El futuro digital es de todos

MinTIC

Blockchain & analítica de datos

para industrias digitales

etraining
Diseñamos nuevas formas de aprender

tecnalia Colombia
Inspiring Business

El futuro digital es de todos

MinTIC

Analítica de Datos

Grupo N°05

Sesión N°07

etraining
Diseñamos nuevas formas de aprender

tecnalia Colombia
Inspiring Business

Teorema CAP

Teorema CAP

El teorema CAP enuncia que es imposible para un sistema de cómputo distribuido garantizar simultáneamente:

- **La consistencia** (**C**onsistency)
- **La disponibilidad** (**A**vailability)
- **La tolerancia al particionado** (**P**artition Tolerance)

Según el teorema, un sistema no puede asegurar más de dos de estas tres características simultáneamente.

"Cheap, Fast, and Good: Pick Two"

Teorema CAP

Consistencia

La **Consistencia** significa que todos los clientes ven los mismos datos al mismo tiempo, sin importar a qué nodo se conecten. Para que esto suceda, cada vez que se escriben datos en un nodo, se deben reenviar o replicar instantáneamente a todos los demás nodos del sistema antes de que la escritura se considere "exitosa".

Disponibilidad

Disponibilidad significa que cualquier cliente que solicite datos obtiene una respuesta, incluso si uno o más nodos están inactivos. Otra forma de expresar esto: todos los nodos de trabajo en el sistema distribuido devuelven una respuesta válida para cualquier solicitud, sin excepción.

Tolerancia de partición

Una **partición** es una interrupción de las comunicaciones dentro de un sistema distribuido: una conexión perdida o temporalmente retrasada entre dos nodos. La tolerancia de partición significa que el clúster debe continuar funcionando a pesar de cualquier número de interrupciones de comunicación entre los nodos del sistema.

Explicación del Teorema CAP

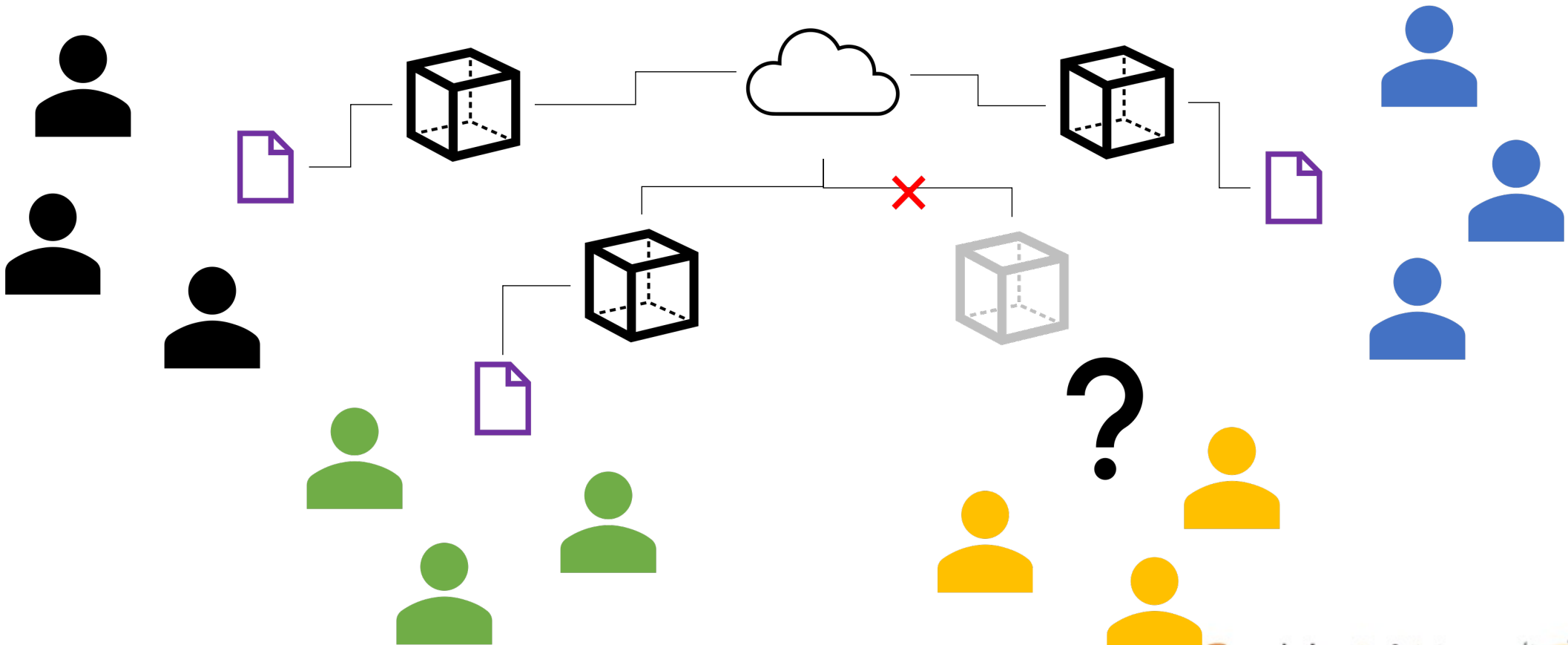
Ningún sistema distribuido está a salvo de las fallas de la red, por lo tanto, la partición de la red generalmente tiene que ser tolerada. En presencia de una partición, una se queda con dos opciones: consistencia o disponibilidad. Al elegir la consistencia sobre la disponibilidad, el sistema devolverá un error o un tiempo de espera si no se puede garantizar que la información particular esté actualizada debido a la partición de la red. Al elegir la disponibilidad por coherencia, el sistema siempre procesará la consulta e intentará devolver la versión disponible más reciente de la información, incluso si no puede garantizar que esté actualizada debido a la partición de la red.

