

Universidad Nacional de Colombia

Facultad de Ingeniería

Computación Paralela y Distribuida

Profesor: Oscar Agudelo R.

Control de Lectura: ¿Qué es el SQL Distribuido?

Nombre del estudiante: Gabriel Andres Anzola Tachak

1. En el documento se menciona que los sistemas de gestión de bases de datos en la nube han mejorado para tratar de satisfacer el escalamiento elástico, la resiliencia y la ubicuidad, pero que siguen sin cumplirlas completamente y, aunque se ha avanzado, no se ha cerrado del todo la brecha. ¿Cuáles son los progresos, hasta la fecha, que presenta el texto como ya alcanzados?

El documento menciona varios progresos importantes alcanzados por las bases de datos en la nube hasta la fecha:

- **Eliminación de la necesidad de hardware físico:** Los desarrolladores ya no necesitan gestionar servidores físicos para sus aplicaciones, y los sistemas pueden escalar de manera automática según la demanda, lo que facilita la resiliencia ante fallos.
 - **Escalabilidad elástica:** Los sistemas pueden ajustar dinámicamente los recursos en función de las necesidades, lo que permite un uso eficiente y económico de los recursos de la nube.
 - **Ubiquidad:** La computación y el almacenamiento pueden estar disponibles en cualquier parte del mundo, permitiendo una experiencia rápida y consistente para los usuarios.
-

2. El documento cita cinco “nuevas” características que las bases de datos distribuidas deberían cumplir para atender el enfoque distribuido para el desarrollo de aplicaciones y las necesidades de las empresas. Describa brevemente estas cinco características.

1. **Escalabilidad:** Las bases de datos distribuidas deben ser capaces de aumentar o disminuir su capacidad de acuerdo con las demandas de la aplicación, asegurando una rápida adaptación a las necesidades de almacenamiento y procesamiento.
 2. **Resiliencia:** El sistema debe ser capaz de continuar operando incluso si una o más partes de la infraestructura fallan. Esto asegura que la base de datos sea altamente disponible y que los datos no se pierdan.
 3. **Localidad de los datos:** Los datos deben estar disponibles localmente para los usuarios y las aplicaciones, lo que garantiza una menor latencia y un mejor rendimiento.
 4. **Soporte para SQL:** Aunque existen sistemas NoSQL, los desarrolladores prefieren SQL por su familiaridad y su capacidad para manejar datos complejos de manera eficiente.
 5. **Cumplimiento de ACID:** Las bases de datos deben seguir los principios de atomicidad, consistencia, aislamiento y durabilidad para garantizar que los datos sean correctos y estén protegidos ante fallos.
-

3. Con base en el teorema CAP, explique que ofrecen las bases de datos del grupo “CA”, las bases de datos del grupo “AP” y las bases de datos del grupo “CP”. Liste ejemplos de bases de datos pertenecientes a cada grupo.

- **Bases de datos del grupo CA (Consistencia y Disponibilidad):** Estas bases priorizan la consistencia y la disponibilidad, pero no soportan tolerancia a particiones. Ejemplos incluyen **Microsoft SQL Server, Oracle y MySQL**.
 - **Bases de datos del grupo AP (Disponibilidad y Tolerancia a Particiones):** Estas bases priorizan la disponibilidad y la tolerancia a particiones, a costa de la consistencia eventual. Ejemplos: **Apache Cassandra, MongoDB, y Amazon DynamoDB**.
 - **Bases de datos del grupo CP (Consistencia y Tolerancia a Particiones):** Estas bases priorizan la consistencia y la tolerancia a particiones, sacrificando algo de disponibilidad. Ejemplos: **Google Spanner, CockroachDB, y YugabyteDB**.
-

4. Para el protocolo Raft, describa de manera breve qué ocurre en el proceso de “Elección del líder” y lo que ocurre en el proceso de “Replicación del registro”.

- **Elección del líder:** En el protocolo Raft, los nodos eligen a un líder que actúa como la fuente de verdad en el grupo. Cuando un nodo no recibe señales ("heartbeats") del líder, se postula como candidato y solicita votos de otros nodos. Si obtiene la mayoría de los votos, se convierte en el líder.
 - **Replicación del registro:** El líder mantiene un registro de todas las transacciones y lo replica en los nodos seguidores. Para que una transacción sea confirmada, la mayoría de los nodos deben replicar correctamente el registro.
-

5. ¿Qué es Multiversion Concurrency Control (MVCC)?

El **Multiversion Concurrency Control (MVCC)** es un mecanismo que permite que múltiples transacciones lean y escriban en una base de datos de manera concurrente, asegurando que los datos sean consistentes. Esto se logra manteniendo múltiples versiones de los datos con marcas de tiempo, permitiendo que las lecturas accedan a una versión específica sin interferir con las actualizaciones en curso.
