Replicación de Datos

Sistemas Distribuidos

Concepto de Replicación

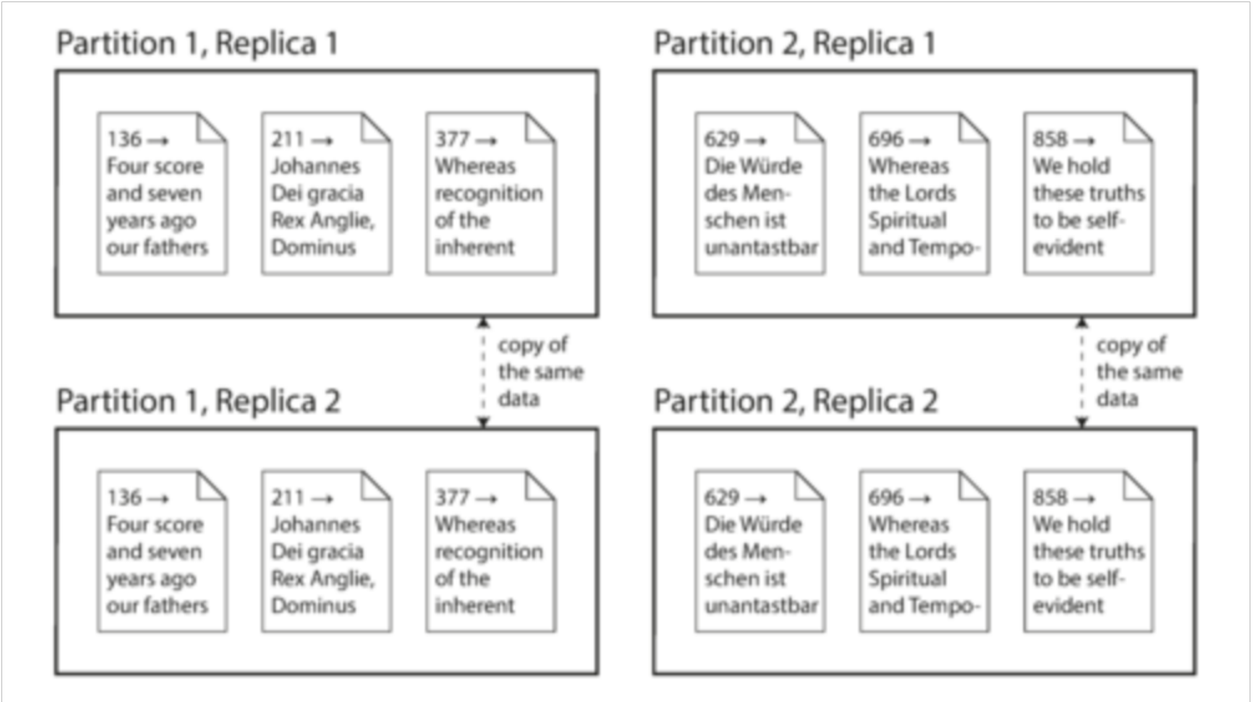
- Replicar: Copiar algo exactamente.
- Replicación se puede comprender en términos de lograr tener una copia de de un mismo datos sobre múltiples instancias.
  - Cada instancia se conoce como replica.
- La replicación es un concepto fundamental en los sistemas distribuidos con el fin de mejorar los servicios.
  - Aspecto fundamental para implementar la tolerancia a fallas.
- La replicación es una forma de introducir la redundancia en los sistemas distribuidos.

Motivación para la Replicación

- Mantener los datos los mas cercano posible a los usuarios…
- Permitir al sistema funcionar en presencia de fallas…
- Básicamente se trata de mejorar la confiabilidad (reliability) y el desempeño (performance) del sistema.
- Retos:
  - Uno de los grandes retos de la replicación tiene que ver con la consistencia.
  - Esto trata sobre el hecho de una vez se actualiza un dato, se debe garantizar que las replicas también se actualicen.

