) juegan un rol fundamental en el desarrollo de software desde su surgimiento en la década de 1960, ya que proveían una forma eficiente de generar aplicaciones complejas, al eliminar la necesidad de programar la persistencia y el acceso a los datos.

Por mención de lo citado, se resalta la manera en que los sistemas de gestión de base de datos han sido una herramienta fundamental para el desarrollo de software o aplicaciones web, en la cual, hace posible la manipulación de grandes cantidades de datos, haciendo está, una manera eficiente al ofrecer una estandarización de gestionar datos sin programar la persistencia y

el acceso a los mismos, contribuyendo grandemente al avance tecnológico de la información en general.

Los **Sistemas de gestión de base de datos** ofrecen varias funcionalidades, como:

- **Creación y definición de la estructura de datos**: los SGBD permiten definir y crear tablas, campos y relaciones entre los datos, estableciendo la estructura y el esquema de la base de datos.
- **Almacenamiento y organización de los datos**: Los SGBD gestionan como se almacenan los datos en el sistema, ya sea en archivos o en formatos más sofisticados como bases de datos relacionales, orientadas a objetos o basadas en documentos.
- **Control de acceso y seguridad:** Estos ofrecen un mecanismo de autenticación y automatización para garantizar que solo los usuarios autorizados puedan acceder y modificar los datos.
- **Consultas y manipulación de datos**: Los usuarios pueden utilizar lenguajes de consulta, como SQL, es decir por medio de sistemas de gestión de base de datos estos pueden realizar consultas y obtener información específica de la base de datos.
- **Mecanismos de integridad de los datos:** Los SGBD aseguran la consistencia y la integridad de los datos mediante la aplicación de restricciones, validaciones y reglas definidas en el esquema de la base de datos; entre los más populares están Oracle, MYSQL, Microsoft SQL Server, PostgreSQL y MongoDB.

Importancia de los Sistemas de gestión de base de datos

Según la (Universidad Europea, 2021) estos sistemas son sumamente importantes

ya que son fundamentales para garantizar un almacenamiento organizado, tener una buena consistencia de datos, acceso concurrente, seguridad y apoyo a la toma de decisiones en las organizaciones.

¿Qué es un servidor?

Un servidor es básicamente una computadora o sistema informático que proporciona servicios, recursos o funciones específicas a otras computadoras o dispositivos conocidos como clientes, en la cual están diseñados para estar siempre encendidos y funcionando para brindar accesos constantes dependiendo de los servicios que estos ofrecen.

Cabe mencionar que los servidores, desempeñan un papel fundamental hoy en día, ya que a medida que ha avanzado la tecnología y con ella el uso del internet de una manera masiva, los servidores tienen gran parte de este funcionamiento, ya que existen diferentes servidores, para que las páginas en internet funcionen a toda hora, es decir; existen servidores web; que son los encargados de mantener las aplicaciones web en la red o internet, servidor de correo electrónico; que se basa en la recepción y entrega de correos electrónicos, permitiendo el intercambio de mensajes entre usuarios, servidores de archivos; se encargan de almacenar archivos y documentos compartidos en la red, servidores de bases de datos; estos se encargan de administrar y almacenar datos permitiendo el acceso y manipulación de la información de la base de datos, servidores de aplicaciones; son básicamente el encargado de ejecutar aplicaciones y software en un entorno centralizado permitiendo que los usuarios accedan a ellas, estos y un sinnúmero de servidores que dan diferentes servicios hacia los usuarios.

Según (Marchionni, 2019, p. 23) los servidores, son equipos informáticos que brindan un servicio en la red, dando información a otros servidores y a los usuarios, teniendo una mayor potencia y dimensión que una PC de escritorio con mayores características.

¿Qué es un balanceador de carga?

Para (Sanchez Calderon y Guzman Pineda, 2021)El balanceo de carga provee alta accesibilidad por medio de la detección automática de fallas de servidores y la repartición del tráfico de clientes a través de los servidores restantes alrededor de los 10 segundos, mientras proveen usuarios con servicio continuo.

En los sistemas de gestión de base de datos un balanceador de cargas ayuda a distribuir el tráfico de peticiones de los usuarios, haciendo que estos sean capaces de emitir respuestas de manera rápida sin que los servidores se colapsen o se sobrecarguen por las múltiples peticiones.

También (Teixeira, 2019, p. 5) dice que son procesos a través del cual el tráfico saliente es distribuido por múltiples enlaces, distribuyendo las cargas a diversos servidores donde estas sean equitativas.

El balanceo de cargas es conocido como la distribución de cargas entre servidores, basándose en el modelo OSI de 7 capas; comienza desde la capa 4 (transporte) que es la información de cabecera de paquetes, como puertos para realizar encaminamientos, extendiéndose hasta la capa 7 (aplicación), efectuando un análisis profundo del contenido real de las solicitudes, operando en diversos niveles para emplear técnicas algorítmicas, como en este caso Round Robin o Least Connection.

¿Qué es ha-proxy?

Ha-proxy, técnicamente es un balanceador de cargas de código abierto y de alta disponibilidad, en el cual es muy utilizado para distribuir el tráfico de una red de varios servidores o recursos que dan servicio a clientes, es decir, son utilizados en servidores web, servidores de aplicaciones o bases de datos, teniendo su función como un intermedio entre el cliente y los servidores.

Ha-proxy optimiza el rendimiento de los sistemas y asegura que los recursos estén utilizándose de manera eficiente, en el cual, ha-proxy ofrece una alta gama de características, como el balanceo de cargas basado en algoritmos, la monitorización de servidores, la gestión de sesiones y la alta disponibilidad de los servicios, estas características ayudan a que esta herramienta sea muy flexible y escalable para la gestión del tráfico de las diferentes bases de datos.

Para (Mejia Viteri et al., 2018, p. 100) Ha-proxy es una solución gratuita, muy rápida y fiable que ofrece alta disponibilidad y permite conmutaciones de balanceo de carga en aplicaciones, (Bastidas Delgado, 2021, p. 18) El papel de un equilibrador de carga a veces se compara con el de un policía de tráfico, ya que está destinado a enrutar sistemáticamente las solicitudes a los lugares correctos en cualquier momento dado.

También (Carlos Navarro, 2018) destaca que los algoritmos de asignación y cargas dinámicos logran equilibrar clusters de servidores heterogéneos al considerar la carga y los servicios ofrecidos.

Por otro lado, (Codig, 2022) indica que las solicitudes entrantes pueden dirigirse de servidores sobrecargados a aquellos con más recursos disponibles para una atención efectiva.

Tipos de balanceadores de carga

Los balanceadores de carga son una herramienta esencial en la gestión eficiente de tráfico, donde presentan una variedad de tipos y configuraciones, desempeñando un papel crucial al distribuir solicitudes de manera equitativa entre servidores, mejorando la disponibilidad y el rendimiento de sus recursos.

En base a esta introducción, se mencionarán los diferentes tipos de balanceadores de carga, destacando como cada uno aborda diversas situaciones como:

Tipo de Balanceador de Carga	Descripción
Balanceador de carga basado en hardware	Son dispositivos físicos especializados en el balanceo de carga. Están diseñados para manejar grandes cantidades de tráfico y distribuir la carga entre múltiples servidores. Suelen ofrecer altos niveles de rendimiento y escalabilidad.
Balanceador de carga basado en software	Se ejecutan como aplicaciones en servidores estándar. Proporcionan balanceo de carga utilizando soluciones de software y son más flexibles en términos de configuración y personalización. Son adecuados para entornos más pequeños o medianos.
Balanceador de carga de DNS	Distribuye las solicitudes de los clientes a diferentes direcciones IP basadas en la resolución DNS. Es más adecuado para sistemas con múltiples ubicaciones geográficas.
Balanceador de carga de Capa 4 (TCP/UDP)	Opera en la capa de transporte (Capa 4) y equilibra la carga en función de la dirección IP y el puerto de destino. Es más rápido y eficiente que los balanceadores de capa 7, pero no es capaz de tomar decisiones basadas en el contenido de la solicitud.
Balanceador de carga de Capa 7 (HTTP/HTTPS)	Opera en la capa de aplicación (Capa 7) y puede tomar decisiones basadas en la información del encabezado HTTP, como la URL, las cookies o los datos del usuario. Es más inteligente y adecuado para aplicaciones web, ya que permite un enrutamiento más sofisticado.
Balanceador de carga global	Distribuye las solicitudes a través de múltiples ubicaciones geográficas y puede direccionar a los usuarios al servidor más cercano o más adecuado en función de su ubicación y las condiciones de la red.
Balanceador de carga local	Se encuentra dentro de una red local y equilibra la carga entre los servidores de esa red. Es adecuado para aplicaciones que operan dentro de una empresa o un centro de datos específico.

Balanceador de carga basado en algoritmos	Utiliza algoritmos específicos para distribuir la carga entre los servidores. Los algoritmos comunes incluyen Round Robin, Least Connections (menos conexiones), IP Hash, entre otros.

Tabla 1: Descripción de los balanceadores de carga. Autora: Mariuxy Vásquez

Como se recalca en la tabla, cada balanceador tiene una función específica y se relaciona con un área específica, para este caso de estudio se toma el balanceador de carga basado en algoritmos, en el cual se tomará como enfoque el Round Robin y Least Connection con el fin comparar sus características y respuestas en los sistemas de gestión de base de datos.

Algoritmos del balanceo de carga con Ha-proxy

Ha-proxy ofrece varios algoritmos para el balanceo de cargas, cada uno de ellos con diferentes enfoques para distribuir las solicitudes entre los servidores, en ese caso se toman dos en cuenta que son:

- **Round Robin o (Rodeo):** este algoritmo distribuye las diferentes solicitudes de clientes en un ciclo secuencial por medio de los servidores backend disponibles; cada una de las solicitudes se envían al siguiente servidor en la lista, se puede decir que cuando llega al último servidor, la distribución vuelve al principio, siendo una opción equitativa para que los servidores reciban aproximadamente la misma cantidad de solicitudes.

Para (Chunga Zuloeta y Chuzon Sanchez , 2019, p. 27) distribuye las conexiones a cada uno de los servidores activos una por una, así el tiempo de ejecución de la distribución de conexión, es en forma homogénea, con una fórmula de: Tiempo de proceso, tiempo del último proceso, tiempo del próximo proceso.

¿Pero, que son servidores Backend?

El backend es la parte de un sistema informático que opera del lado del servidor, encargándose de procesar, gestionar los datos, la lógica y la funcionalidad del sistema

interpretando las solicitudes hacia un servidor de base de datos u otros. Es decir, el backend gestiona el funcionamiento interno y las comunicaciones necesarias para que una aplicación se ejecute de manera efectiva.

Según (Perez Ibarra et al., 2021)Se denomina BackEnd a la capa oculta de acceso a los datos de un software, teniendo toda la lógica de la aplicación que maneja los datos de una aplicación, estos se encuentran almacenados en una base de datos dentro de un servidor.

También (Perreto et al., 2021, p. 22) aporta que; básicamente es la conexión del frontend con el backend es un tipo de interfaz, es decir, una conexión funcional entre dos sistemas, programas, dispositivos o componentes de cualquier tipo, que proporciona una comunicación de distintos niveles, permitiendo el intercambio de información.

Cabe mencionar que el algoritmo Round Robin tiene dos algoritmos principales por turno, que son:

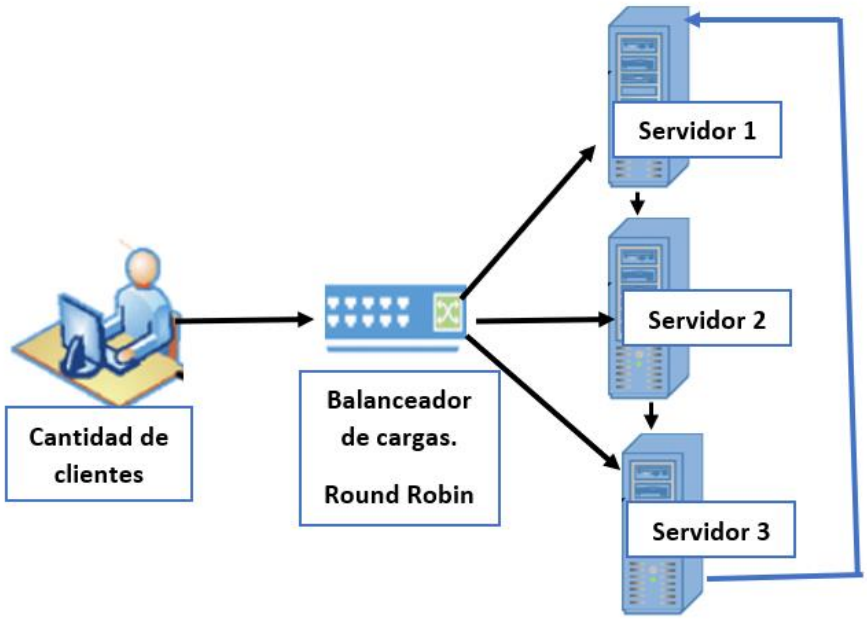
- **Round Robin con peso;** esta es una variante del algoritmo Round Robin, que tiene en cuenta la capacidad o eficiencia diferencial de los servidores en un grupo, es decir, en lugar de asignar una carga igual a cada servidor, el algoritmo permite asignar pesos o valores de ponderación diferentes a cada servidor en función de sus capacidades individuales.

  se podría decir que la idea de este algoritmo es asegurarse de que los servidores más potentes manejen una carga con mayor peso.
- **Round Robin dinámico;** esta variante toma como factor principal la carga de trabajo actual de cada servidor, el tiempo de respuesta , el uso de recursos o cualquier otro indicador relevante para lograr determinar que servidor será seleccionado para manejar la próxima solicitud, teniendo como objetivo asegurar que las solicitudes se envíen al

servidor más adecuado; en otras palabras el round Robin dinámico ayuda a enviar las solicitudes a un servidor según el peso del tiempo de ejecución o respuesta.

