

Hit #2

Después de completar el ejercicio y considerar el contenido teórico y práctico de los sistemas distribuidos vistos en clase, les solicitamos responder a los siguientes enunciados. Para ello, deben enfocarse en proporcionar respuestas concisas y orientadas a la implementación para aplicar en los siguientes Trabajos Prácticos que abordaremos en la asignatura ✨.

Relevancia y Utilidad del Servicio:

- ¿En qué escenarios específicos sería crucial y absolutamente necesario disponer de servicios como el desarrollado en este ejercicio?

En situaciones donde se necesite ejecutar tareas de manera remota, a demanda de los clientes. Este servicio sería esencial en aplicaciones que necesiten ejecutar tareas pre-compiladas de manera dinámica y sin tener que instalar nada en los clientes.

- ¿Cómo estos servicios podrían beneficiar a las empresas o proyectos en términos de eficiencia, escalabilidad y flexibilidad?

Eficiencia: Pequeños sistemas independientes que se levantan una vez y no consumen más recursos.

Escalable: Se pueden agregar distintas tareas de forma independiente sin afectar el funcionamiento de las demás.

Flexibilidad: Se puede ajustar a cualquier tipo de tarea específica que se pueda desarrollar con software.

Se adapta a diferentes requisitos sin necesidad de modificar el código del servidor.

Alternativas de stack tecnológico:

- Además de la arquitectura basada en un Servidor HTTP, ¿qué otras tecnologías alternativas podrían haber sido empleadas para implementar la ejecución de tareas remotas?

Se podrían haber empleado tecnologías como sockets TCP/IP para comunicación directa entre el cliente y el servidor, o WebSockets para permitir una comunicación bidireccional y en tiempo real. Estas tecnologías ofrecen una alternativa más personalizada en comparación con HTTP, pero también requieren una gestión más detallada de la conexión y pueden no ser tan adecuadas para cuando la simplicidad sea prioritaria.

- ¿Qué consideraciones deberían tenerse en cuenta al elegir una tecnología alternativa para garantizar la eficacia y la escalabilidad del sistema?

Al elegir una tecnología alternativa, es fundamental considerar aspectos como la capacidad de comunicación, escalabilidad, seguridad y facilidad de implementación y mantenimiento. La tecnología seleccionada debe poder manejar las solicitudes del cliente, escalar según las necesidades del sistema, garantizar la seguridad de las comunicaciones y ser de fácil implementación y mantención a largo plazo.

Desacoplamiento y Escalabilidad:

- A pesar de que la solución es escalable, se observa una limitación en términos de sincronización entre las partes. ¿Qué estrategias o técnicas podrían implementarse para desacoplar las diferentes partes del sistema y mejorar su escalabilidad?

La limitación se da en cuanto a la cantidad de mensajes. Una técnica es la cola de mensajes, por ejemplo RabbitMQ para desacoplar los componentes del sistema. Los mensajes pueden enviarse de manera asíncrona entre los diferentes servicios, lo que permite una comunicación más flexible y escalable.

- ¿Cómo afectaría la implementación de un sistema de mensajería o eventos en la arquitectura para abordar la limitación de sincronización y mejorar la escalabilidad del sistema?

La implementación de reintentos de mensaje por parte del servidor web en caso de que no haya respuesta del servidor de tareas, como puede ser RabbitMQ, es una estrategia para mejorar la fiabilidad y la tolerancia a fallos en la comunicación entre ambos, gestionando la comunicación asíncrona entre los diferentes servidores.

Esta estrategia garantiza que, en caso de que una solicitud no obtenga respuesta debido a problemas de red, fallos o sobrecarga del servidor web, el cliente pueda intentar nuevamente enviar la solicitud para asegurar que se procese correctamente.

- ¿Qué ventajas y desventajas tendría la introducción de un patrón de comunicación asíncrona en comparación con la comunicación síncrona actualmente utilizada?

Desacoplamiento: Permite que los componentes del sistema operen de manera independiente, lo que significa que no necesitan esperar la respuesta inmediata de otro componente antes de continuar con su propio trabajo. Esto reduce la dependencia entre los componentes.

Escalabilidad: Al desacoplar los componentes y permitir que operen de manera asíncrona, es más fácil escalar el sistema agregando más instancias de los componentes según sea necesario.

Seguridad y Autenticación:

- ¿Qué medidas de seguridad y autenticación deberían implementarse en este servicio para proteger los datos y garantizar la integridad de las transacciones entre el cliente y el servidor?

Para proteger los datos y garantizar la integridad de las transacciones entre el cliente y el servidor web, tendrán que implementar las siguientes medidas:

- Autenticación de lado del cliente: Esto se logra utilizando tokens de acceso o certificados. De esta forma, se garantiza que solo los usuarios autorizados pueden acceder y utilizar los servicios del servidor, verificando la identidad del cliente antes de procesar la solicitud.
- Autenticación por parte del servidor: Utilizando certificados SSL/TLS firmados por una autoridad confiable, es posible asegurar que el cliente

se comunique con un servidor legítimo, garantizando la integridad de los datos transmitidos.

- Autorización: Se determina las acciones que puede realizar el cliente en función de sus roles y permisos asociados, logrando limitar el acceso a recursos. Esto viene acompañado de un control de acceso a recursos, prohibiendo el acceso no autorizado a recursos del servidor.
- ¿Cómo se podría mejorar la seguridad de las comunicaciones entre el cliente y el servidor, especialmente al considerar la transferencia de datos sensibles?

Para mejorar la seguridad de la comunicación entre el cliente y el servidor, es necesario implementar el protocolo HTTPS, que incorpora a HTTP una capa adicional de seguridad mediante el uso de certificados SSL o TLS para cifrar la comunicación. Por lo tanto, al enviar datos sensibles, estos permanecen privados y protegidos contra terceros no autorizados.