Replicación vs Particionamiento

- La replicación permite mantener la copia del mismo dato sobre múltiples nodos. Redundancia.
- El particionamiento divide los datos en múltiples partes
  - También conocido como “sharding”



Algoritmos para Replicar Cambios en Nodos

Algoritmos para Replicación de Datos

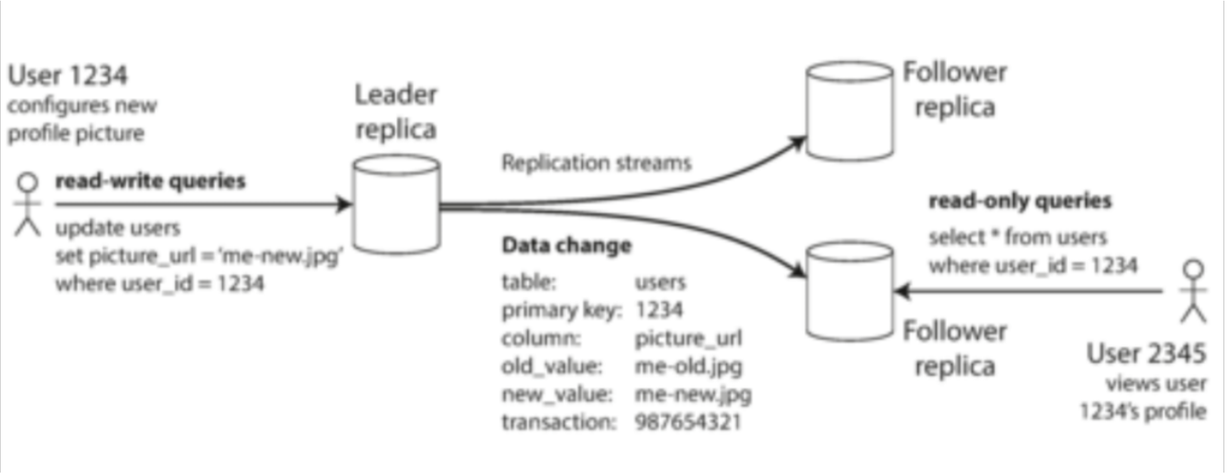
- Single Leader (Leaders and Followers)
- Multi-leader.
- Leaderless.
- En estas aproximaciones podemos encontrar que pueden ser:
  - Sincrónicos vs Asincrónicos.