Arquitectura round Robin

Round Robin tiene un enfoque simple pero efectivo para la distribución de las solicitudes, es decir tiene varios servidores de destino, en el cual el balanceador de cargas actúa como intermediario entre los clientes que proceden a realizar las solicitudes y el servidor; el balanceador de cargas contiene una lista de servidores y cada uno de estos servidores tiene una dirección IP o un nombre de host para ser reconocido o diferenciado, en la cual el balanceador contiene o actúa con una distribución secuencial, es decir, cuando una solicitud llega al balanceador de cargas este asigna al primer servidor, cuando este, está ocupado y llega otra solicitud procede a asignar al siguiente servidor en la lista de manera secuencial, una vez que se



ha distribuido una solicitud a un servidor ese servidor pasa al final de la lista.

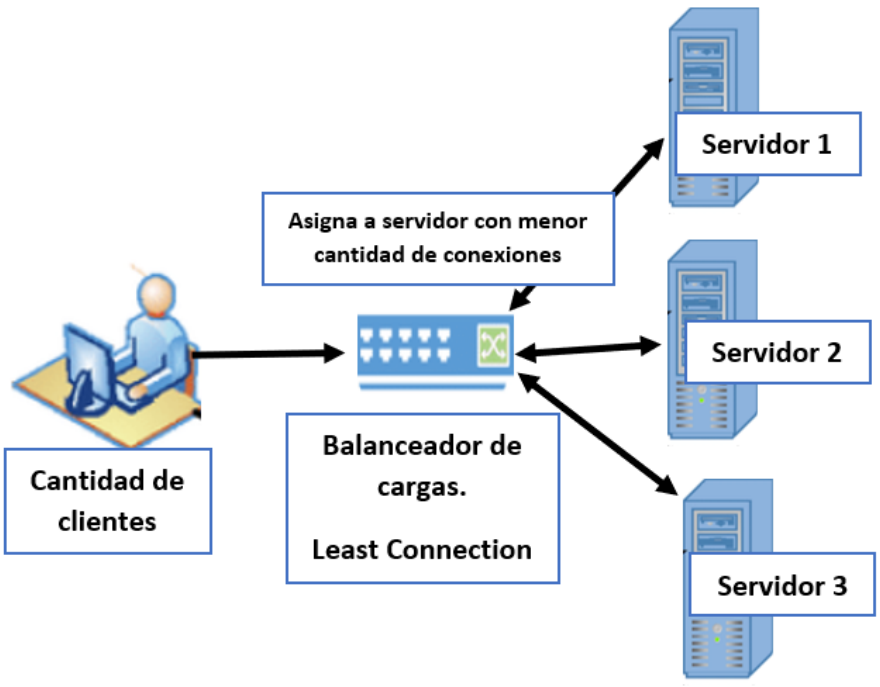
Gráfico 1: Balanceo Round Robin de manera secuencial

- **Least Connection (menos conexiones);** Este algoritmo de ha-proxy se basa en enviar las diferentes solicitudes al servidor con menos cantidad de conexiones abierta, este algoritmo es útil cuando los servidores tienen capacidades de rendimiento diferentes, ya que se enfrasca en distribuir el tráfico en función de la carga de trabajo actual de cada servidor.

Según (Llano Miraval y Castellanos Landazabal, 2021, p. 10)Least connection es un algoritmo de balanceo de carga dinámico, este distribuye los paquetes de datos al nodo con menor conexiones en el momento del pedido de transferencia es un método para seleccionar todos los elementos en un grupo dependiendo del número de conexiones que tenga cada servidor.

Arquitectura de Least Connection

Least Connection tiene otro enfoque, en el que utiliza la distribución de solicitudes entrantes entre múltiples servidores y recursos disponible, es decir este algoritmos se basa en la asignación de solicitudes al servidor con menor cantidad de conexiones activas, según su arquitectura; al igual que los otros algoritmos, se cuenta con varios servidores y el balanceador



de cargas actúa como intermediario; el balanceador de cargas realiza un seguimiento del número actual de conexiones en cada servidor, es decir, cada servidor tiene un contador que indica cuantas conexiones están siendo atendidas en ese momento.

Gráfico 2: Balanceo Least Connection, asigna solicitudes al servidor con menos cantidad de conexiones activas.

Tabla comparativa de Round Robin y Least Connection

En esta comparativa, se explorarán teóricamente los aspectos positivos o características de cada uno de estos algoritmos destacando sus ventajas y fortalezas particulares, permitiendo comprender mejor cuando y como implementar Round Robin y Least Connection en función de los requisitos y las características únicas de una infraestructura.

Características	Round Robin	Least Connection
Distribución Equitativa	Asigna igualmente a todos los servidores	Asigna según las conexiones activas de los servidores
Implementación Simple	Fácil de configurar y administrar	Relativamente sencillo de implementar
Tolerante a Fallas	Si un servidor falla, otros siguen funcionando	No sobrecarga servidores ya cargados
Escalabilidad	Funciona bien con incremento de servidores	Maneja dinámicamente el tráfico
Uso Uniforme de Recursos	Aprovecha todos los recursos disponibles	Distribuye carga según la capacidad
Latencia Predecible	Respuestas uniformes para solicitudes	Puede mejorar la latencia
Buena para Recursos Similares	Ideal cuando los recursos son	Distribuye en base a carga actual.

	equivalentes	
Balance Dinámico	Adecuado para situaciones con carga variable	Evita desbordamiento de servidores
Eficaz en Pequeñas Implementaciones	Apropiado para entornos más pequeños	Útil en entornos de alto tráfico

Tabla 2: Comparación de los dos algoritmos. Autora: Mariuxy Vásquez

MARCO METODOLÓGICO

DISEÑO DE INVESTIGACIÓN

El desarrollo de este caso de estudio, busca analizar los algoritmos Round Robin y Least Connection de Ha-proxy en sistemas de gestión de base de datos, para ello se empleará VirtualBox para comparar su rendimiento en equilibrio de cargas, destacando diferencia entre ambos en entornos de administración de base de datos.

TIPOS DE INVESTIGACION

Investigación experimental: Por medio de este tipo de investigación se procederá a crear un laboratorio de prueba virtualizado, haciendo uso de la herramienta VirtualBox, donde se crearán dos servidores de bases de datos, el servidor que actuara como balanceador de cargas y una máquina que actúe como cliente, para poder enviar las solicitudes al balanceador y así poder realizar pruebas utilizando las diferentes configuraciones de los algoritmos de balanceo de cargas Round Robin y Least Connection.

Investigación comparativa: Por medio del laboratorio que se realizará, nos permitirá hacer uso de la investigación comparativa, ya que aquí se procederá a tomar las estadísticas de ambos algoritmos con el finde de comparar ambos resultados para poder ser evaluados.

METODO

Método cualitativo: Este método tiene un enfoque en la compresión de fenómenos

partiendo de las experiencias y perspectivas de las personas, ayudando a obtener una comprensión detallada que permite abordar temas complejos y multifacéticos en esta investigación.

Método bibliográfico: Este método nos brinda un enfoque de investigación basado en la revisión y estudio de fuentes académicas, técnicas y literarias, con el fin de recopilar, evaluar y sintetizar información relevante proveniente de fuentes confiables.

TECNICAS

Población de los estudiantes: Para esta técnica se tomó en cuenta a 36 estudiantes de la carrera de Sistemas de Información de 8va semestre entre las secciones vespertina y matutina, para realizar la correspondiente encuesta.

Población de los profesionales: Para el correspondiente análisis de datos se tomó a 4 profesionales del área de sistemas, 2 profesionales pertenecientes a la Gobernación de Babahoyo, 1 perteneciente al departamento de sistemas del municipio del cantón Baba y un docente de la Universidad técnica de Babahoyo.

Encuestas: Estas encuestas se realizan con el objetivo de tener criterios y perspectivas de una población determinada para tener conocimiento de las experiencias que tienen al momento de hacer solicitudes hacia los servidores, es decir por parte del cliente. También se tomará en cuenta opiniones de expertos sobre la implementación de algoritmos para la distribución de cargas en servidores.

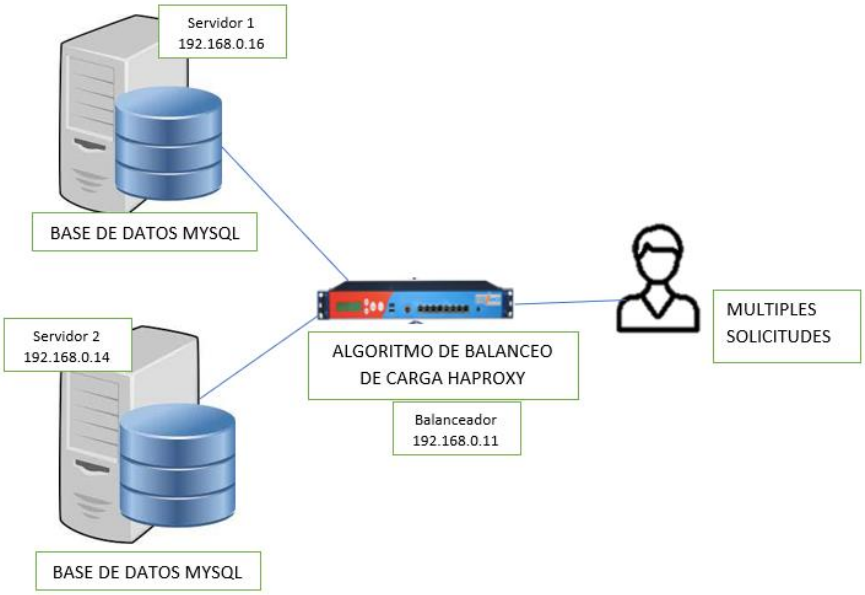
Observación: Aquí se procede a observar la manera y forma en que los algoritmos funcionan, esto incluye observar el tráfico de red, la distribución de carga y los tiempos de respuesta.

RESULTADOS

Como resultado de las encuestas realizadas a los estudiantes de la Universidad Técnica de Babahoyo, revelan fuertes problemas con respecto a sus experiencias en diversos sitios web, en donde han presentado lentitud, retrasos en respuesta y han estado en la posición de abandonar un sitio web cuando enfrentan problemas frecuentes, en donde, se resalta la importancia de abordar los problemas de rendimiento de los SGBD.

También, las encuestas realizadas a profesionales indican que hay un desafío grande en los sistemas de gestión de base de datos al momento de responder solicitudes, manifestando que el uso de un balanceador de cargas, como Ha-proxy tendrá un impacto positivo en el rendimiento y tiempo de respuesta mejorando la calidad de los servicios, pero, existe un desacuerdo igualitario en implementación de algoritmos, indicando que ambos cumplen sus funciones a cabalidad.

Previo al análisis de los algoritmos de balanceo de cargas Round Robin y Least Connection, se procedió a realizar esta investigación comparativa mediante el uso de la



herramienta de VirtualBox con la siguiente arquitectura.

Gráfico 3: Arquitectura aplicada en VirtualBox.

Se procede a crear tres máquinas virtuales con la versión de Ubuntu server 20.04, en la cual, dos de estas, actuaran como servidores de base de datos MYSQL y la otra como el balanceador de cargas Ha-proxy, siendo aquí donde se configurarán los diferentes algoritmos de balanceo de cargas que actuaran como intermediarios entre el usuario y los servidores.

Para él envió de solicitudes, se utilizó una maquina alterna que servirá como cliente, para realizar la simulación del envió de solicitudes hacia la IP del balanceador de cargas Ha-proxy

```
y:~$ siege -c 10 -r 30000  http://192.168.0.5/
```

con el siguiente comando:

En donde (-c) será la cantidad de usuarios, (-r 30000) se refiere a la cantidad de solicitudes que se enviaran y (http://192.168.100.75/) es la dirección IP del balanceador.

