IIC2523
Sistemas Distribuidos

Hernán F. Valdivieso López
(2025 - 2 / Clase 13)

Encuesta Temprana de Medio Semestre

Aspectos positivos

- Organización del curso
- Buenas diapositivas
- Dinámica del profesor

Aspectos por mejorar

- Relacionar más la materia con ejemplos reales
- Aspectos administrativos (tareas y programa)
- Nada

Compromisos

- Ir a clases y participar
- Repasar con tiempo la materia
- Nada

Replicación de Datos II
¿Qué protocolos tenemos para replicar?

Temas de la clase

1. Protocolos de replicación
    a. *Gossip*
    b. *Quorum-Based*
    c. *Primary-Based Protocols*
    d. *Replicated-Write Protocols*
2. Tecnologías que ocupan estos Protocolos

Protocolos de Replicación

Gossip

Quorum-Based

Primary-Based Protocols

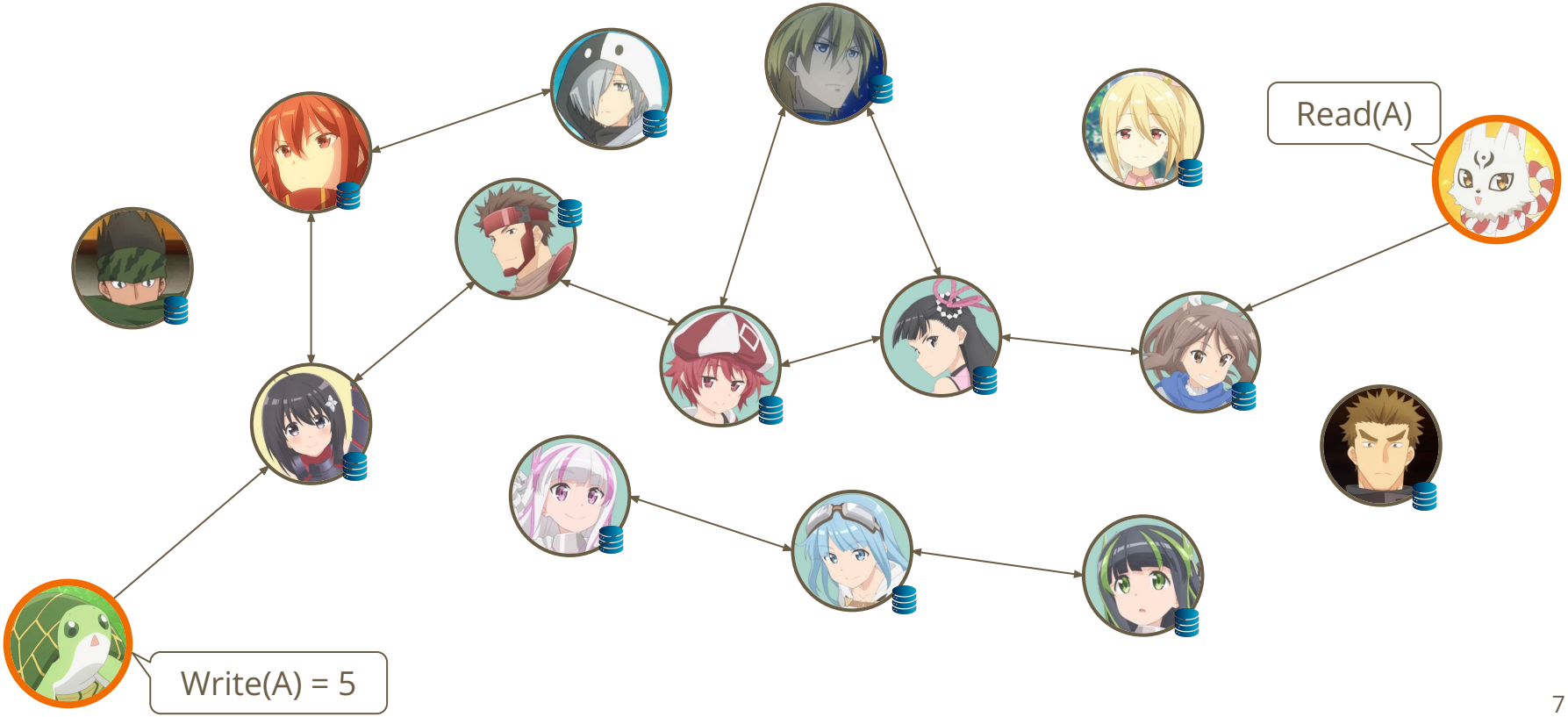
Replicated-Write Protocols

¿Cómo se ve reflejado en la actualidad?