Leaders and Followers

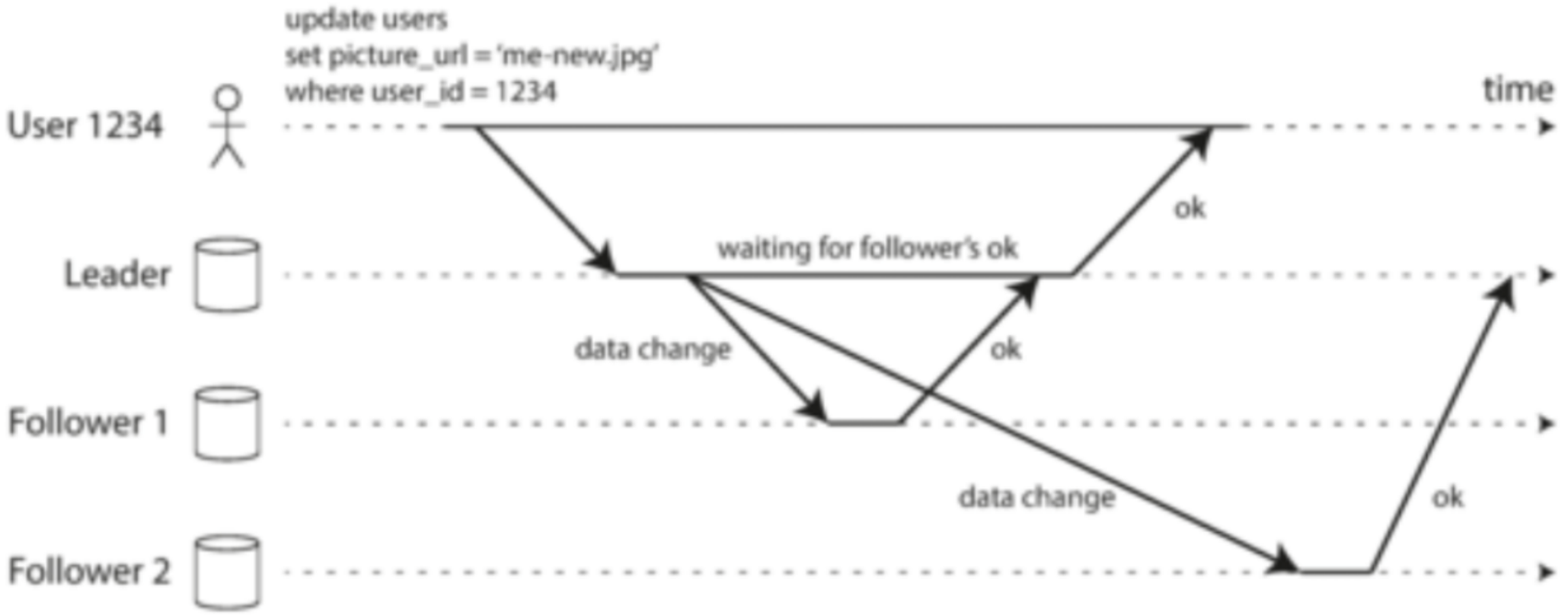
- Cada nodo que almacena una copia de la base de datos, se denomina replica.
- En presencia de múltiples replicas, el gran interrogante a resolver es como garantizamos que un dato finalice en todas las replicas.
- Toda operación de lectura a la base de datos debe ser procesada por todas y cada una de las replicas.
  - De otra forma las replicas no tendrían consistencia en los datos.
- Esta técnica también se conoce como replicación active/pasive o master/slave.

Leaders and Followers

- En este escenario una de las replicas es designada a ser el líder (master).
- Cuando se desea realizar una operación de escritura en la base de datos, el cliente contacta a la replica líder y efectúa la operación, de carácter local, en esta replica.
- El lider procede a notificar el cambio en el dato al resto de replicas (slaves/passive/stand-by) a través de un log de replicación o stream de cambios.
    - Los slaves deben ejecutar esas operaciones en el mismo orden.
- Finalmente, ya los clientes pueden realizar sus consultas bien sea del líder o de los seguidores.