Gestión de Errores y Resiliencia:

- ¿Qué estrategias deberían implementarse para gestionar errores y fallos en el servicio, tanto en el lado del cliente como en el del servidor?

Del lado del cliente:

Para gestionar errores y fallos en el servicio, es necesario implementar mecanismos para mostrar mensajes de error descriptivos y útiles en el caso de que ocurran problemas durante la interacción con el servidor.

También puede configurarse el cliente para que realice reintentos automáticos en caso de fallos como errores de red o temporales en el servidor, o reconexiones automáticas en el caso de detectar desconexiones inesperadas.

Del lado del servidor:

Como puntapié inicial frente a errores o fallos, es necesario implementar un sistema de registros de errores en el servidor que registre información detallada sobre cualquier error o fallo ocurrido, que pueden ser mensajes, datos de solicitud y códigos de estado que faciliten la depuración y solución del problema.

Una de las estrategias a considerar frente a fallos, es implementar mecanismos de respaldo y recuperación de datos para garantizar que no se pierda la información. Esto involucra la creación de copias de seguridad de las bases de datos y distribución de manera segura.

En base a lo anterior, es posible tener múltiples servidores con la misma información y lograr realizar una distribución del tráfico de red adecuado, logrando reducir la probabilidad de fallas del lado del servidor.

Por último, realizar pruebas de unidad e integración en el servidor pueden servir para identificar y corregir posibles puntos de fallo, garantizando el rendimiento y estabilidad del sistema.

- ¿Cómo se podría diseñar el sistema para ser más resiliente ante posibles fallos de red o problemas de disponibilidad de recursos?

El seguir adelante frente a situaciones adversas suele ser difícil, pero es posible diseñar el sistema utilizando diferentes técnicas con el fin de afrontar estas situaciones:

Partiendo de que nuestro sistema es distribuido con 2 nodos solamente, es posible definir mayor cantidad de nodos para distribuir la carga de trabajo y recursos, ya que por el momento tenemos un único punto de falla que es el servidor web.}

Pueden utilizarse colas de mensajes para almacenar las solicitudes del cliente antes de procesarlas, para suavizar la carga y reducir la probabilidad de pérdida de datos en el caso de que haya fallos en la red.

El uso de cache por problemas de disponibilidad de ciertos recursos resulta de gran ayuda ya que le permite al sistema continuar sirviendo datos almacenados de manera local mientras se intenta restaurar el servicio.

Destacamos también utilizar el protocolo SNMP para monitorizar el estado del sistema y detectar caídas o problemas de red, logrando responder rápidamente a fallos.

Monitorización y Diagnóstico:

- ¿Qué herramientas y técnicas podrían utilizarse para monitorear y diagnosticar el rendimiento y el estado del servicio en tiempo real?

El monitoreo y gestión de diagnóstico es algo fundamental en cualquier sistema, tanto en software como hardware. El protocolo SNMP permite realizar esta función a la perfección, fallas en dispositivos, caída de la red, conocer datos a tiempo real tales. Una herramienta será la encargada de utilizar el protocolo, como por ejemplo zabbix, y si se desea ver únicamente el tráfico de red se podría utilizar smokeping.

En caso de fallas, estas deberán quedar registradas en un archivo log, permitiendo no solo tener un registro de lo que ocurrió, sino que también saber de antemano como solucionar algún problema o poder evitarlo. También se podría utilizar para ver tendencias, saber a qué hora o en qué fecha es más propenso a que ocurra algo puede ser de gran ayuda para evitar cualquier imperfecto.

- ¿Qué métricas serían importantes de rastrear para evaluar el rendimiento y la eficacia del servicio?

El tiempo de respuesta es crucial a la hora de evaluar el rendimiento del sistema, saber cuánto tarda en llegar una solicitud es algo que siempre se debe tener en cuenta no solo para brindar un buen servicio al cliente, sino que también se utiliza como control. Sumado a esto está el tiempo de disponibilidad, siempre se trata de que el sistema funcione en tiempo y forma, si bien es imposible garantizar al 100% la disponibilidad, está siempre debe ser tenida en cuenta. Para eso se puede medir la tasa de errores, saber qué y cuándo puede fallar algo es crucial en cualquier sistema, permite tomar acciones preventivas y trata de hacer que no te agarren por sorpresa ningún imprevisto. Para lograrlo, conocer el tráfico en la red puede ser de gran ayuda, saber los momentos en lo que los servidores reciben más consultas hace que uno

pueda estar preparado y tomar las acciones necesarias, así como también conocer los momentos de menor tráfico permite llevar a cabo tareas como backups sin comprometer a los usuarios.

Escalabilidad y Rendimiento:

- ¿Cómo se podría escalar vertical u horizontalmente el servicio para manejar cargas de trabajo variables y picos de tráfico?

Horizontal: Lo ideal sería un balanceador de cargas, de esta forma los servidores web se distribuirán de una forma más óptima las peticiones.

Vertical: Destinar más recursos a los servidores o los servicios de tarea, si se conocen picos de actividades por parte del cliente, tener los servidores en la nube y brindar más potencia sólo cuando sea necesario podría ser una buena opción para ahorrar dinero.

- ¿Qué consideraciones de diseño y configuración podrían influir en el rendimiento y la escalabilidad del servicio a largo plazo?

Puede que haya peticiones más frecuentes que otras, como la utilización de cierto servicio de tarea. Por cómo está hecho el sistema, este servicio se levanta solo cuando es requerido, por lo que la opción de dejarlo levantado durante un tiempo a aquellos que suelen ser utilizados en mayor medida ayudaría a la escalabilidad al no destinar tantos recursos en encenderlo y apagarlo.