Escenario 1:

Características de los servidores

Procesar	Intel(R) Core (TM) i3-8130U CPU@ 2.20GHz 2.21
RAM	1G
Disco Duro	10G
Sistema Operativo	Ubuntu 20.04

Tabla 3: Caracteristicas de los servidores. Autora: Mariuxy Vásquez

Para este escenario se procedió a enviar 30.000 solicitudes al balanceador de cargas, donde procedió a realizar un cálculo de los porcentajes de diferencia en la cual permite observar cómo varían las métricas de los dos algoritmos haciendo uso de la siguiente formula:

Porcentaje de Diferencia = ((Valor Nuevo - Valor Antiguo) / Valor Antiguo) * 100

En base a lo explicado se obtuvieron los siguientes porcentajes, para visualizar de forma más detallada las estadísticas de las mediciones del sistema, por favor vea el anexo 7:

Métrica	**ROUND ROBIN**	**LEAST CONNECTION**	**Diferencia (%)**
Transacciones totales	600,000 hits	600,000 hits	N/A
Disponibilidad	100.00%	100.00%	N/A
Tiempo transcurrido (segundos)	963.52	941.79	-2.25%
Datos transferidos (MB)	1852.80	1852.80	N/A
Tiempo de respuesta (segundos)	0.02	0.02	N/A
Tasa de transacción	622.72 trans/sec	637.08 trans/sec	+2.31%
Rendimiento (MB/seg)	1.92	1.97	+2.60%
concurrencia	9.96	9.96	N/A
Transacciones exitosas	600,000	600,000	N/A
Transacciones fallidas	0	0	N/A

Transacción más larga	0.10	0.10	N/A
Transacción más corta	0.00	0.00	N/A

Tabla 4: Resultados de los algoritmos, calculo en porcentajes. Autora Mariuxy Vásquez

Entre las diferencias de las meticas se denotan las siguientes observaciones:

- **Transacciones totales:** Ambos escenarios tienen un número igual de transacciones totales, es decir, 600,000 hits cada uno. No hay diferencia entre ellos.
- **Disponibilidad:** La disponibilidad es del 100.00% en ambos casos, lo que significa que ambos algoritmos tienen un alto grado de disponibilidad.
- **Tiempo transcurrido (segundos):** "Least Connection" tiene un tiempo transcurrido de 941.79 segundos (15.70 minutos), entonces Least Connection fue un 2.25% más rápido en responder todas las solicitudes que "Round Robin," que tiene un tiempo de 963.52 segundos (16.06 minutos).
- **Datos transferidos (MB):** Ambos escenarios transfirieron la misma cantidad de datos, 1852.80 MB cada uno.
- **Tiempo de respuesta (segundos):** El tiempo de respuesta promedio es el mismo en ambos casos, 0.02 segundos, por lo que la diferencia porcentual es "N/A."
- **Tasa de transacción:** "Least Connection" tiene una tasa de transacción de 637.08 transacciones por segundo (10.62 minutos), que es un 2.31% rápida que la tasa de transacción en "Round Robin".
- **Rendimiento (MB/seg):** El rendimiento en términos de transferencia de datos por segundo es mejor en "Least Connection," con 1.97 MB/seg, que es un 2.60% más eficiente que el rendimiento en "Round Robin," que es de 1.92 MB/seg.
- **Concurrencia:** La concurrencia es igual en ambos casos, 9.96, lo que significa que ambos escenarios manejan una cantidad similar de conexiones concurrentes.

- **Transacciones exitosas:** Ambos escenarios tienen la misma cantidad de transacciones exitosas, 600,000, por lo que la diferencia porcentual es "N/A.".
- **Transacciones fallidas:** Ninguno de los algoritmos tubo transacciones fallidas.
- **Transacción más larga y más corta:** Las transacciones más largas y más cortas son iguales en ambos escenarios, por lo que la diferencia porcentual es "N/A."

Con respecto a la distribución de cargas del balanceador tenemos como resultado que

phpmyadmin_backend

	Queue			Session rate			Sessions				
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot
phpmyadmin1	0	0	-	0	436		0	9	50	300 000	300 000
phpmyadmin2	0	0	-	0	435		0	10	50	300 000	300 000
Backend	0	0		0	871		0	10	26 213	600 000	600 000

Round Robin distribuye las cargas de manera igualitaria como se visualiza en la imagen

Gráfico 4: Distribución de cargas Round Robin.

mientras que Least Connection distribuye las cargas en ambos servidores dependiendo de la

phpmyadmin_backend

	Queue			Session rate			Sessions				
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot
phpmyadmin1	0	0	-	0	433		0	5	50	333 964	333 964
phpmyadmin2	0	0	-	0	430		0	5	50	334 036	334 036
Backend	0	0		0	863		0	10	26 213	668 000	668 000

cantidad de solicitudes o sesiones activas que tiene cada servidor.

Gráfico 5: Distribución de cargas Least Connection.

Ahora, con respecto al consumo de recursos de los servidores de base de datos en peticiones, con el uso de los dos algoritmos de balanceo de carga en su distribución se pudo observar lo siguiente:

Para visualizar de forma más detallado el rendimiento, por favor vea el anexo. En base a

la comparación de los dos algoritmos se muestra el siguiente resultado.

ASPECTO	ROUND ROBIN	LEAST CONNECTION	PROXY
%CPU (RANGO)	5.0% - 1.0%	4.3% - 1.0%	42.9%
%MEM	1.4%	1.5%	2.8%

Tabla 5: Rendimiento de los servidore con los 2 algoritmos. Autora: Mariuxy Vásquez

Round Robin tiene un rango de consumo de CPU ligeramente más alto (5.0% - 1.0%) en comparación con Least Connection (4.3% - 1.0%), pero ambos algoritmos tienen rangos estrechos y consistentes, lo que indica que los servidores están manejando la carga de manera equitativa y no están sobrecargados. Por lo tanto, referente al consumo de memoria, se podría decir que ambos algoritmos tienen un consumo moderado, haciendo que estas pequeñas diferencias tengan un impacto significativo en el rendimiento de los servidores. Mientras que el balanceador de cargar tiene un consumo de 42.9% de recursos de CPU y 2.8% de memoria, estando en un rango aceptable, ya que este es el encargado recibir todas las solicitudes y distribuirlas.

Escenario 2:

Procesar	Intel(R) Core (TM) i3-8130U CPU@ 2.20GHz 2.21 (2 procesadores)
RAM	2G
Disco Duro	20G
Sistema Operativo	Ubuntu 20.04

Tabla 6: Características de los servidores. Autora: Mariuxy Vásquez

Para este escenario se enviaron 100.000 solicitudes al balanceador de cargas donde los resultados fueron los siguientes.

Métrica	Round Robin	Least Connection	Porcentaje de Diferencia
Transacciones totales	2,000,000	2,000,000	0.00 %
Disponibilidad	100.00 %	100.00 %	0.00 %

Tiempo transcurrido	2,836.49 secs	2,337.12 secs	17.59 %
Datos transferidos	6,176.00 MB	6,176.00 MB	0.00 %
Tiempo de respuesta promedio	0.01 secs	0.01 secs	0.00 %
Tasa de transacciones	705.10 trans/sec	855.75 trans/sec	21.38 %
Rendimiento (Throughput)	2.18 MB/sec	2.64 MB/sec	21.10 %
Concurrencia	9.95	9.97	0.20 %
Transacciones exitosas	2,000,000	2,000,000	0.00 %
Transacciones fallidas	0	0	0.00 %
Transacción más larga	1.16 secs	0.76 secs	34.48 %
Transacción más corta	0.00 secs	0.00 secs	0.00 %

Tabla 7: Comparación de estadísticas de los algoritmos, esenario2. Autora: Mariuxy Vásquez

En donde se procede a describir las observaciones encontradas en las estadísticas.

- **Transacciones totales:** Ambos algoritmos tienen la misma cantidad de transacciones, por lo que el porcentaje de diferencia es del 0.00 %.
- **Disponibilidad:** La disponibilidad es del 100.00 % en ambos casos, lo que significa que no hubo diferencia en este aspecto.
- **Tiempo transcurrido:** El tiempo transcurrido en "Round Robin" es 2,836.49 segundos que es aproximadamente 47.27 minutos y en "Least Connection" es 2,337.12 segundos que es 38.95 minutos, por lo tanto, la diferencia es de 17.59 %, lo que indica que "Least Connection" fue más rápido en completar todas las transacciones.
- **Datos transferidos:** En este acaso ambos algoritmos transfirieron la misma cantidad de datos.
- **Tiempo de respuesta promedio:** El tiempo de respuesta promedio es el mismo en ambos casos 0.01 segundos que son 10 milisegundos.
- **Tasa de transacciones:** La tasa de transacciones en "Round Robin" es de 705.10

transacciones por segundo, que convertidos a minutos son (11.74 m), mientras que en "Least Connection" es de 855.75 transacciones por segundo que son 14.26 minutos, en donde el porcentaje de diferencia es del 21.38 %, lo que indica que en este caso Round Robin es más eficiente en el manejo de transacciones.

- **Rendimiento (Throughput):** El rendimiento o velocidad "Round Robin" es de 2.18 MB por segundo, mientras que en "Least Connection" es de 2.64 MB por segundo. El porcentaje de diferencia es del 21.10 %. Por lo tanto, en este aspecto, "Least Connection" se consideraría mejor en términos de velocidad, ya que los datos se transfieren más rápido.
- **Concurrencia:** La concurrencia es muy similar en ambos casos (9.95 en "Round Robin" y 9.97 en "Least Connection"), con una diferencia mínima del 0.20 %. Es decir, ambos algoritmos son capaces de manejar múltiples conexiones simultáneamente sin que una interfiera con las demás, pero podemos decir que Least Connections está manejando un número se conexiones mejor.
- **Transacciones exitosas:** Ambos algoritmos tienen la misma cantidad de transacciones exitosas.
- **Transacciones fallidas:** En ninguno de los algoritmos hubo transacciones fallidas.
- **Transacción más larga:** La transacción más larga en "Round Robin" fue de 1.16 segundos, mientras que en "Least Connection" fue de 0.76 segundos teniendo una diferencia del 34.48 %, lo que indica que las transacciones con un valor más bajo se consideran generalmente mejor, indicando que las transacciones individuales se procesaron de manera más rápida y eficiente en "Least Connection".
- **Transacción más corta:** Ambos algoritmos tienen el mismo porcentaje de transacciones

más cortas que es del 0%

Ahora, con respecto al consumo de recursos tenemos:

ASPECTO	ROUND ROBIN	LEAST CONNECTION	PROXY
%CPU (RANGO)	4.5% - 1.0%	4.3% - 1.0%	43,7
%MEM	1.0%	1.3%	2,8

Tabla 8: Consumo de recursos de los servidores. Autora: Mariuxy Vásquez

Como se visualiza en la tabla, se puede apreciar que hay una diferencia muy significativa entre ambos algoritmos lo que significa que a pesar que las solicitudes se eleven el consumo de recursos de los servidores por carga de solicitud es equitativo en ambos y el consumo del balanceador no es elevado, pero se puede denotar que Least Connection tiene un menor consumo en CPU, pero en memoria es un poco más elevado.

Escenario 3:

Procesar	Intel(R) Core (TM) i3-8130U CPU@ 2.20GHz 2.21 (2 procesadores)
RAM	3G
Disco Duro	30G
Sistema Operativo	Ubuntu 20.04

Tabla 9: Características de los servidores. Autora: Mariuxy Vásquez

En este escenario se enviaron 200.000 solicitudes hacia el balanceador de cargas y los resultados fueron los siguientes:

Métrica	Round Robin	Least Connection	Diferencia
Transacciones totales	4,000,000 hits	4,000,000 hits	0.00 %
Disponibilidad	100.00 %	100.00 %	0.00 %
Tiempo transcurrido	4,034.95 secs (62.25 m)	3,906.58 secs (65.11 m)	3.18 % menos en Least Conn.
Datos transferidos	12,351.99 MB	12,351.99 MB	0.00 %
Tiempo de respuesta promedio	0.01 secs	0.01 secs	0.00 %
Tasa de transacciones	991.34 trans/sec	1023.91 trans/sec	3.28 % más en Least Conn.
Rendimiento (Throughput)	3.06 MB/sec	3.16 MB/sec	3.27 % más en Least Conn.
Concurrencia	9.96	9.97	0.01 % más en Least Conn.
Transacciones exitosas	4,000,000	4,000,000	0.00 %
Transacciones fallidas	0	0	0.00 %
Transacción más larga	1.97 secs	1.01 secs	48.23 % menos en Least Conn.
Transacción más corta	0.00 secs	0.00 secs	0.00 %

Tabla 10: Comparación de estadísticas de los algoritmos, esenario3. Autora: Mariuxy Vásquez

Diferencias notables:

1. **Tiempo transcurrido:** "Least Connection" tiene un tiempo transcurrido un 3.18 % menor que "Round Robin," lo que sugiere que fue más rápido en completar las transacciones en este escenario.
2. **Tasa de transacciones:** "Least Connection" tiene una tasa de transacciones un 3.28 % más alta que "Round Robin", lo que significa que puede manejar un mayor número de transacciones por segundo, ya que una tasa de transacciones más alta generalmente se considera beneficiosa, ya que indica una mayor capacidad de procesamiento y rendimiento.