Replicación: Sincrónico vs Asincrónico

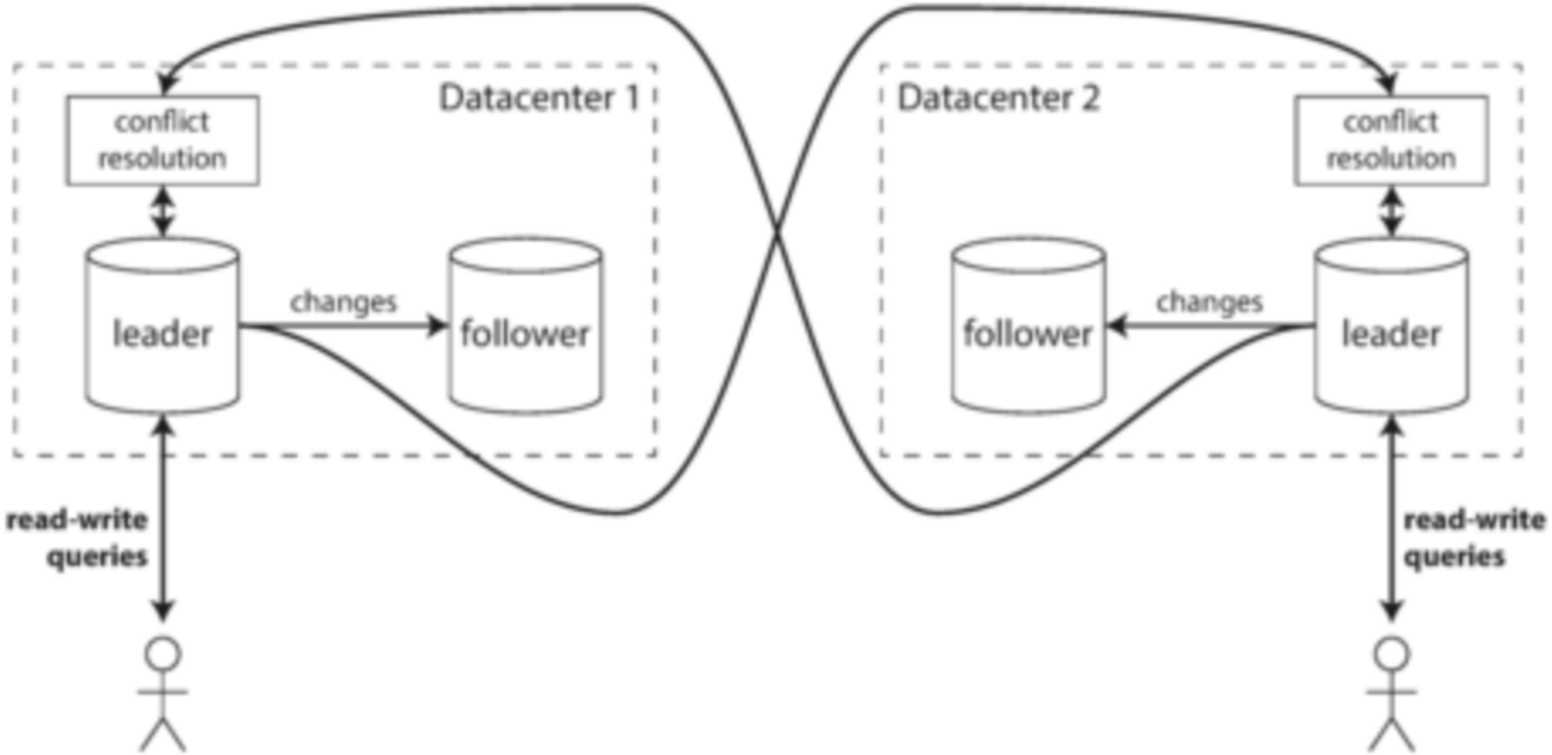


MultiLeader Replication

- Las aproximaciones con un solo líder, representa que si el líder no esta disponible, se presenta un único punto de falla.
    - Igualmente todas las operaciones de escritura, pasan por el líder.
    - Así, si por cualquier razón el líder no esta disponible, no se puede efectuar la operación de escritura.
- ¿Cómo lo solucionamos?, la forma más expedita es agregar otra instancia que permita las operaciones de escritura.
- También denominado replicación Master/Master o Active - Active.

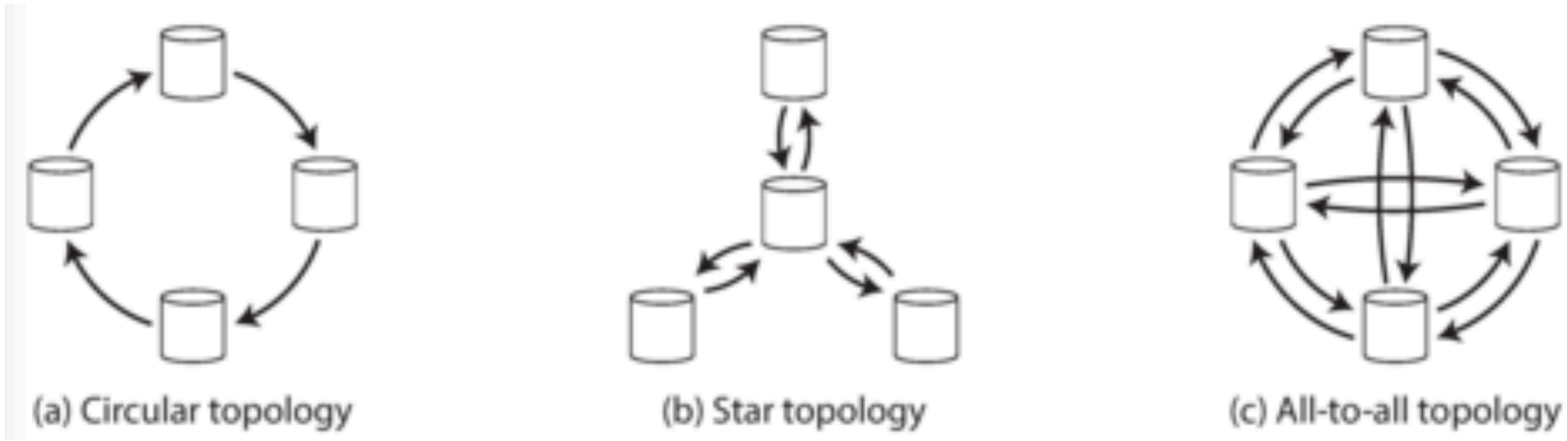
Casos de Uso para Replicación MultiLeader