Introducción

- Son conjunto de reglas que definen el proceso para replicar la información o incluso, cómo comunicarse.
- Algunos protocolos son no excluyentes, es decir, se pueden mezclar con otros para garantizar ciertos beneficios o restricciones.
- Estudiaremos 4 protocolos:
  - *Gossip*
  - *Quorum-Based*
  - *Primary-Based Protocols*
  - *Replicated-Write Protocols*

Protocolos de Replicación - *Gossip*



Protocolos de Replicación - *Gossip*

- También conocido como Protocolo Epidémico.
  - Un nodo "infectado" con nuevos datos los propaga a sus vecinos directos, y ellos a su vez lo propagan a los demás.
- Un enfoque *lazy* para la propagación de actualizaciones. Las réplicas intercambian información sobre sus actualizaciones de estado más recientes de forma periódica con un subconjunto de sus vecinos directos.

Protocolos de Replicación - *Gossip*

Consistencia

- Se espera que todas las réplicas tengan la misma información, pero no hay garantías sobre cuándo ocurrirá o en qué orden exacto verán las actualizaciones.
  - Consistencia eventual.

Protocolos de Replicación - *Gossip*

Consistencia

- Se espera que todas las réplicas tengan la misma información, pero no hay garantías sobre cuándo ocurrirá o en qué orden exacto verán las actualizaciones.
  - Consistencia eventual.
- Si los mensajes respetan FIFO entre nodos y se utilizan relojes vectoriales, se puede asegurar consistencia causal.
  - 🤔 ¿Por qué? 🤔

Protocolos de Replicación - *Gossip*

Consistencia

- Se espera que todas las réplicas tengan la misma información, pero no hay garantías sobre cuándo ocurrirá o en qué orden exacto verán las actualizaciones.
  - Consistencia eventual.
- Si los mensajes respetan FIFO entre nodos y se utilizan relojes vectoriales, se puede asegurar consistencia causal.


Nodo K $W_k(x)=1$


Nodo F $R_f(x)\rightarrow 1$ $W_f(x)=2$ $W_f(z)=1$


Nodo S $R_s(z)\rightarrow 1$ (--) $R_s(X)$ [Esto debería ser 2]

Protocolos de Replicación - *Gossip*

Consistencia

- Se espera que todas las réplicas tengan la misma información, pero no hay garantías sobre cuándo ocurrirá o en qué orden exacto verán las actualizaciones.
  - Consistencia eventual.
- Si los mensajes respetan FIFO entre nodos y se utilizan relojes vectoriales, se puede asegurar consistencia causal.

 Nodo K $W_k(x)=1$

 Nodo F $R_f(x)\rightarrow 1$ $W_f(x)=2$ $W_f(z)=1$

 Nodo S $R_s(z)\rightarrow 1$ (--) $R_s(X)$ [Esto debería ser 2]

Si $W_k(x)=1$ llegue en (--); con relojes vectoriales y sus componentes de A y B del reloj, es posible notar que $W_f(x)=2$ viene después que $W_k(x)=1$ y no ejecutar esa operación en la base de datos.