3. **Rendimiento (Throughput):** El rendimiento en "Least Connection" es un 3.27 % más alto que en "Round Robin," lo que significa que transfiere datos a una velocidad ligeramente más rápida.
4. **Concurrencia:** La concurrencia es similar en ambos casos, con una diferencia mínima del 0.01 %, lo que significa que ambos escenarios manejaron una cantidad muy similar de conexiones concurrentes.
5. **Transacción más larga:** La transacción más larga en "Least Connection" es un 48.23 % más corta que en "Round Robin," lo que indica ser mejor en términos de tiempo de respuesta en "Least Connection."

Mientras que el consumo de los servidores se obtuvieron los siguientes resultados:

ASPECTO	ROUND ROBIN	LEAST CONNECTION	PROXY
%CPU (RANGO)	4.5% - 1.0%	3.7% - 1.3%	53.0
%MEM	1.0%	0.7%	2,7

Tabla 11: Consumo de recursos. Autora: Mariuxy Vásquez

En términos de uso de CPU y memoria, "Least Connection" parece ser más eficiente en general, ya que usa menos recursos que "Round Robin", lo que lo hace más eficiente, mientras que en el balanceador cargas tenemos un consumo de recursos más elevado pero estable.

Escenario 4:

Para este escenario se tomará en cuenta una variación de características de los servidores

Servidor 1	
Procesar	Intel(R) Core (TM) i3-8130U CPU@ 2.20GHz 2.21 (2 procesadores)
RAM	3G
Disco Duro	30G
Sistema Operativo	Ubuntu 20.04

Tabla 12: Características servidor 1. Autora: Mariuxy Vásquez

Servidor 2	
Procesar	Intel(R) Core (TM) i3-8130U CPU@ 2.20GHz 2.21
RAM	2G
Disco Duro	20G
Sistema Operativo	Ubuntu 20.04

Tabla 13: Características servidor 1. Autora: Mariuxy Vásquez

Como se visualiza en las tablas 12 y 13, se configuraron los servidores con diferentes características, en donde se enviaron 30.000 solicitudes y los resultados de ambos algoritmos son los siguiente:

Métrica	Round Robin	Least Connection	Diferencia
Transactions	600,000 hits	600,000 hits	0.00%
Availability	100.00%	100.00%	0.00%
Tiempo transcurrido	809.64 segundos	660.69 segundos	-18.37%
Datos transferidos	1852.80 MB	1852.80 MB	0.00%
Tiempo de respuesta promedio	0.01 segundos	0.01 segundos	0.00%
Tasa de transacciones	741.07 trans/s	908.14 trans/s	22.55%
Rendimiento (Throughput)	2.29 MB/seg	2.80 MB/seg	22.18%
Concurrencia	9.96	9.95	-0.10%
Transacciones exitosas	600,000	600,000	0.00%
Transacciones fallidas	0	0	0.00%
Transacción más larga	0.11 segundos	0.10 segundos	-9.09%
Transacción más corta	0.00 segundos	0.00 segundos	0.00%

Tabla 14: Estadísticas de los algoritmos escenario 4. Autora: Mariuxy Vásquez

En base a estos resultados se puede destacar que:

1. **Tiempo transcurrido**: Para Round Robin, el tiempo transcurrido fue de 809.64 segundos, que en minutos serian (13.49 m), mientras que en Least Connection fue de 660.69 segundos, quen en minutos es (11.01). Esto indica que el algoritmo Least Connection completó las transacciones más rápido.
2. **Tasa de transacciones**: Least Connection tuvo una tasa de transacciones ligeramente

mayor, con 908.14 transacciones por segundo (15.14 m), en comparación con las 741.07 transacciones por segundo (12.35 m) en Round Robin. Indicando que significa que Least Connection es capaz de procesar un mayor número de transacciones por segundo.

3. **Rendimiento (Throughput)**: Least Connection también tuvo un rendimiento superior, con un rendimiento de 2.80 MB por segundo, en comparación con los 2.29 MB por segundo de Round Robin. Es decir que Least Connection es mejor debido a su capacidad para transferir datos a una velocidad más rápida.
4. **Concurrencia**: La concurrencia es similar en ambas pruebas, con 9.96 en Round Robin y 9.95 en Least Connection. La diferencia es mínima, alrededor del -0.10%.
5. **Transacción más larga y más corta**: La transacción más larga en Round Robin fue de 0.11 segundos, mientras que en Least Connection fue de 0.10 segundos. La diferencia es del -9.09%, lo que indica que Least Connection tuvo una transacción más corta en términos de tiempo.

En base a estos resultados, el algoritmo Least Connection fue el que funciono de una manera mejor en términos de tiempo transcurrido, tasa de transacciones y rendimiento en comparación con Round Robin.

Con respecto al rendimiento de los servidores se obtuvieron los siguientes datos:

SERVIDOR 1			
ASPECTO	ROUND ROBIN	LEAST CONNECTION	PROXY
%CPU (RANGO)	5.0% - 3.0%	4.0% - 1.3%	44,2
%MEM	0.7%	0.7%	2,7

Tabla 15: Rendimiento del servidor 1, escenario 4. Autora: Mariuxy Vásquez

SERVIDOR 2			
ASPECTO	ROUND ROBIN	LEAST CONNECTION	PROXY

%CPU (RANGO)	3.3% - 1.0%	4.0% - 1.3%	44,2
%MEM	0.5%	0.5%	2,7

Tabla 16: Rendimiento del servidor 2, escenario 4. Autora: Mariuxy Vásquez

Como se puede apreciar en las tablas 15 y 16 , se puede interpretar que, Round Robin envía cargas a los servidores sin tomar en cuenta sus características haciendo que se consuman más recursos de los servidores, mientras que Least Conecction toma en cuenta estos aspectos y distribuye las cargas dependiendo de las características de los servidores, haciendo que se consuman recursos de manera igualitaria, evitando de una manera más precisa la sobrecarga de los servidores con menos características.

phpmyadmin_backend

	Queue			Session rate			Sessions					
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last
phpmyadmin1	0	0	-	290	412		5	10	50	31 979	31 979	0s
phpmyadmin2	0	0	-	289	411		4	10	50	31 979	31 979	0s
Backend	0	0		580	824		8	10	26 213	63 958	63 958	0s

Por otro lado, como se puede apreciar en las siguientes imágenes:

Gráfico 6: distribución round robin

Gráfico 7: Distribución Least Connection

phpmyadmin_backend

	Queue			Session rate			Sessions			
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total
phpmyadmin1	0	0	-	431	497		5	5	50	190 914
phpmyadmin2	0	0	-	432	498		4	5	50	191 181
Backend	0	0		864	994		7	10	26 213	382 095

Con base a esto se corrobora la distribución y el consumo de recursos de lo antes mencionado.

DISCUSION DE RESULTADOS

Encuestas dirigidas a los usuarios: Según las encuestas realizadas a los alumnos de la Universidad Técnica de Babahoyo ,tomando de la población a 36 estudiantes de la carrera de Sistemas de Información entre las secciones verpertina y matutina, en donde se pudo observar

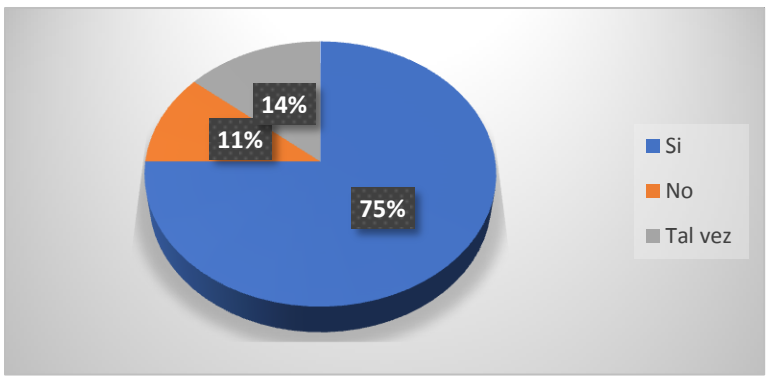
que hay un gran porcentaje de resultados negativos con respecto a solicitudes que haces los estudiantes a diversas páginas, expresando molestias como; los retrasos o lentitud en respuestas, bloqueos e interrupción , teniendo un impacto negativo en la confianza de los sitios web dinámicos como en la calidad de los servicios, haciendo que estos abandono el sitio, en la cual, los usuarios manifiestan que se deben tomar medidas para mejorar los servicios que prestan tanto en empresas públicas como privadas.

Encuestas dirigidas a expertos: Según la encuesta realizadas, tomando como muestra a 4 profesionales dedicados al uso y administración de servidores y docente, expresan que los algoritmos de balanceo de cargas son una gran ayuda para la distribución de solicitudes y evitar la sobrecarga de los servidores de base de datos, haciendo que los recursos disminuyan y que las solicitudes se distribuyan equitativamente, donde el 60% opta por implementar Round Robin y el 40% por Least Connection.

Criterio propio: Según los resultados de la practica ambos algoritmos cumplen su función de manera eficiente, pero se puede decir que, Least Connection tiene un porcentaje de rapidez en respuesta y menos consumo en recursos de los servidores, es decir, se desarrollaron 4 pruebas en donde Round Robin demora un poco más en solventar todas las solicitudes y este no toma en cuenta las características de los servidores, mientras que least connection se desenvolvió de una mejor manera y solvento las solicitudes de una manera más equitativa al someterse a grandes cantidades de solicitudes y conexiones activas.

Encuesta dirigida a Usuarios

1. ¿Has experimentado alguna vez retrasos en la obtención de respuestas al interactuar con



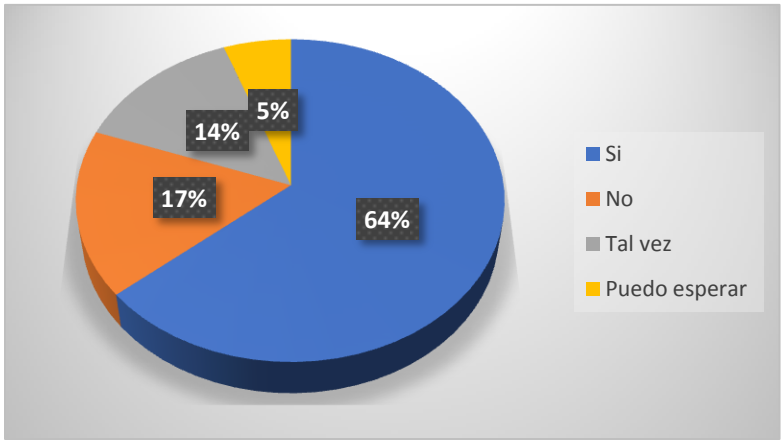
sistemas en línea?

OPCIONES	FRECUENCIA	PORCENTAJE
Si	27	75%
No	4	11%
Tal vez	5	14%
TOTAL	36	100%

Análisis: En base a esta encuesta el 75% de los encuestados ha experimentado retrasos en la obtención de respuestas al interactuar con sistemas en línea, mientras que el 11% afirmo no haberlo tenido y el 14% respondió que tal vez. El porcentaje más alto señala un problema de latencia en las interacciones en línea, lo que puede afectar negativamente la experiencia de usuario y la eficiencia de los sistemas.

2. ¿Consideras que un tiempo de respuesta lento en un sitio web o aplicación afecta tu experiencia como usuario?

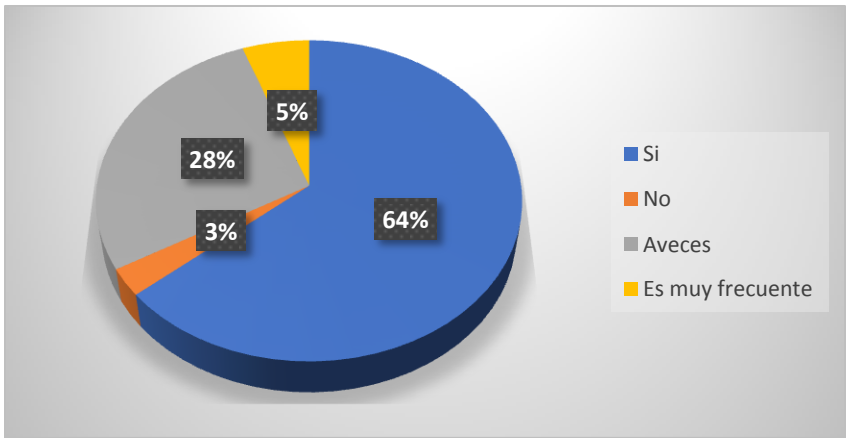
OPCIONES	FRECUENCIA	PORCENTAJE
Si	23	64%
No	6	17%
Tal vez	5	14%
Puedo esperar	2	6%
TOTAL	36	100%



Análisis: Según los resultados de la encuesta, el 64% de los participantes considera que un tiempo de respuesta lento en un sitio web o aplicación afecta negativamente su experiencia como usuario. El 17% indicó que no siente impacto, mientras que el 14% respondió "Tal vez". Solo el 6% señaló que puede esperar ante la presencia de un tiempo de respuesta lento. En consideración de estos hallazgos se interpresa que a la mayoría de los usuarios les afecta la velocidad de respuesta en las plataformas en línea.