Operación en Múltiples Data Center



- ¿Qué pasa si un mismo dato es actualizado por dos clientes diferentes a través de dos líderes diferentes?

Topologías de Configuración para MultiLeader



Leaderless

- En esta aproximación no se define un maestro o líder dentro de la arquitectura para la replicación.
- Esto implica que los clientes pueden realizar las operaciones de escritura a múltiples nodos.
- De igual forma pueden realizar la operación de lectura de múltiples nodos en paralelo.

Teorema CAP

CAP

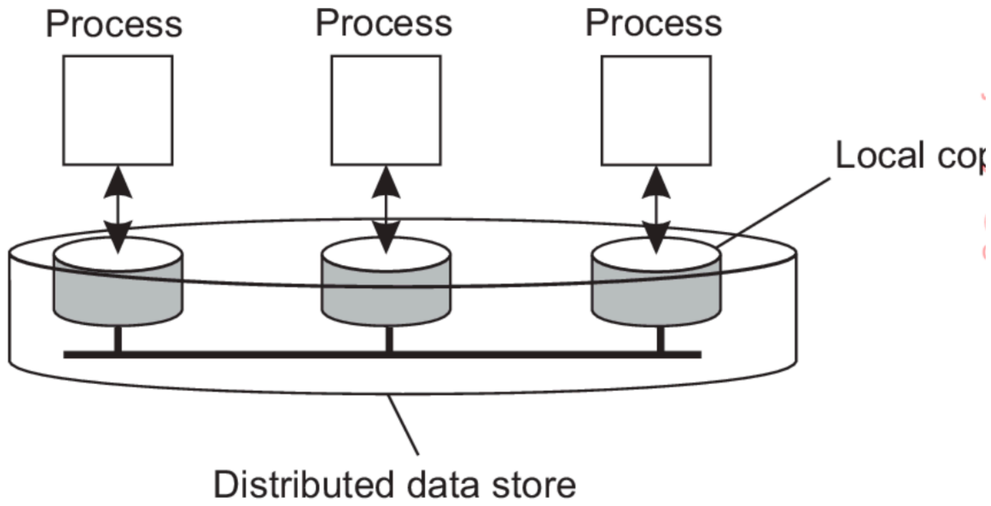
- Consistency : Esta relacionada con la característica de que todos los clientes ven los mismos datos al mismo tiempo. Todo esto independiente del nodo al cual se conecten.
  - Para que esto suceda, siempre se debe garantizar que cada vez que se escriban datos en un nodo, se deben enviar a todas las réplicas.
- Availability: Esta relacionada con la disponibilidad que se debe proporcionar de los datos. De esta forma, cada vez que un cliente emite una consulta, se debe emitir una respuesta, independiente si no todos los nodos están activos.
- Partition Tolerance: Esta relacionada con la falla en las comunicaciones en un sistema distribuido, bien sea por la perdida de la conexión o latencia en las comunicaciones.
  - Esto implica que el cluster de nodos debe continuar trabajando en presencia de posibles fallas entre la comunicación de ellos.