Protocolos de Replicación - *Gossip*

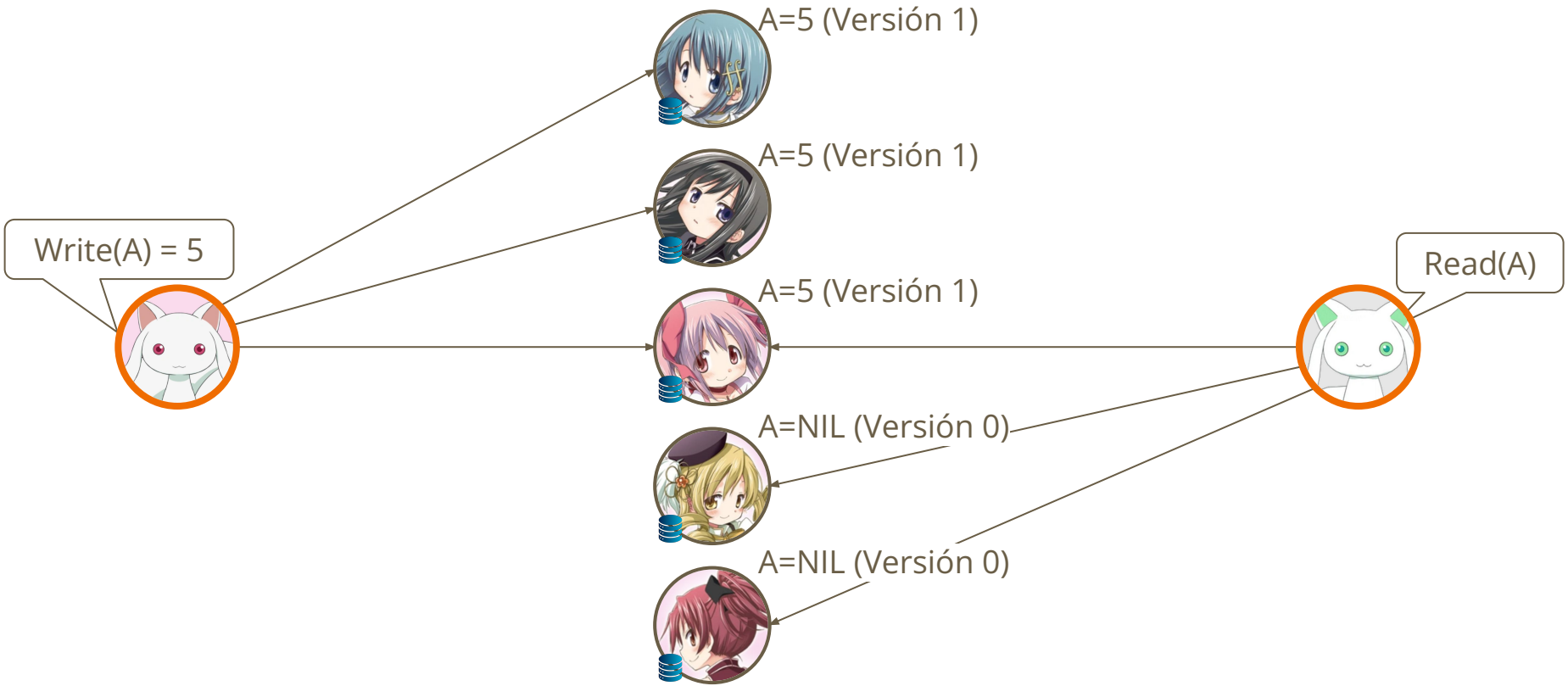
Ventajas

- Extremadamente robusto, tolerante a fallas de nodos y escalables para sistemas muy grandes y dinámicos.
- No requieren un coordinador central.
- Muy directo de implementar.

Desventajas

- No son adecuados para aplicaciones que requieren consistencia estricta e inmediata, por ejemplo, transacciones financieras.
- La consistencia eventual puede tomar más tiempo de lo que uno desea.
- Alta redundancia de mensajes.

Protocolos de Replicación - *Quorum-Based*



Protocolos de Replicación - *Quorum-Based*

- Para realizar operaciones de lectura o escritura, un subgrupo de réplicas debe participar y dar su "voto", formando un quórum.
- Reglas de Quórum
  - Sea **N** la cantidad de nodos, **RQ** el quórum de lectura y **WQ** el quórum de escritura.
  - Para obtener un dato, se debe consultar a RQ nodos y el dato más reciente es el elegido.
  - Para escribir un dato, se debe obtener previamente la aprobación de WQ nodos.
  - La consistencia se garantiza si la intersección de cualquier quórum de lectura y cualquier quórum de escritura es no vacía: `(RQ + WQ > N)`
  - Se exige `WQ > N/2` para evitar conflictos de escrituras.
  - Para cualquier operación, si no se logra el quórum esperado, la operación no se ejecuta.
- Si una réplica quiere leer un dato, puede usar el suyo, pero igual debe respetar el quórum.