3. ¿Has notado que ciertos sitios web o aplicaciones se vuelven lentos cuando hay un gran número de usuarios utilizando simultáneamente?

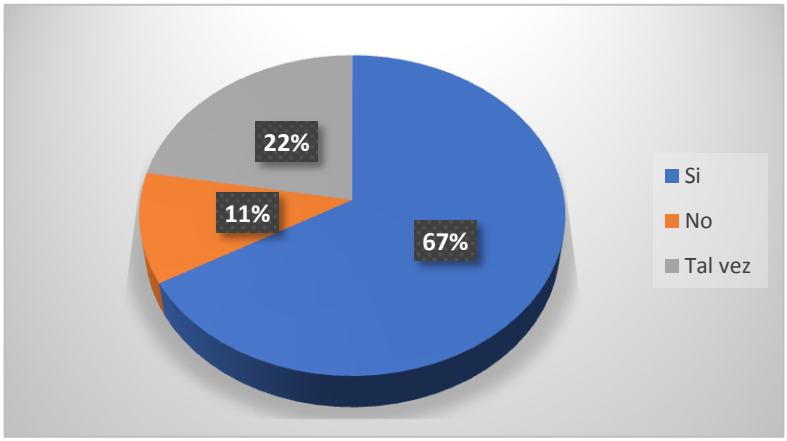
OPCIONES	FRECUENCIA	PORCENTAJE
Si	23	64%
No	1	3%
A veces	10	28%
Es muy frecuente	2	6%
TOTAL	36	100%



Análisis: De acuerdo con los resultados de la encuesta, el 64% de los encuestados ha notado que ciertos sitios web o aplicaciones se vuelven lentos cuando hay un gran número de usuarios simultáneamente. Un pequeño porcentaje, el 3%, indicó que no ha notado este fenómeno, mientras que el 28% respondió que esto sucede "a veces" y el 6% considera que es "muy frecuente". Esto sugiere que la congestión de usuarios puede ser un problema recurrente en algunas plataformas en línea.

4. ¿Crees que las organizaciones que ofrecen diferentes servicios a través de la web, deberían tomar medidas para mejorar el tiempo de respuesta en los servicios en línea?

OPCIONES	FRECUENCIA	PORCENTAJE
Si	24	67%
No	4	11%
Tal vez	8	22%
TOTAL	36	100%

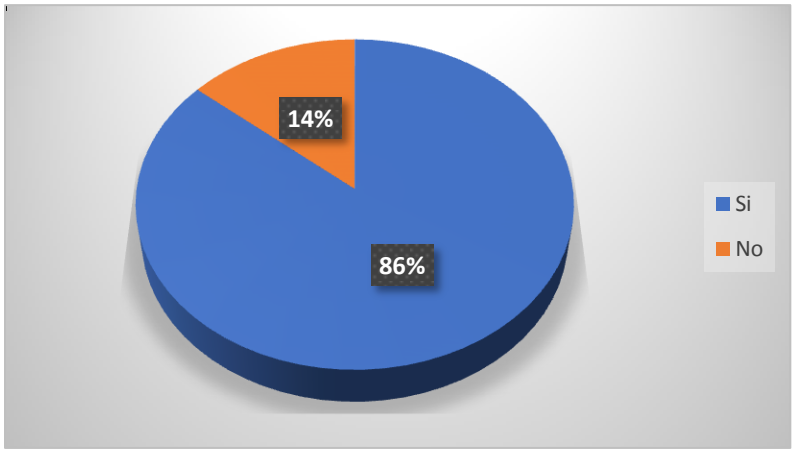


Análisis: Según los resultados de la encuesta, el 67% de los encuestados cree que las organizaciones que ofrecen servicios en línea deberían tomar medidas para mejorar el tiempo de respuesta. El 11% opinó que no es necesario hacerlo, mientras que el 22% respondió "Tal vez", indicando cierta ambigüedad al respecto. Del presente análisis de datos se destaca que la mayoría de los encuestados apoyan la necesidad de mejorar la velocidad de respuesta en los servicios en

línea.

5. ¿Estaría dispuesto a abandonar un sitio web si enfrenta problemas frecuentes?

OPCIONES	FRECUENCIA	PORCENTAJE
Si	31	86%
No	5	14%
TOTAL	36	100%



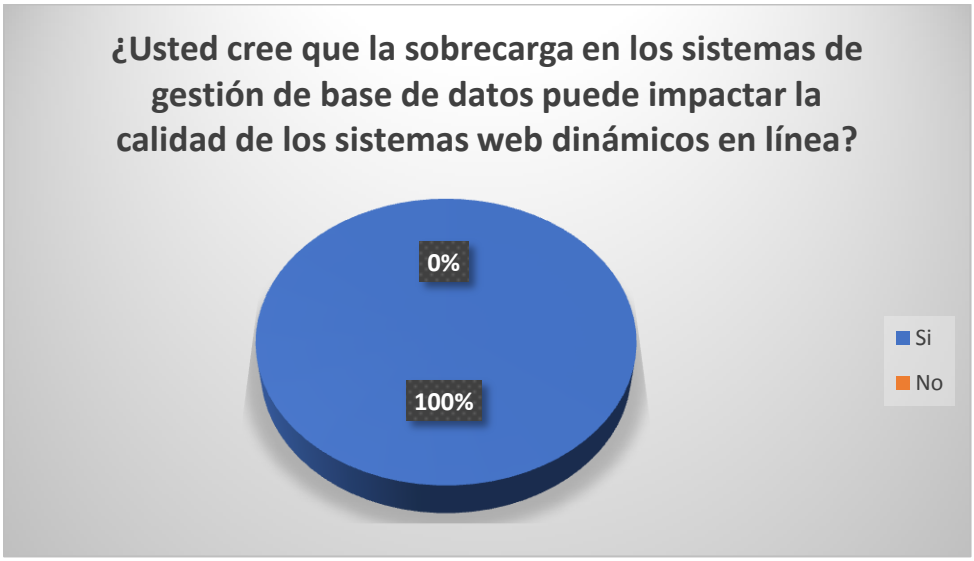
Análisis: Los resultados de la encuesta indican que el 86% de los participantes estarían dispuesto a abandonar un sitio web si enfrenta problemas frecuentes, mientras que el 14% no considera que tomaría esa medida. Esto resalta la importancia de la satisfacción del usuario y la

influencia que los problemas frecuentes pueden tener en la retención de usuarios en un sitio web.

Encuesta dirigida a profesionales

1. ¿Usted cree que la sobrecarga en los sistemas de gestión de base de datos puede impactar la calidad de los sistemas web dinámicos en línea?

OPCIONES	FRECUENCIA	PORCENTAJE
Si	4	100%
No	0	0%
TOTAL	4	100%

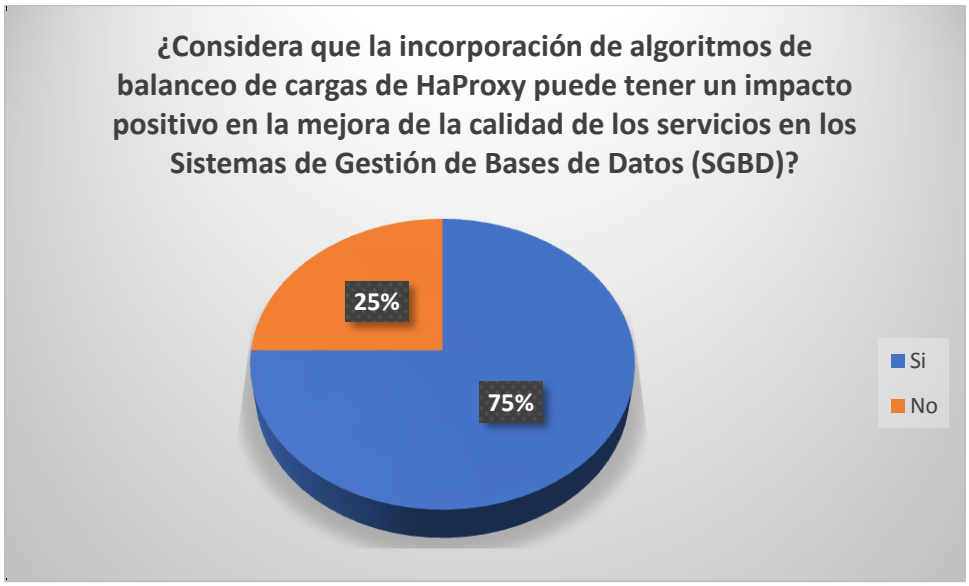


Análisis: El resultado del 100% en la encuesta, indica que todos los encuestados creen

que la sobrecarga en los sistemas de gestión de bases de datos impacta la calidad de los sistemas web dinámicos en línea. Este alto nivel de consenso sugiere una fuerte percepción de la relación entre la sobrecarga de la base de datos y la calidad de los sistemas.

2. ¿Considera que la incorporación de algoritmos de balanceo de cargas de Ha-Proxy puede tener un impacto positivo en la mejora de la calidad de los servicios en los Sistemas de Gestión de Bases de Datos (SGBD)?

OPCIONES	FRECUENCIA	PORCENTAJE
Si	3	75%
No	1	25%
TOTAL	4	100%

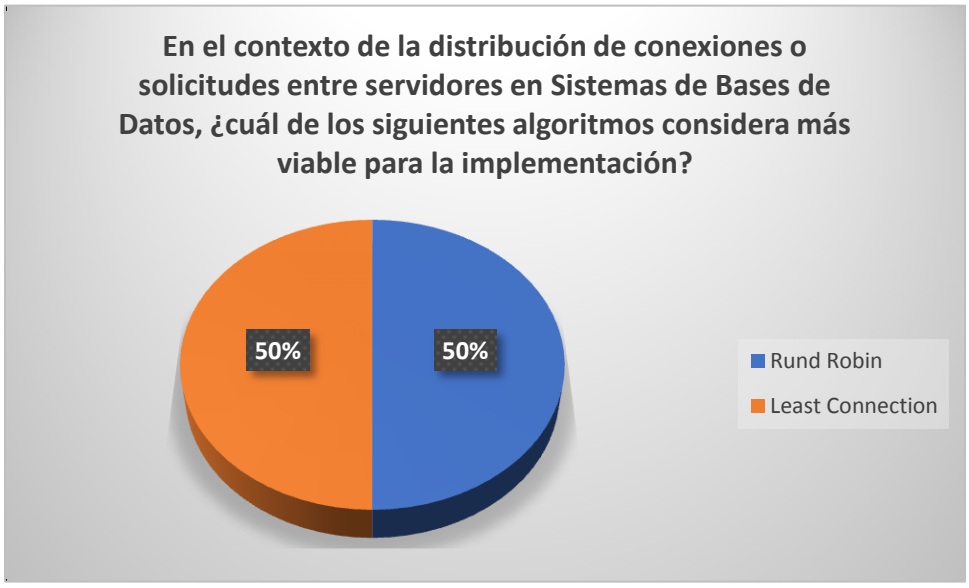


Análisis: El 75% de los encuestados considera que la incorporación de algoritmos de

balanceo de cargas de HaProxy tendrá un impacto positivo en la mejora de la calidad de los servicios en los Sistemas de Gestión de Bases de Datos (SGBD), sugiriendo un fuerte apoyo a esta estrategia. Sin embargo, es importante tener en cuenta que el 25% restante no comparte esta opinión, lo que indica que hay cierta diversidad de perspectivas en este tema. En general, la mayoría ve el balanceo de carga como una medida beneficiosa para mejorar la calidad de los servicios en los SGBD.

3. En el contexto de la distribución de conexiones o solicitudes entre servidores en Sistemas de Bases de Datos, ¿cuál de los siguientes algoritmos considera más viable para la implementación?

OPCIONES	FRECUENCIA	PORCENTAJE
Rund Robin	2	50%
Least Connection	2	50%
TOTAL	4	100%

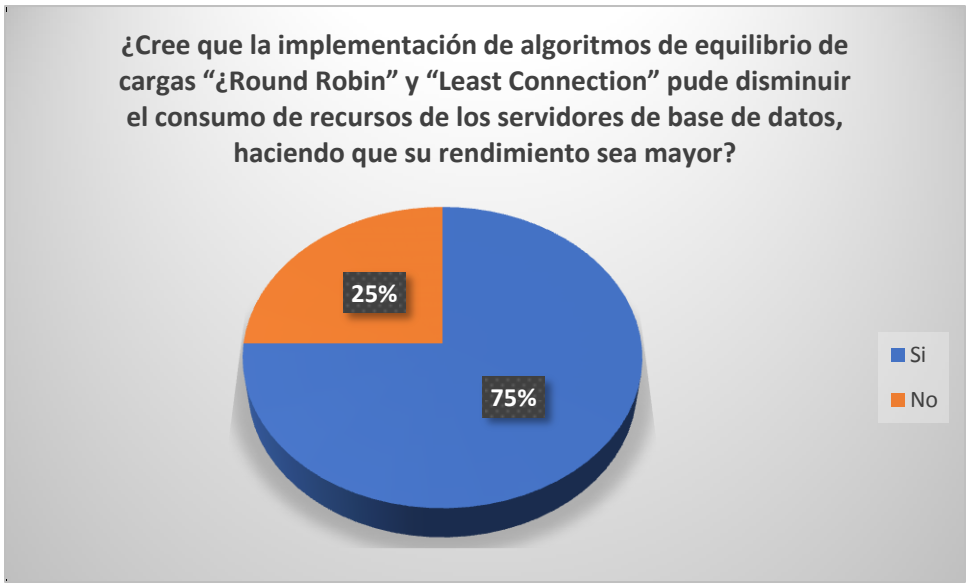


Análisis: Los resultados de la encuesta indican que el 50% considera el algoritmo Round

Robin como una herramienta más viable para la implementación en el contexto de distribución de conexiones o solicitudes entre servidores en sistemas de base de datos, mientras que el otro 50% prefiere el algoritmo Least Connection. Esto muestra una división equitativa de opiniones entre los encuestados, lo que indica que ambos algoritmos son considerados viables.

