TECNOLÓGICA DEL ORIENTE
INSTITUCIÓN DE EDUCACIÓN SUPERIOR

SISTEMAS DISTRIBUTIVOS

ESCALABILIDAD, TOLERANCIA A FALLOS Y SEGURIDAD

ESCALABILIDAD, TOLERANCIA A FALLOS Y SEGURIDAD

La escalabilidad, la tolerancia a fallos y la seguridad son pilares fundamentales en el diseño y la operación de sistemas distribuidos, permitiendo que estos sistemas crezcan de forma eficiente, operen de manera confiable ante fallos y protejan la integridad y confidencialidad de los datos y servicios distribuidos (Muñoz Escoí, 2013).

Escalabilidad

La escalabilidad en sistemas distribuidos se refiere a la capacidad de manejar un aumento en la carga de trabajo de forma eficiente al añadir más recursos, como nodos o almacenamiento, sin comprometer el rendimiento del sistema.

Ejemplos.

- **Amazon Web Services (AWS).** Permite escalar servidores de manera automática para atender picos de tráfico en aplicaciones web.
- **Bases de datos distribuidas (Cassandra, MongoDB).** Permite añadir nodos para gestionar mayores volúmenes de datos sin interrupciones.

La escalabilidad puede ser:

- **Vertical.** Aumentar la capacidad de un solo nodo (CPU, memoria).
- **Horizontal.** Añadir más nodos al sistema, permitiendo un crecimiento distribuido y balanceado.

El uso de balanceadores de carga y particionamiento de datos (sharding) son estrategias comunes para mantener la escalabilidad en sistemas distribuidos.

Tolerancia a fallos

La tolerancia a fallos se refiere a la capacidad de un sistema distribuido de seguir funcionando correctamente ante fallos en algunos de sus componentes. Esta característica asegura la disponibilidad y confiabilidad del servicio, incluso cuando ocurren errores en nodos individuales o fallos en la red (Cardador Cabello, 2015).

Técnicas comunes

- **Replicación de datos.** Mantener múltiples copias de datos en diferentes nodos, para evitar pérdida de información.
- **Detección de fallos.** Uso de heartbeat o señales de latido, para verificar la disponibilidad de nodos.
- **Algoritmos de consenso (Raft, Paxos).** Para asegurar consistencia en la toma de decisiones ante fallos.

Ejemplo. En un sistema de almacenamiento distribuido como HDFS, si un nodo falla, los bloques de datos replicados en otros nodos, permiten continuar las operaciones sin pérdida de información.

Seguridad

La seguridad en sistemas distribuidos es esencial para proteger datos y servicios de accesos no autorizados, mantener la confidencialidad, integridad y disponibilidad, y garantizar la confianza de los usuarios en el sistema.

Aspectos claves:

- **Autenticación.** Verificar la identidad de usuarios o procesos que acceden al sistema. Ejemplo. Uso de OAuth 2.0 para autenticación segura en APIs.
- **Autorización.** Controlar los permisos y accesos a recursos distribuidos.
- **Cifrado.** Proteger los datos en tránsito (TLS) y en reposo mediante algoritmos de cifrado.
- **Auditoría y monitoreo.** Registrar actividades para detectar comportamientos anómalos y posibles brechas de seguridad.

Ejemplo práctico. Un sistema distribuido de banca en línea, utiliza cifrado SSL para proteger transacciones entre clientes y servidores, con autenticación multifactor y políticas de control de acceso, para proteger la integridad de los datos.

Interrelación entre los tres conceptos

- La escalabilidad permite atender más usuarios o procesar mayores volúmenes de datos, a medida que crece el sistema.
- La tolerancia a fallos asegura que, aunque algunos componentes fallen, el sistema continúe operativo.
- La seguridad protege el sistema mientras este se expande y enfrenta posibles amenazas externas o internas.

Un sistema distribuido bien diseñado equilibra estos tres aspectos para garantizar un rendimiento óptimo, una alta disponibilidad y un entorno confiable para los usuarios y las aplicaciones críticas.