Protocolos de Replicación - *Quorum-Based*

Consistencia (asumiendo `RQ + WQ > N`)

- Previene inconsistencia al momento de obtener el dato.
  - Dado que al menos existirá 1 nodo que será parte de lectura y escritura. Así que siempre se tendrá el dato más reciente.
  - Pueden existir réplicas con datos que no se han actualizado con el cambio de un WRITE
- Se garantiza la consistencia secuencial.

Protocolos de Replicación - *Quorum-Based*

Ventajas (asumiendo `RQ + WQ > N`)

- Es muy robusto ante particiones de red, ya que solo la partición que logra formar un quórum puede continuar operando, previniendo inconsistencias.
- Mientras se respete el Quórum, tolera una alta cantidad de fallas de nodos.

Desventajas

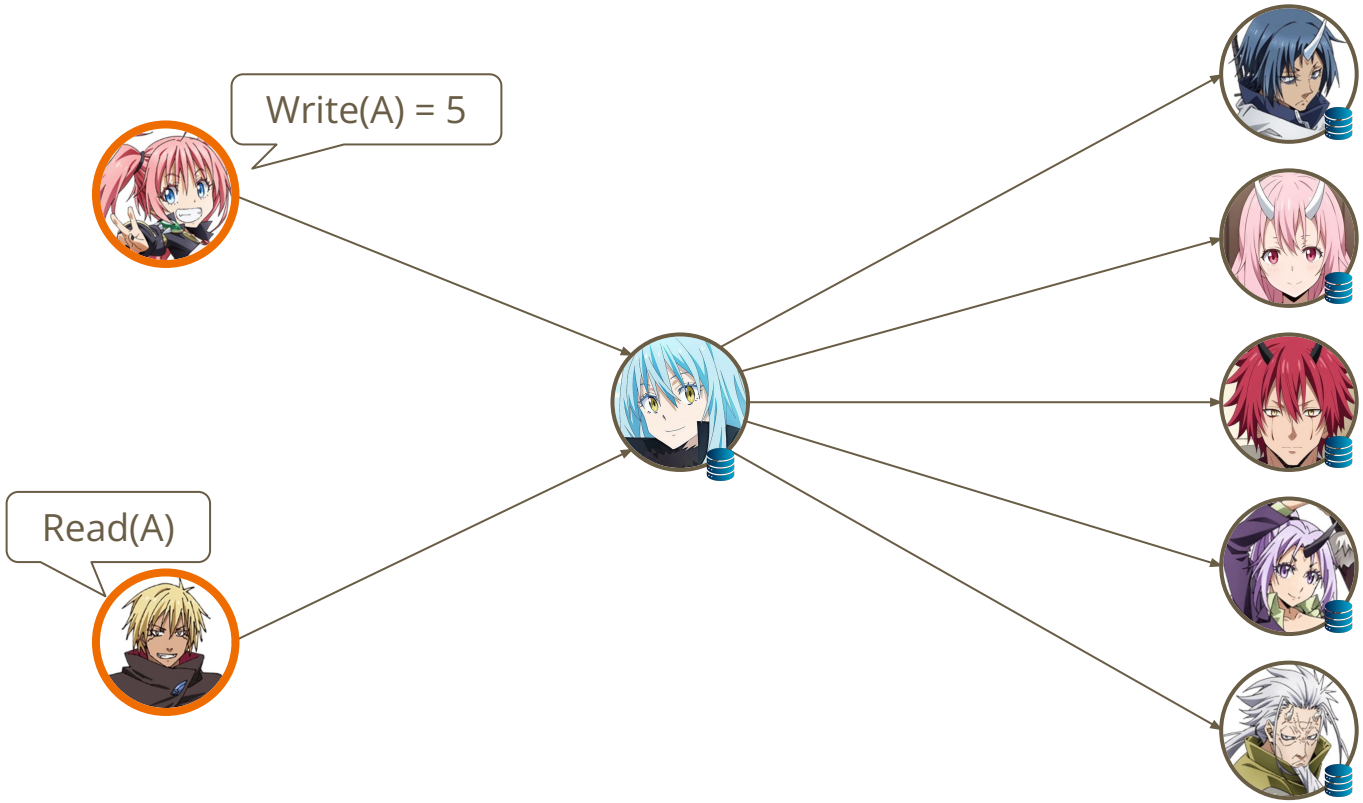
- Mayor latencia de las operaciones
- La probabilidad de que no haya disponibilidad va a aumentar.