4. ¿Cree que la implementación de algoritmos de equilibrio de cargas "¿Round Robin" y "Least Connection" pude disminuir el consumo de recursos de los servidores de base de datos, haciendo que su rendimiento sea mayor?

OPCIONES	FRECUENCIA	PORCENTAJE
Si	3	75%
No	1	25%
TOTAL	4	100%



Análisis: Los resultados de la encuesta indican que el 75% de los encuestados cree que la implementación de algoritmos de equilibrio de carga "Round Robin" y "Least Connection" puede disminuir el consumo de recursos de los servidores de base de datos, mejorando su

rendimiento. Solo el 25% opina lo contrario. Podemos concluir que los algoritmos en estudio pueden tener un impacto positivo en la eficiencia de los servidores de bases de datos.

CONCLUSIONES

Las respuestas de las encuestas proporcionan una visión valiosa en la percepción, experiencia de los usuarios y la retención, porque la eficiencia en los servicios en línea y los tiempos de respuestas son factores importantes para la satisfacción de los usuarios, surgiendo la necesidad de un enfoque para mejorar el rendimiento de los servidores.

Por otro lado, las encuestas realizadas a profesionales en el área, reflejan un alto grado de acuerdo sobre la influencia de la sobrecarga en los sistemas de gestión de base de datos y la eficiencia de los algoritmos de equilibro de cargas, porque mejoran significativamente el rendimiento y el tiempo de respuesta de la base de datos, considerando también la importancia de los diferentes enfoques para la implementación de los algoritmos que nos ayudan a distribuir las cargas de manera equitativa entre servidores.

Según los resultados encontrados en los 4 escenarios realizados en esta investigación, se puede concluir que en general el algoritmo “Least Connection” supera a “Round Robin” en varios aspectos claves. Destacando con tiempo transcurrido menor, una mayor tasa de transacciones, mejor rendimiento y distribuyendo las cargas de una mejor manera, haciendo que los recursos de los servidores de base de datos sean igualitarios o estén a un nivel acorde a sus características en comparación con Round Robin.

RECOMENDACIONES

Como recomendación en base a la práctica y comparación de los algoritmos Round Robin y Least Connection tenemos:

1. Antes de elegir un algoritmo de balanceo de cargas, es necesario definir claramente los objetivos de rendimiento, las necesitades de la aplicación y el sistema de base de datos, considerando aspectos como la cantidad esperada de transacciones, la eficiencia temporal y la capacidad de adaptabilidad a cambios en las variaciones de cargas.
2. Es importante tener una atención a la sobrecarga de los sistemas de gestión de base de datos, porque las encuestas realizadas a profesionales destaco la importancia de reconocer y abordar este aspecto en los sistemas, siendo fundamental monitorear, optimizar y con ello supervisar el rendimiento de los servidores y la distribución de las cargas, permitiendo identificar problemas que pueden surgir y ajustar las configuraciones según sea necesario para así garantizar la calidad de los servicios web y la satisfacción de los usuarios.
3. Evalúa como los algoritmos manejaran el crecimiento futuro de la carga, es decir, aunque Least Connection muestra mejores ventajas en escenarios de base de datos es necesarios recordar que la elección del algoritmo debe adaptarse a necesidades específicas de un entorno, porque algunos algoritmos pueden ser más escalables que otros a medida que las

necesidades aumentan.

4. La eficiencia en el uso de recursos es importante, es recomendable observar cómo el algoritmo administra los recursos de CPU y memoria. Se recomienda posteriores investigaciones sobre la configuración y el ajuste de los algoritmos en entornos más específicos.

REFERENCIAS

Bastidas Delgado, J. W. (2021). *Análisis, diseño e implementación de balanceador de carga segmentando la red del sistema de balanzas de pequeñas y medianas empresas del sector industrial ciudad de Guayaquil*. Retrieved 6 de 8 de 2023, from http://repositorio.ug.edu.ec/bitstream/redug/52246/1/B-CINT-PTG-N.603%20Bastidas%20Delgado%20Jorge%20Washington.pdf

Carlos Navarro, J. A. (2018). *Arquitectura de balanceo de cargas dinamico para la logica de conmutacion de EtherChannel*. Retrieved 6 de 8 de 2023, from https://repository.javeriana.edu.co/bitstream/handle/10554/15400/CarlosNavarroJaimeAlejandro2013.pdf?sequence=1

Chunga Zuloeta, J. L., y Chuzon Sanchez , W. (2019). *COMPARACIÓN DE ALGORITMOS DE BALANCEADORES DE CARGA UTILIZANDO CLÚSTER HOMOGÉNEO EN SERVIDORES WEB.* Retrieved 6 de 8 de 2023, from file:///C:/Users/ma/Downloads/Chunga%20Zuloeta%20-%20S%C3%A1nchez%20Chuz%C3%B3n.pdf

Codig, K. (22 de 07 de 2022). *Load balancing o balanceo de carga*. Retrieved 6 de 8 de 2023,

from https://keepcoding.io/blog/que-es-load-balancing-o-balanceo-de-carga/

Llano Miraval, J. D., y Castellanos Landazabal, S. A. (30 de 11 de 2021). *Nodo de balanceo de carga enfocado a redes de proveedores de internet.* Retrieved 6 de 8 de 2023, from https://repository.javeriana.edu.co/bitstream/handle/10554/62087/303-attachment-1640129012.pdf?sequence=1

Marchionni, E. A. (2019). *Administrador de servidores.* Retrieved 6 de 8 de 2023, from https://books.google.es/books?hl=es&lr=&id=CfhGJ7yylRgC&oi=fnd&pg=PA34&dq=que+son+servidores&ots=5J8eZT_Px3&sig=Pp1X6QzHI_tkoPB3cCwi7ol9-Aw#v=onepage&q=que%20son%20servidores&f=false

Mejia Viteri, J. T., Gonzales Valero, M. I., y España Leon, A. R. (23 de 2 de 2018). *Programacion de algoritmos de balanceo de carga con HA-Proxy de servicios HTTP*. Retrieved 06 de 08 de 2023, from https://revistas.utb.edu.ec/index.php/sr/article/view/416/304

Perez Ibarra, S. G., Quispe, R. J., Mullicundo, F. F., y Lamas, D. A. (4 de 2021). *HERRAMIENTAS Y TECNOLOGÍAS PARA EL DESARROLLO WEB DESDE EL FRONTEND AL BACKEND.* Retrieved 6 de 8 de 2023, from SEDICI: http://sedici.unlp.edu.ar/bitstream/handle/10915/120476/Ponencia.pdf-PDFA.pdf?sequence=1&isAllowed=y

Perreto, C., Gonzales, A., Frascaroli, M., y Toaza, L. (10 de 2021). *EL METODO TOPSIS INTEGRADO A UN DESARROLLO BACK-END PARA LA SELECCION DE UN DISPOSITIVO MOVIL.* Retrieved 6 de 8 de 2023, from REVISTA: https://revistas.unc.edu.ar/index.php/epio/article/view/35540/35671

Sanchez Calderon, J. D., y Guzman Pineda, J. J. (2021). *BALANCEADOR DE CARGA*

UTILIZANDO ALGORITMOS DE CALIDAD DE SERVICIOS(QOS) EN REDES DEFINIDAS. Retrieved 5 de 8 de 2023, from https://repositorio.unicordoba.edu.co/bitstream/handle/ucordoba/4220/SanchezCalderonJavierDavid-GuzmanPinedaJuanJose.pdf?sequence=1&isAllowed=y

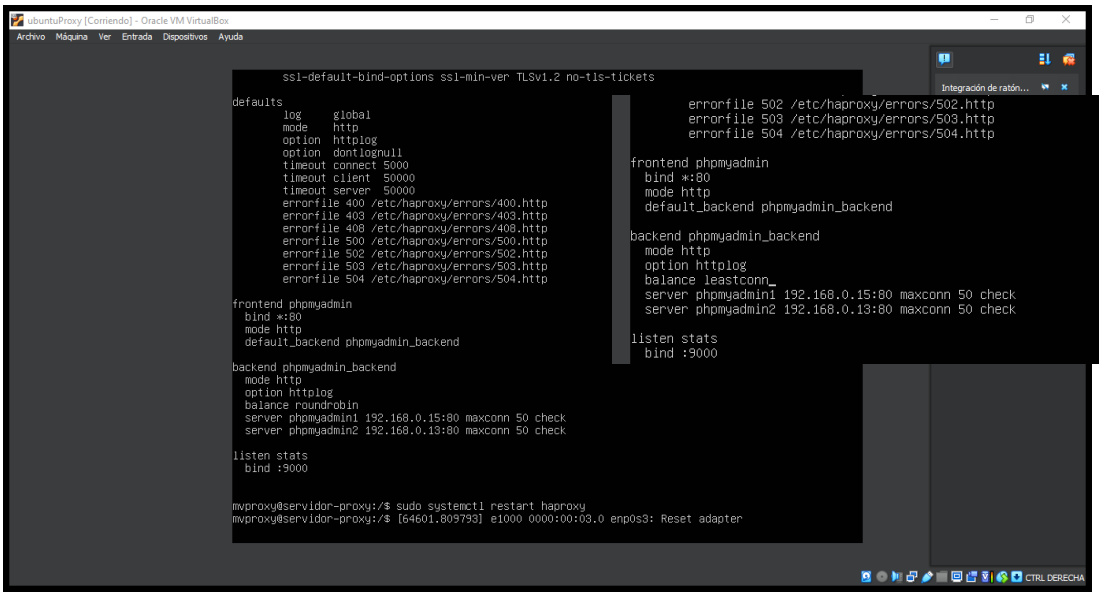
Teixeira, A. (2019). *Balanceo de cargas.* Retrieved 5 de 08 de 2023, from https://mum.mikrotik.com/presentations/CL16/presentation_3125_1456819785.pdf

Tesone, F., Thomas, P., Marrero , L., Olsowy, V., y Pesado, P. (8 de 10 de 2021). *SEDICI.* Retrieved 1 de 08 de 2023, from Un Análisis Experimental de Sistemas de gestion de base de datos: http://sedici.unlp.edu.ar/bitstream/handle/10915/130353/Documento_completo.pdf?sequence=1&isAllowed=y

Universidad Europea. (27 de 12 de 2021). *Sistemas de gestion de base de datos*. Retrieved 05 de 08 de 2023, from https://universidadeuropea.com/blog/para-que-sirve-gestor-base-datos/