En ausencia de falla de la red, es decir, cuando el sistema distribuido se está ejecutando normalmente, se puede satisfacer tanto la disponibilidad como la consistencia.

Con frecuencia, la CAP se malinterpreta como si uno tuviera que elegir abandonar una de las tres garantías en todo momento. De hecho, la elección es realmente entre la consistencia y la disponibilidad solo cuando ocurre una partición de red o falla; en cualquier otro momento, no hay que hacer concesiones.

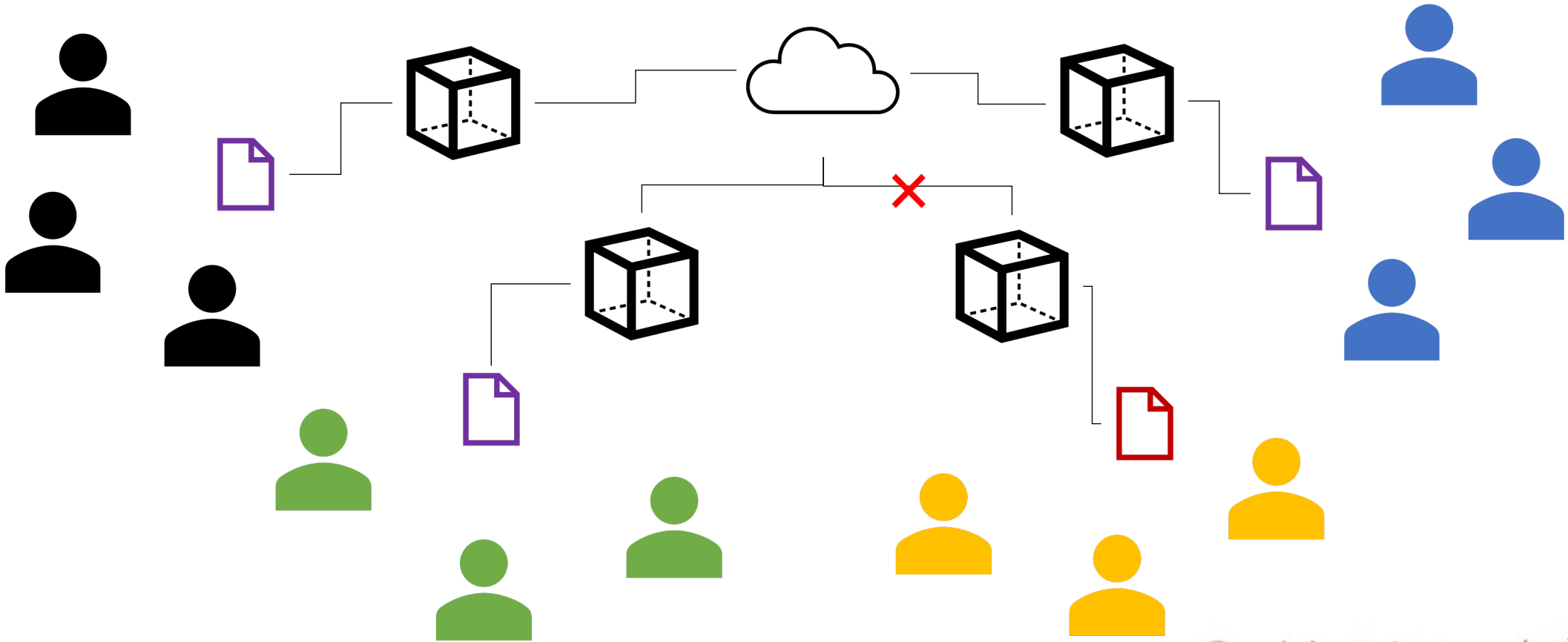
Teorema CAP

Consistencia y Tolerancia de partición



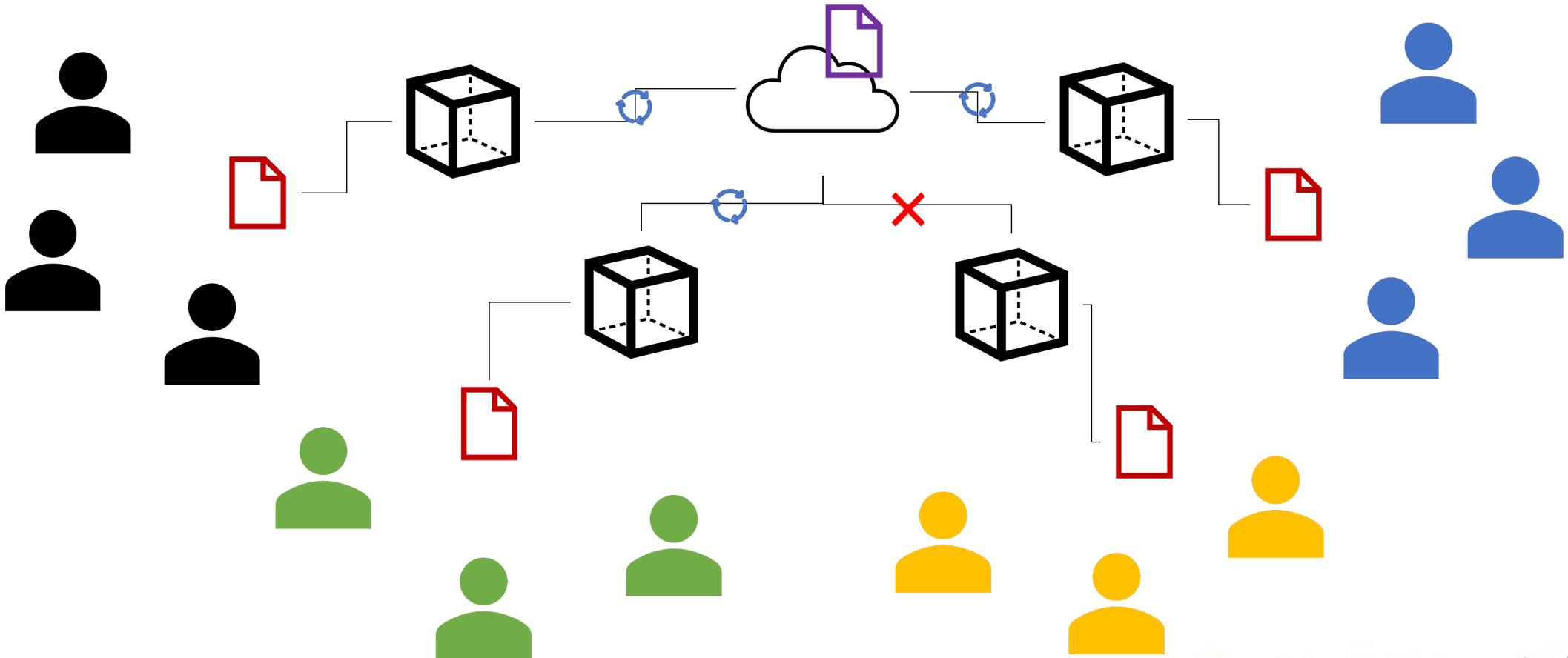
Teorema CAP

Disponibilidad y Tolerancia de partición

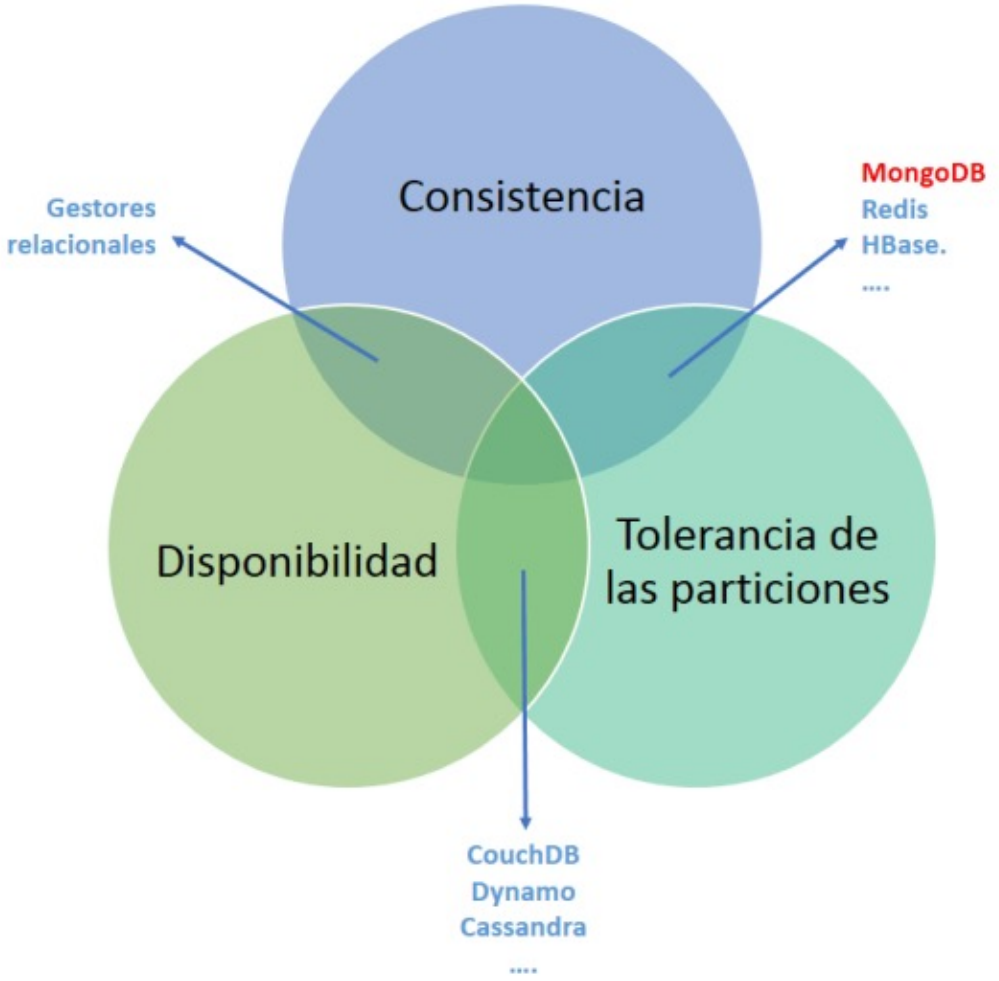


Teorema CAP

Disponibilidad y Consistencia



Explicación del Teorema CAP



Caso Aplicado 1

Contexto

- Mis datos son de tipo transaccional.
- Mi software se encarga de llevar el inventario de una compañía así como las ventas y compras.
- En cada momento se debe garantizar la integridad de los datos y no puedo permitir que un usuario reciba información desactualizada acerca del inventario.

Preguntas

¿Qué tipo de base de datos debo utilizar?