Protocolos de Replicación - *Quorum-Based*

Variaciones en Quórum

- *Read-One Write-All* (ROWA) RQ = 1 y WQ = N. Rápida lectura pero lenta escritura.
- *Majority Quorum (Voting-Based Quorum)*. Tanto RQ y WQ deben ser mayor a N/2.
- Weighted Voting. Cada nodo tiene un peso, y el quórum se define en función de la suma de pesos, en vez de la cantidad de nodos.
- *Dynamic Voting*. Ajusta el N, RQ y WQ según la cantidad de nodos activos.

Protocolos de Replicación - *Primary-Based Protocols*



Protocolos de Replicación - *Primary-Based Protocols*

- También conocido como Replicación Pasiva, este protocolo se basa en un único gestor de réplicas primario que recibe todas las solicitudes de los clientes.
- El primario ejecuta las operaciones (especialmente las escrituras) y luego propaga el nuevo estado a los gestores de réplicas secundarios (también llamados *backups* o esclavos).
- En caso de que el gestor primario falle, se elige un nuevo nodo entre los secundarios para que asuma el rol de primario y continúe con la operación del sistema.

Protocolos de Replicación - *Primary-Based Protocols*

Consistencia

- Si la lectura y escritura se realiza exclusivamente con el nodo primario, se puede garantizar consistencia fuerte.
- Si el nodo primario falla, se puede mantener la consistencia fuerte solo si el nuevo nodo primario (promovido desde una réplica) ha recibido y aplicado todas las actualizaciones previas al fallo.
- Si los clientes pueden leer directamente desde las réplicas (una variante común del protocolo), ya no se puede garantizar consistencia fuerte, pero sí se mantiene consistencia secuencial.