ANEXOS

M



áqui

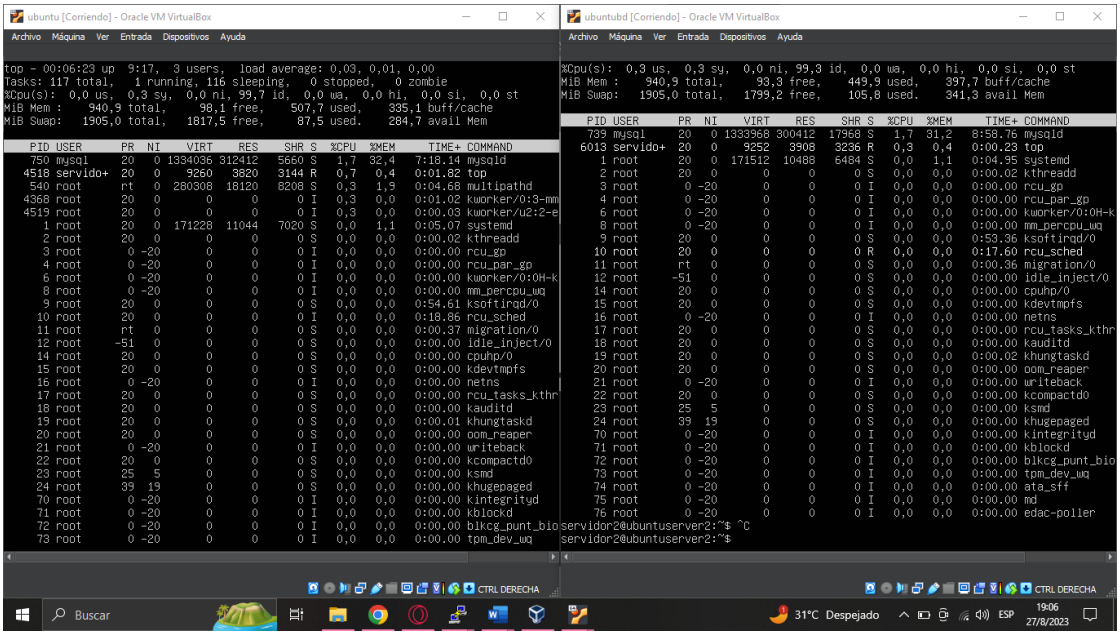
nas

virtu

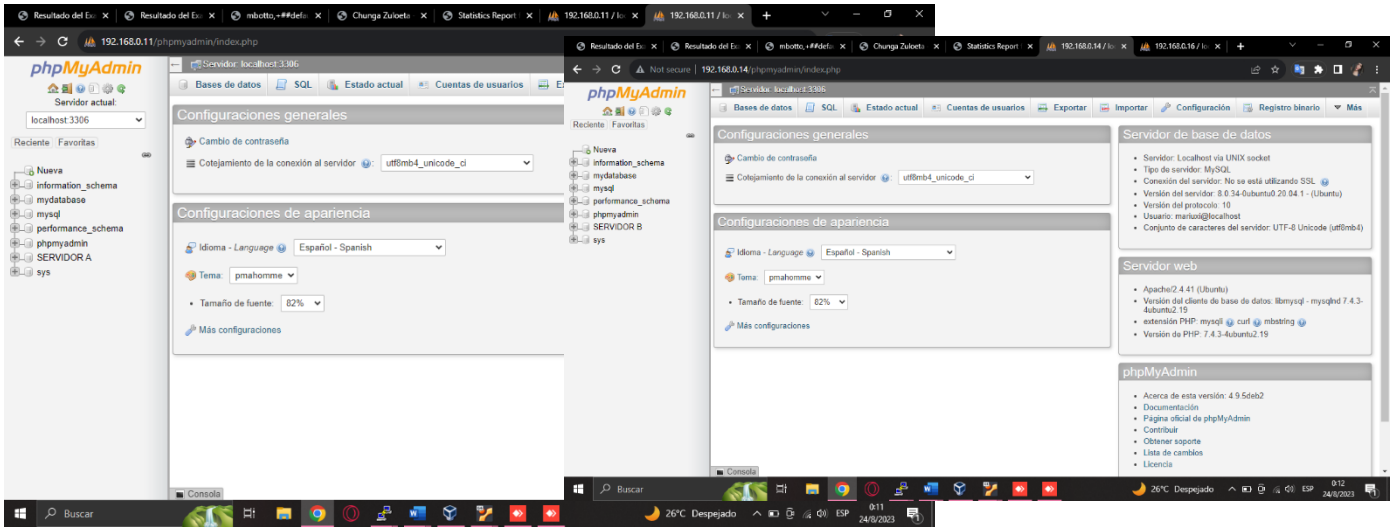
ales

A

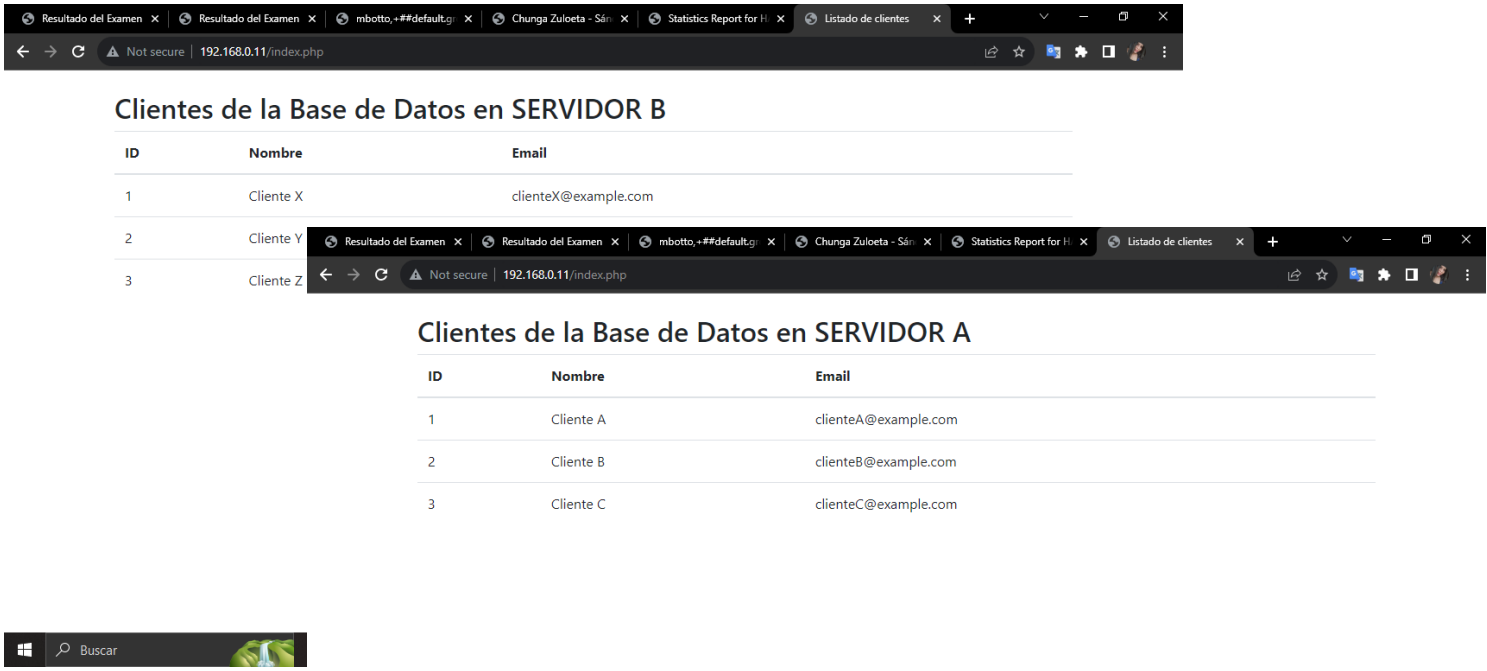
nexo 1: Servidor de ha-proxy, configuración de algoritmos, Round Robin y Least Connection



Anexo 2: Servidores de base de datos, servidor 1 y servidor 2

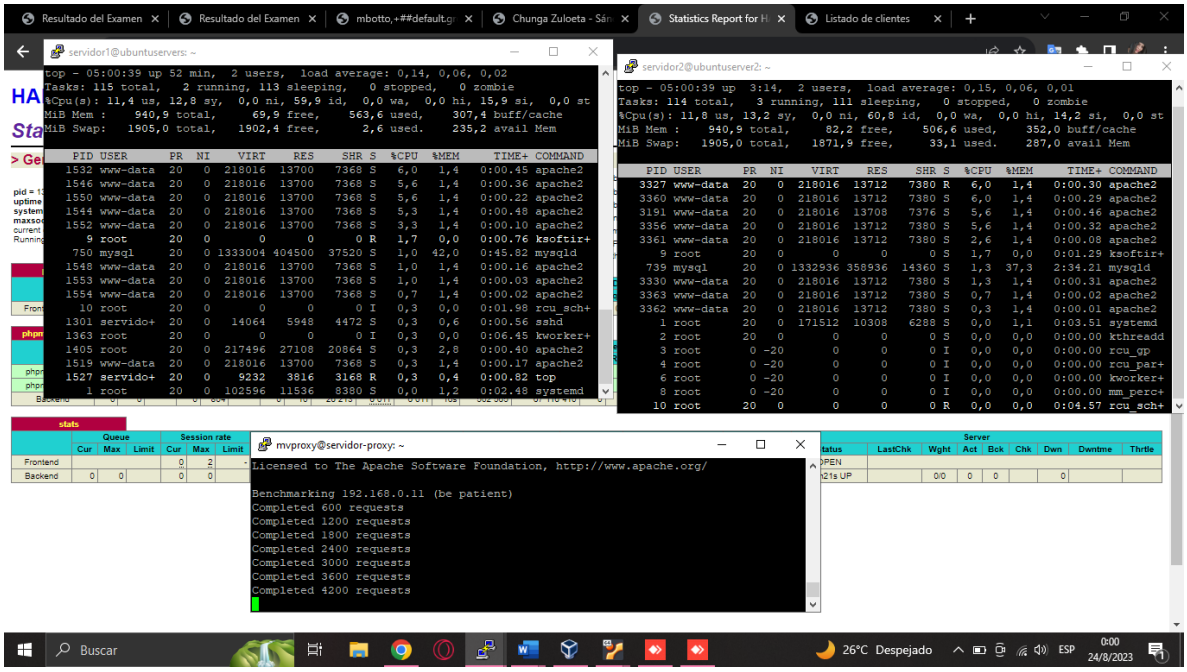


Anexo 3: Interfaz de phpmyadmin, administrador de base de datos

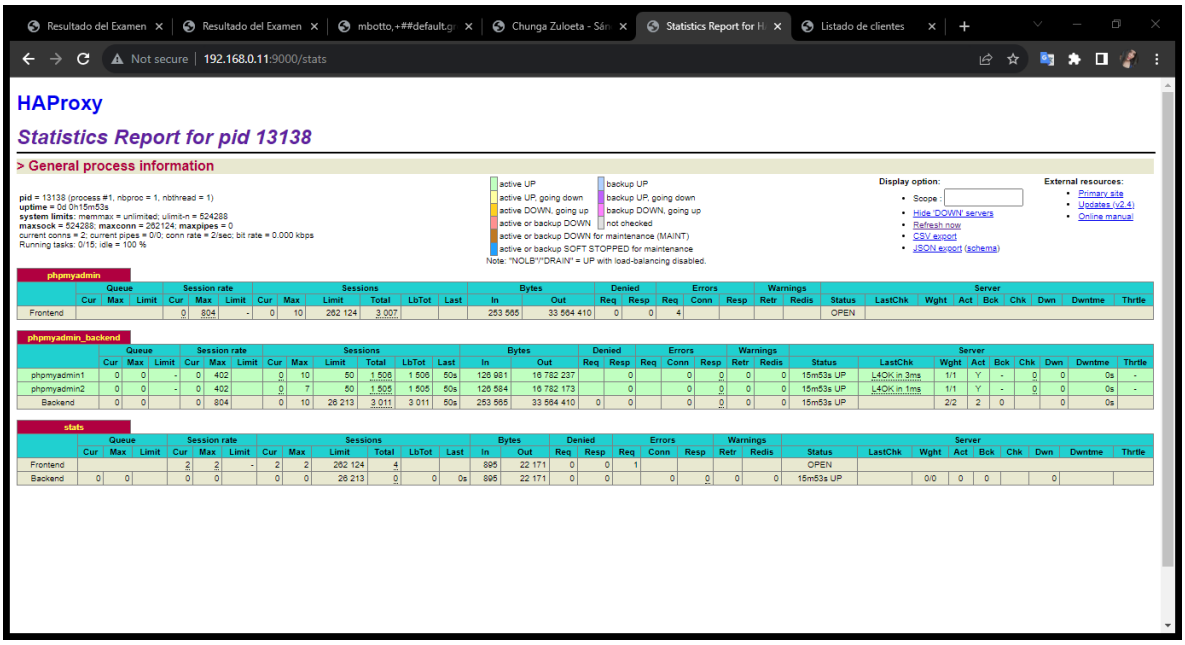


Anexo 4:

Distribución de servidores



Anexo 5: Maquina cliente y envió de cargas a servidores de base de datos



Anexo 6: Estadísticas proporcionadas por él envió de solicitudes al terminar y por el reporte de estado de Ha-proxy.

Escenario 1:

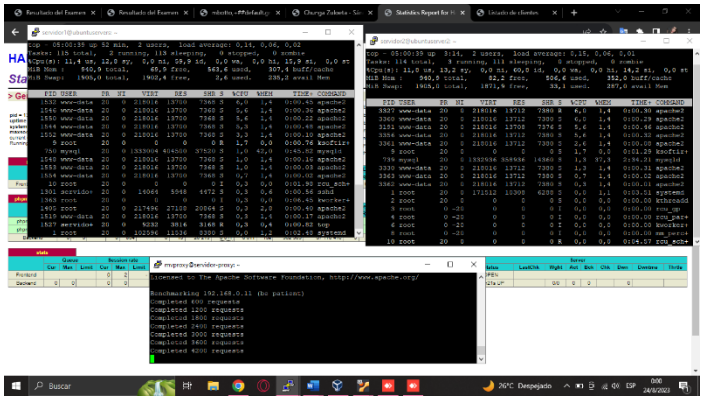
ROUND ROBIN	LEAST CONNECTION

backend phpmyadmin_backend
  mode http
  option httplog
  balance roundrobin_
  server phpmyadmin1 192.168.0.16:
  server phpmyadmin2 192.168.0.74:
 	
listen stats
  bind :9000
  mode http
  stats enable
  stats uri /stats
  stats hide-version
  stats auth haproxy:haproxy
	frontend phpmyadmin
	  bind *:80
	  mode http
	  default_backend phpmyadmin_ba
	
	backend phpmyadmin_backend
	  mode http
	  option httplog
	  balance leastconn
	  server phpmyadmin1 192.168.0.
	  server phpmyadmin2 192.168.0.

