

Sistemas Operativos

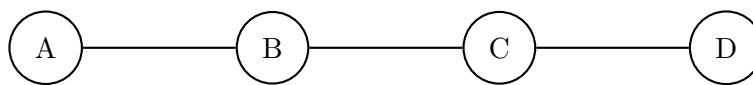
Práctica 7: Sistemas distribuidos

Notas preliminares

- Los ejercicios marcados con el símbolo ★ constituyen un subconjunto mínimo de ejercitación. Sin embargo, aconsejamos fuertemente hacer todos los ejercicios.

Ejercicio 1 ★

Un sistema distribuido tiene cuatro nodos: A, B, C y D, conectados de la siguiente forma:



Responda las siguientes preguntas justificando:

a) Dados los siguientes escenarios:

- El nodo B se cae
- El enlace entre A y B se cae.
- B está muy sobrecargado, y su tiempo de respuesta es 100 veces mayor a lo normal.

¿Puede A discernir entre cada uno de ellos?

- b) Si A recibe un mensaje de D, a través de B, ¿Se puede asumir que D no está caído?.
- c) Si B recibe un mensaje de A y uno de C, ¿Se puede saber si A envió su mensaje antes que C, o viceversa? ¿Por qué?

Ejercicio 2 ★

Suponer un sistema donde los n procesadores están comunicados mediante un bus ultra rápido de baja latencia, de manera tal que los tiempos de acceso a memoria remota son comparables con los tiempos locales. Imaginar que se cuenta además con un entorno de programación que permite manejar la memoria remota como si fuera local. ¿Consideraría a tal sistema como *distribuido* o *paralelo*? Justificar.

Ejercicio 3 ★

Un algoritmo de *commit* distribuido funciona de la siguiente manera: opera sobre una red donde los paquetes pueden perderse o demorarse, y cuando un nodo quiere escribir sobre cierto archivo que está replicado en todos los nodos, envía un pedido de escritura. Si recibe confirmación de todos los nodos, escribe y le notifica a los demás que pueden hacer lo propio. Alguien nota que este algoritmo puede fallar si los nodos se caen entre la escritura y la notificación, y propone para solucionarlo el envío de mensajes de confirmación de parte de los nodos. ¿Este algoritmo modificado resuelve el problema? Justificar.

Ejercicio 4 ★

Se tiene un sistema distribuido donde los nodos están conectados bajo una topología de anillo, cada uno con un ID que los identifica. Se quiere implementar un algoritmo de elección de líder donde aquel elegido sea el que tenga el menor ID.

- Proponga un protocolo para resolver este problema. Analice la complejidad de su protocolo con respecto a la cantidad de mensajes que se utilizan.
- Si su solución tuvo $O(n^2)$ cantidad de mensajes utilizados, proponga otro protocolo que disminuya esa cantidad.

Ayuda: Considere un protocolo por fases y en elegir líderes en un vecindario local de cada nodo.

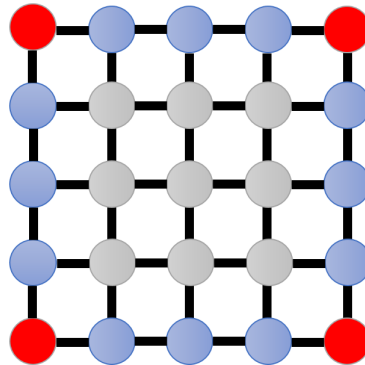
Ejercicio 5 ★

Se tiene replicas de una base de datos en m nodos distintos, cada uno con su propio identificador. Estos nodos estan distribuidos en grupos en distintas partes del mundo. Todos los nodos estan conectados entre si y tienen un líder para poder organizarse. Supongamos que falla el líder

- Proponga un algoritmo para elección de un nuevo líder, basándose en el menor ID.
- Suponga que uno de los grupos queda desconectado de todos los demás, formandose una partición de la red. Después de un cierto tiempo, el grupo se puede volverse a conectar. Proponga un protocolo de elección de líder que contemple esta situación. Cada partición que se provoque tiene que elegir un nuevo líder y seguir funcionando bajo sus órdenes. En caso que se vuelvan a unir las particiones, deberán buscar un nuevo líder.

Ejercicio 6

Se tiene un sistema distribuido conectado con una topología en malla. Proponga algún protocolo de elección de líder bajo esta topología.



Ejercicio 7 ★

¿Qué es 2PC? ¿Qué problema resuelve? ¿Qué fallas tiene? ¿Hay algún protocolo que resuelva alguna de esas fallas?

Ejercicio 8

Un servidor usa ordenamiento por timestamps para el control de la concurrencia entre sistemas distribuidos. Este servidor funciona para que una app bancaria permita realizar operaciones de manera remota. Entre otras cosas, comprar y vender diferentes tipos de divisas. Un usuario puede comprar cualquier tipo de moneda dentro del sistema con cualquiera que posea en su cuenta. El banco cuenta con un sistema que recibe los pedidos, que a su vez se puede comunicar con otro sistema que controla el stock de los diferentes tipos de divisas y un tercer sistema que mantiene registro del estado de cuenta de los usuarios.

- Describir un protocolo que permita a los usuarios comprar divisas manteniendo en todo momento la consistencia, aprovechando el uso de los timestamps.
- Una persona nos planteó el uso del protocolo 2PC para resolver este problema. Sin embargo, alguien nos argumentó que el uso de este protocolo es redundante bajo estas condiciones. ¿Estaría de acuerdo con esta persona? Justifique

Ejercicio 9

Relacione el mecanismo de consenso de Blockchain con el consenso requerido para resolver el problema de consenso bizantino. Justifique su respuesta en cada caso.

- a) ¿Es el mismo tipo de consenso el que requieren ambos ámbitos? ¿Por qué? ¿Cuáles son las similitudes y/o diferencias?
- b) Si es el mismo tipo, ¿Blockchain resuelve el problema bizantino? Si son diferentes. ¿Existe alguna modificación a la estrategia planteada en Blockchain para que se adapte al problema bizantino?

Referencias:

- Paper de Lamport: <https://www.microsoft.com/en-us/research/uploads/prod/2016/12/The-Byzantine-Generals-Problem.pdf>
- Paper Bitcoin original: <https://bitcoin.org/bitcoin.pdf>
- Nota explicativa cointelegraph: <https://cointelegraph.com/blockchain-for-beginners/how-does-blockchain-solve-the-byzantine-generals-problem>