Protocolos de Replicación - *Primary-Based Protocols*

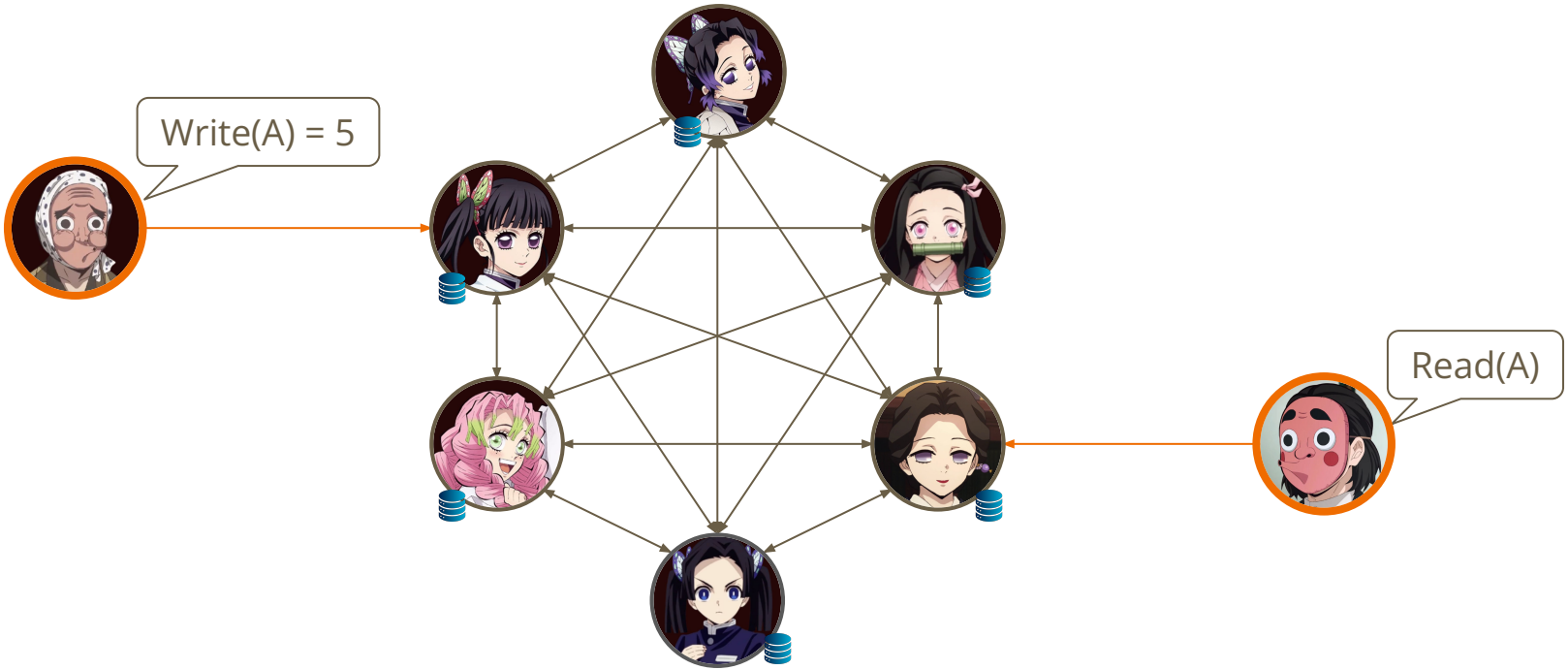
Ventajas

- Eficiencia en escrituras consecutivas. Se empaquetan varias operaciones como una gran operación y transmitiendo a las réplicas sólo el estado final.
- Relajando la consistencia, se puede separar el proceso de escritura y lectura para reducir la carga del nodo primario.
- Manejo de operaciones no deterministas. Se puede disponer de múltiples *thread* para procesar operaciones concurrentemente y solo transmitir el resultado final.

Desventajas

- Cuello de botella en el primario.
- Tolerancia a fallos limitada.

Protocolos de Replicación - *Replicated-Write Protocols*



Protocolos de Replicación - *Replicated-Write Protocols*

- También conocido como Replicación Activa, este protocolo se basa en que todos los gestores de réplicas están activos y procesan las solicitudes de los clientes de forma concurrente.
- Para garantizar la consistencia entre réplicas, se utiliza un mecanismo de *Totally Ordered Multicast (TOM)*.
  - Se asigna un número de secuencia global a la operación (por un coordinador o por consenso entre nodos).
  - Cada nodo ejecuta las operaciones respetando el número de secuencia.
- En algunas variantes, se pueden usar protocolos basados en quórum para optimizar el procesamiento y reducir la cantidad de réplicas necesarias para aceptar una operación.

Protocolos de Replicación - *Replicated-Write Protocols*

Consistencia

- El orden total en el que los gestores de réplicas procesan las solicitudes no es necesariamente el mismo que el orden en tiempo real en que los clientes realizan sus solicitudes. Por lo que no se cumple consistencia fuerte.
  - Usando sincronización de relojes físicos se podría encaminar consistencia fuerte, pero dependerá de las características físicas del sistema y la red.
- Si los clientes no se comunican con otros clientes mientras esperan respuestas a sus solicitudes, se garantiza consistencia secuencial dado el *Totally Ordered Multicast*.

Protocolos de Replicación - *Replicated-Write Protocols*

Ventajas

- Hay una alta tolerancia a fallos ante caída de algunos nodos y no existe un único punto de fallo.
- Se puede reducir el peso de los mensajes comunicando solo los cambios del sistema en vez del estado del sistema (Piensen en los *commits* de *Git*).