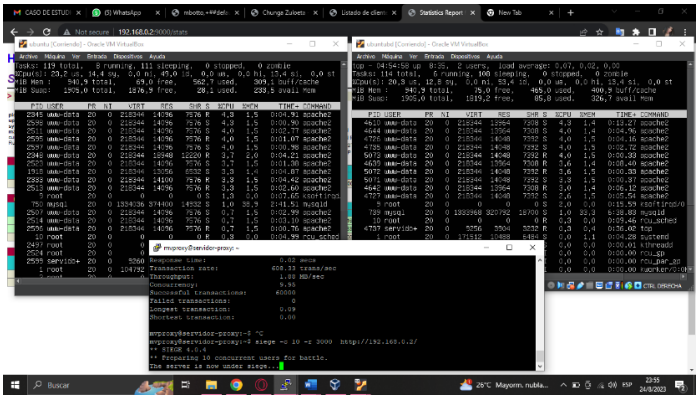
```
mvproxy@servidor-proxy: ~
mvproxy@servidor-proxy:~$ siege -c 10 -r 1000  http://1
** SIEGE 4.0.4
** Preparing 10 concurrent users for battle.
The server is now under siege...
Transactions:                  20000 hits
Availability:                 100.00 %
Elapsed time:                  33.19 secs
Data transferred:              61.76 MB
Response time:                  0.02 secs
Transaction rate:             602.59 trans/sec
Throughput:                     1.86 MB/sec
Concurrency:                    9.93
Successful transactions:       20000
Failed transactions:               0
Longest transaction:            0.09
Shortest transaction:           0.00

mvproxy@servidor-proxy:~$ ^C
```

RESULTADOS DEL BALANCEO	RESULTADOS DEL BALANCEO
mvproxy@servidor-proxy:~$ siege -c 10 ** SIEGE 4.0.4 ** Preparing 10 concurrent users for b The server is now under siege... Transactions: 600000 h Availability: 100.00 % Elapsed time: 963.52 s Data transferred: 1852.80 M Response time: 0.02 s Transaction rate: 622.72 t Throughput: 1.92 M Concurrency: 9.96 Successful transactions: 600000 Failed transactions: 0 Longest transaction: 0.10 Shortest transaction: 0.00 mvproxy@servidor-proxy:~$	mvproxy@servidor-proxy:~$ siege -c 10 - ** SIEGE 4.0.4 ** Preparing 10 concurrent users for bat The server is now under siege... Transactions: 600000 hit Availability: 100.00 % Elapsed time: 941.79 sec Data transferred: 1852.80 MB Response time: 0.02 sec Transaction rate: 637.08 tra Throughput: 1.97 MB/ Concurrency: 9.96 Successful transactions: 600000 Failed transactions: 0 Longest transaction: 0.10 Shortest transaction: 0.00 mvproxy@servidor-proxy:~$

Anexo 7: Estadísticas del escenario 1



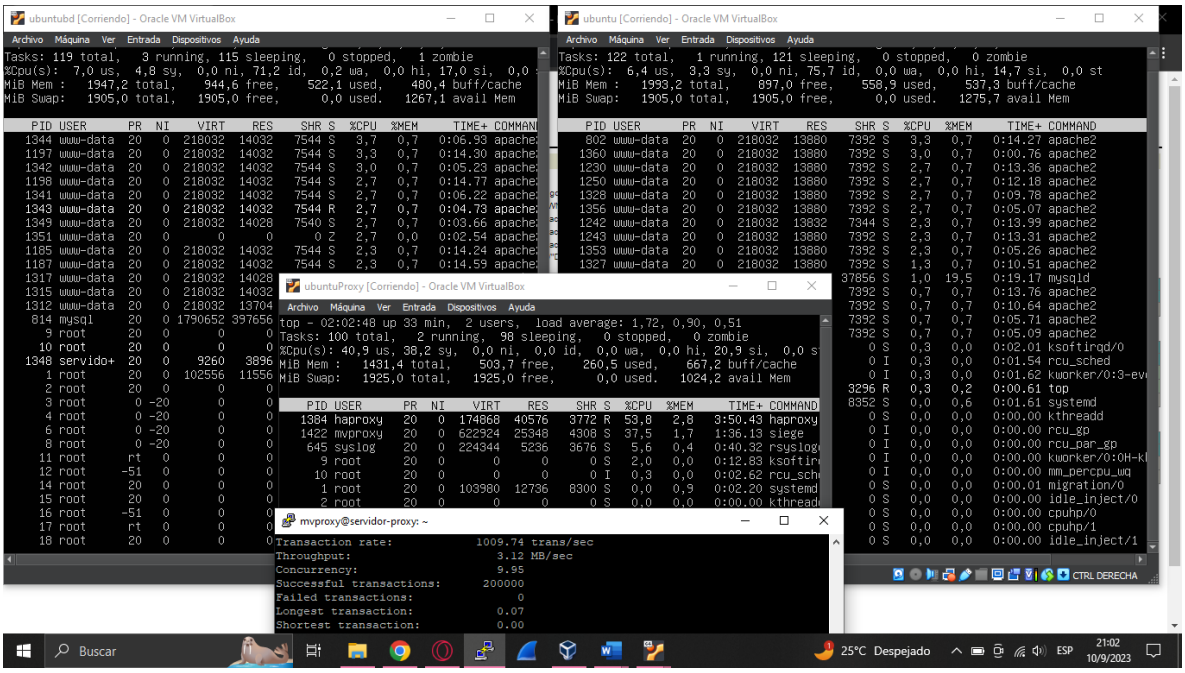
ALGORITMO ROUND ROBIN

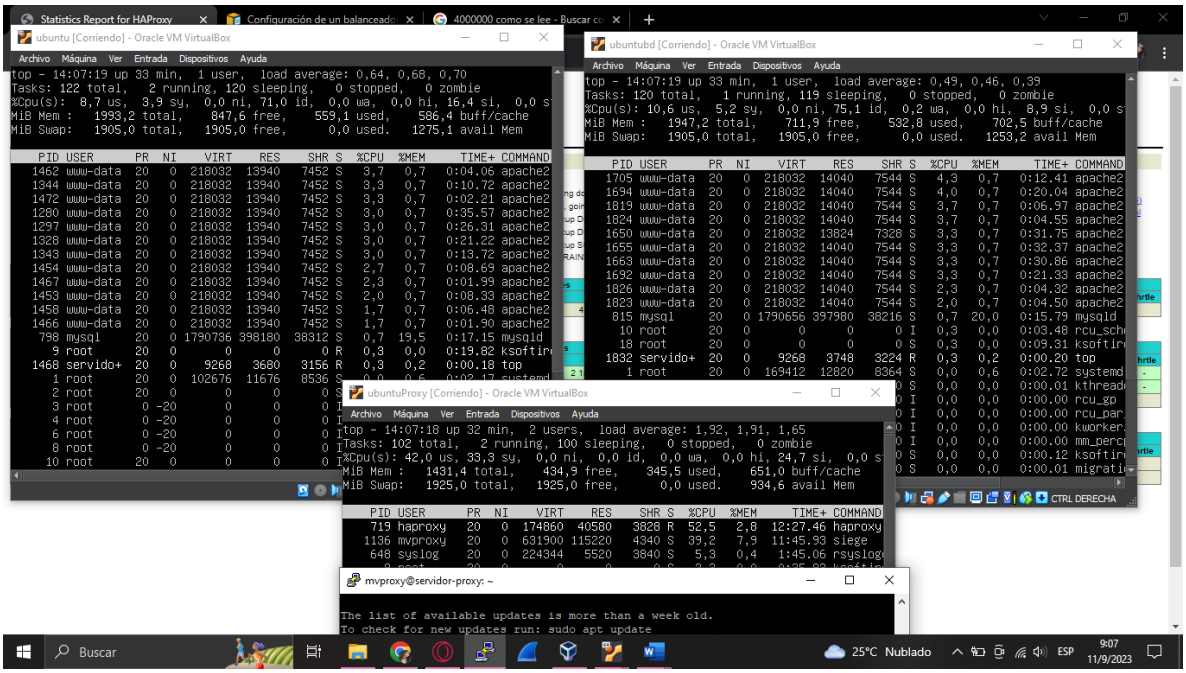


ALGORTIMO LEAST CONNECTION

Anexo 8: Consumo de recursos, Round Robin y Least Connection.

Escenario 2:

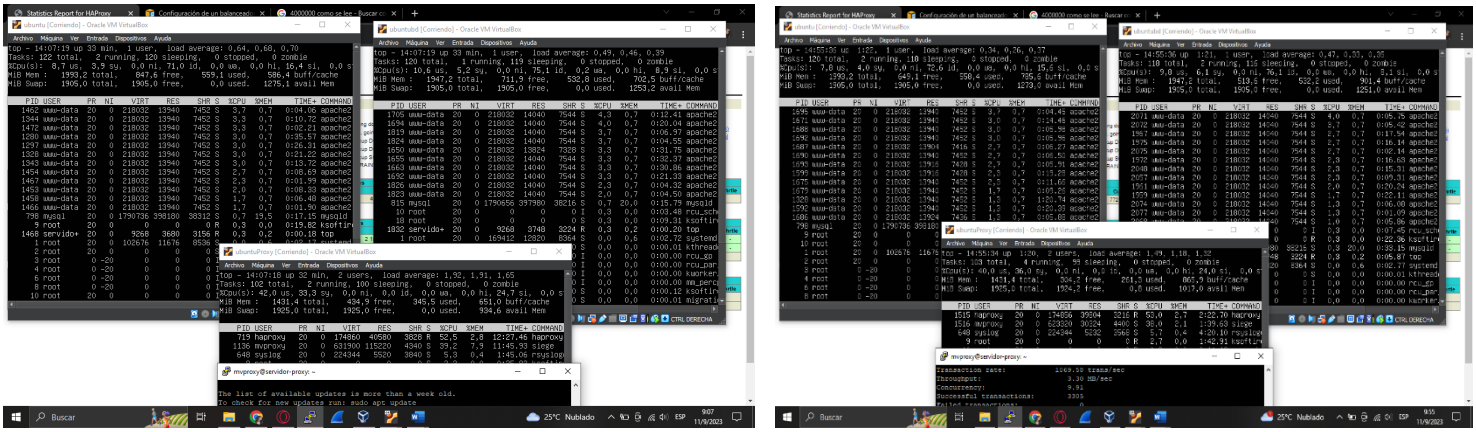




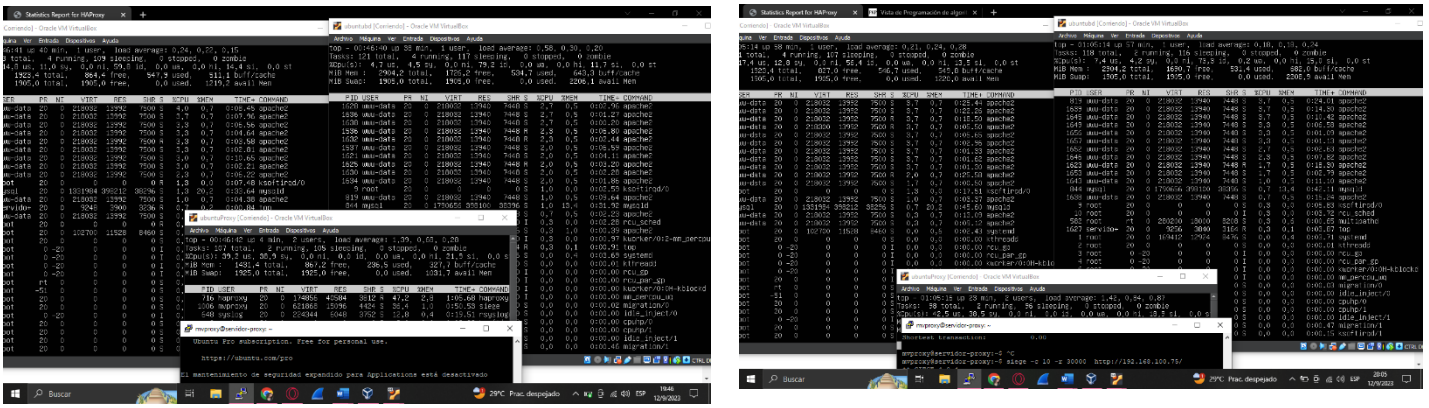
Anexo 9: Consumo de recursos y estadística Least Connection

Anexo 10: Consumo de recursos y estadística Round Robin

Escenario 3:



Anexo 11: Consumo de Round Robin y Least Connection



Escenario 4:

Anexo 12: Consumo de Round Robin y Least Connection

Anexo 13: Estadística Round Robin y Least Connection