Consistency Models

- ¿Qué sucede si múltiples clientes leen o modifican diferentes copias de datos simultáneamente o dentro de un período corto de tiempo?

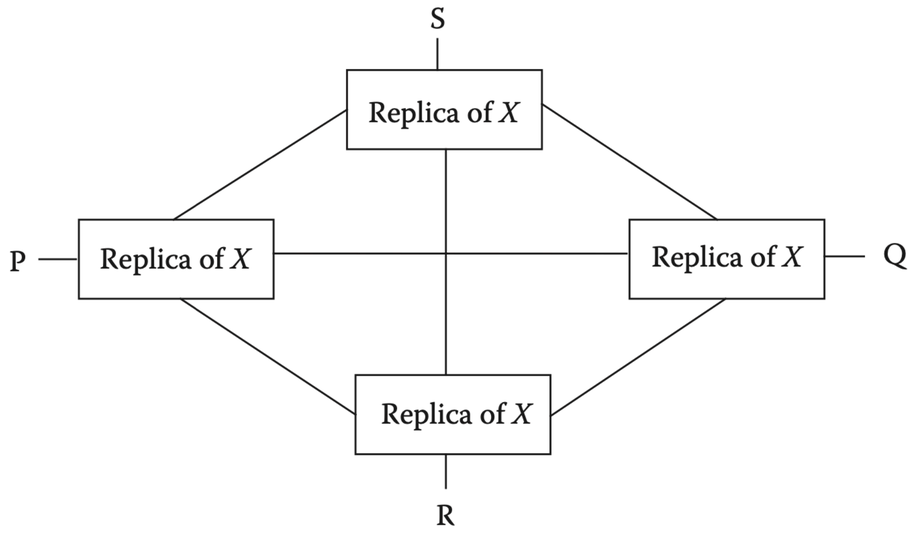
Consistency Models

- Un modelo de consistencia se puede entender como un contrato entre los procesos y el data store.
- De esta forma, se puede afirmar que si los procesos se comprometen a cumplir ciertas reglas, el data store se compromete a trabajar de forma correcta.
- Normalmente, los procesos realizan operaciones de lectura y escritura. Estos esperan que el data store, le entregue la última versión del dato.
- Ahora, en ausencia de un reloj global, realmente se vuelve complejo determinar cuál es la ultima operación de escritura que se realizo sobre un dato.



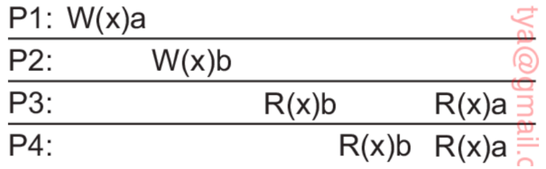
Consistency Models

- Strict Consistency
  - Este modelo corresponde a la verdadera transparencia en el proceso de realización.
  - Por ejemplo, si alguno de los procesos ejecuta una operación de escritura sobre la variable x=5 en el tiempo “t” y si esta es la última operación de escritura sobre dicha variable, entonces a un tiempo t’ > t cualquier proceso que efectúe una operación de lectura sobre la variable x debe recibir el valor de 5.

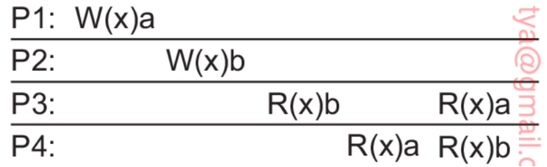


Consistency Models

- Sequential Consistency:
  - Es un modelo importante de consistencia.
  - Propuesto por Leslie-Lamport.
  - “El resultado de cualquier operación es el mismo como si las operaciones de lectura y escritura de todos los procesos en el data store se ejecutaran en algún orden secuencial y las operaciones de cada proceso individual aparecieran en esta secuencia en el orden especifico por su programa”



(a)



(b)