Desventajas

- Costo de coordinación alto.
- Escalabilidad limitada.
- Latencia de escritura alta.
- Alta tasa de mensajes, incluso muchos redundantes.

Protocolos de Replicación

¿Cómo se ven reflejados en la actualidad?

- *Practical Byzantine Fault Tolerance* (pBFT) se inspira en *Replicated-Write Protocols* para aceptar o no el cambio.
- Azure SQL Database utiliza *Primary-Based Protocols*, de forma asíncrona, para replicar la información en hasta cuatro réplicas geográficas.
- *MySQL Server* utiliza *Primary-Based Protocols* para replicar los datos. Puede ser asíncrono (responde antes de replicar) o síncrono (responder después de replicar).
- Consul (administrador de red privada para servicios distribuidos) utiliza una variación del protocolo *gossip* (SWIM) para administrar los servicios de la red y comunicar los mensajes.
- *Google Spanner* (servicio de base de datos distribuido) utiliza *Quorum* para la escritura de datos.

Protocolos de Replicación

¿Cómo se ven reflejados en la actualidad?

- Apache Cassandra (Base de datos NoSQL) utiliza el protocolo *Gossip* para la comunicación entre nodos y el protocolo basado en *Quorum* para las réplicas de datos.
- Amazon Dynamo (Base de datos NoSQL de uso interno de Amazon) utiliza el protocolo *Gossip* para trackear el estado del sistema y una variación del protocolo basado en *Quorum* para la consistencia de las réplicas.
- Google file system (GFS) utiliza *Primary-Based Protocols* para garantizar la consistencia, especialmente para las mutaciones de datos. No obstante, exige que todas las réplicas hayan completado con éxito la mutación antes de indicar que la operación fue exitosa.

Poniendo a prueba lo que hemos aprendido 👀

Si un cliente hace una solicitud a un nodo que incluye una operación no determinista, por ejemplo, depende del tiempo o de consultar un servicio externo cuya respuesta puede cambiar según el tiempo que se le consulte.

El cliente sabe el resultado de su solicitud si ésta es procesada dentro de los próximos 2 minutos.

Asumiendo que no existe ningún tipo de conflicto en la operación ¿Cuál protocolo **NO** garantiza que la respuesta obtenida sea igual a la esperada?

a. *Gossip*
b. *Quorum-Based*
c. *Primary-Based Protocols*
d. *Replicated-Write Protocols*
e. Ninguno de las anteriores.

Poniendo a prueba lo que hemos aprendido 👀

Si un cliente hace una solicitud a un nodo que incluye una operación no determinista, por ejemplo, depende del tiempo o de consultar un servicio externo cuya respuesta puede cambiar según el tiempo que se le consulte.

El cliente sabe el resultado de su solicitud si ésta es procesada dentro de los próximos 2 minutos.

Asumiendo que no existe ningún tipo de conflicto en la operación ¿Cuál protocolo **NO** garantiza que la respuesta obtenida sea igual a la esperada?

a. *Gossip*

b. *Quorum-Based*

c. *Primary-Based Protocols*

d. *Replicated-Write Protocols*

e. Ninguno de las anteriores.

Próximos eventos

Próxima clase

- Transacciones Distribuidas
  - ¿Cómo administramos un conjunto de operaciones como una única gran operación?

Evaluación

- Lunes, a las 13:30, se publica el control 4 que evalúa hasta esta clase. Tendrán 1 semana para hacerlo.
- Investigación, ya liberé el enunciado con los temas. Todavía no deben empezar la investigación como tal, solo inscribirse y, opcionalmente, indicar los temas de interés y/o proponer un nuevo tema.

IIC2523
Sistemas Distribuidos

Hernán F. Valdivieso López
(2025 - 2 / Clase 13)

Créditos (animes utilizados)

Itai no wa Iya nano de Bōgyoryoku ni Kyokufuri Shitai to Omoimasu (Bofuri)

Tensei shitara Slime Datta Ken

Kimetsu no Yaiba

Puella Magi Madoka Magica