¿Qué partes del teorema CAP cumple esta base de datos?

¿Debo almacenar esta información en la nube o lo puedo hacer en mis servidores locales?

Caso Aplicado 2

Contexto

- Mis datos son transaccionales.
- Todos los usuarios deben recibir la misma información en cada momento.
- Se generan muchísimos datos por lo que un sólo servidor no sería suficiente para aceptar el gran volumen de datos.

Preguntas

¿Qué tipo de base de datos debo utilizar?

¿Qué partes del teorema CAP cumple esta base de datos?

¿Debo almacenar esta información en la nube o lo puedo hacer en mis servidores locales?

MongoDB

Se utiliza con frecuencia para aplicaciones de big data y en tiempo real que se ejecutan en varias ubicaciones diferentes.

En relación con el teorema de CAP, MongoDB es un almacén de datos de CP: resuelve las particiones de red manteniendo la coherencia, al tiempo que compromete la disponibilidad.

MongoDB es un sistema maestro único: cada conjunto de réplicas puede tener solo un nodo principal que recibe todas las operaciones de escritura. Todos los demás nodos del mismo conjunto de réplicas son nodos secundarios que replican el registro de operaciones del nodo principal y lo aplican a su propio conjunto de datos.

Cuando el nodo principal deja de estar disponible, el nodo secundario con el registro de operaciones más reciente se elegirá como el nuevo nodo principal. Una vez que todos los demás nodos secundarios se ponen al día con el nuevo maestro, el clúster vuelve a estar disponible. Como los clientes no pueden realizar solicitudes de escritura durante este intervalo, los datos se mantienen consistentes en toda la red.

Caso Aplicado 3

Contexto

- Mis dato necesitan transaccionalidad.
- Puedo permitir que un usuario reciba información desactualizada acerca del estado de la aplicación.
- Se generan muchísimos datos por lo que un sólo servidor no sería suficiente para aceptar el gran volumen de datos.

Preguntas

¿Qué tipo de base de datos debo utilizar?

¿Qué partes del teorema CAP cumple esta base de datos?

¿Debo almacenar esta información en la nube o lo puedo hacer en mis servidores locales?

Cassandra

Es una base de datos de columna ancha que le permite almacenar datos en una red distribuida. Sin embargo, a diferencia de MongoDB, Cassandra tiene una arquitectura sin maestro y, como resultado, tiene múltiples puntos de falla, en lugar de uno solo.

En relación con el teorema CAP, Cassandra es una base de datos AP: ofrece disponibilidad y tolerancia a la partición, pero no puede ofrecer coherencia todo el tiempo. Debido a que Cassandra no tiene un nodo principal, todos los nodos deben estar disponibles de forma continua. Sin embargo, Cassandra proporciona consistencia eventual al permitir a los clientes escribir en cualquier nodo en cualquier momento y reconciliar las inconsistencias lo más rápido posible.

Como los datos solo se vuelven inconsistentes en el caso de una partición de red y las inconsistencias se resuelven rápidamente, Cassandra ofrece la funcionalidad de "reparación" para ayudar a los nodos a ponerse al día con sus pares. Sin embargo, la disponibilidad constante da como resultado un sistema de alto rendimiento que podría valer la pena en muchos casos.

Revisemos el caso de tu empresa

- ¿Hoy en día cómo almaceno mis datos?
- ¿Necesito una base de datos relacional?
- ¿Mis datos se generan en un gran volumen de datos?
- ¿Mis datos tienen una gran variedad?
- ¿Mis datos se generan a gran velocidad?
- ¿Qué infraestructura tengo hoy para enfrentar el Big Data?
- ¿Qué desafíos tengo dentro de mi empresa para afrontar el Big Data?
- ¿Nos conviene utilizar una base de datos particionada?
- ¿Debe mantener mi empresa consistencia en los datos al particionarlos?
- ¿Qué nos haría falta para iniciarnos en el mundo del Big Data?
- ¿Necesitaría dinero externo a mi compañía para iniciar en Big Data?
- ¿Me conviene utilizar servicios en la nube para almacenar datos?

El futuro digital es de todos

MinTIC

Gracias

etraining
Diseñamos nuevas formas de aprender

tecnalia Colombia
Inspiring Business