Laboratorio de Análisis y Diseño de Algoritmos Distribuidos en Redes

2024

Consenso y Tolerancia a Fallos en Sistemas Distribuidos

Objetivo

En esta asignación de laboratorio, los estudiantes implementarán y analizarán algoritmos distribuidos que se centran en lograr consenso de manera tolerante a fallos. El objetivo es entender los desafíos y técnicas utilizadas en sistemas distribuidos para asegurar el acuerdo entre múltiples nodos, incluso en presencia de fallos.

Tarea

Entorno:

- Implemente los algoritmos diseñados en el simulador PyDistSim.

Parte 1: Introducción a los Algoritmos de Consenso

- Elección de Líder:
    - Implemente un protocolo de elección de líder donde un nodo entre un grupo de nodos es elegido como líder.
    - Cada nodo intentará convertirse en líder transmitiendo su ID a los demás. El nodo con el ID más alto es elegido.
    - Simule el proceso.
- Replicación de Registros:
    - Amplíe su implementación para incluir replicación de *registros* del tipo <*clave, valor*>.
    - El líder propondrá registros para replicarlos en los nodos seguidores.
    - Los seguidores deben reconocer la recepción de las entradas de registros, y el líder las confirma solo después de que la mayoría haya reconocido.
- Manejo de Fallos:
    - Introduzca fallos en los nodos (líder o seguidores).
    - Implemente mecanismos de recuperación para manejar fallos de nodos, asegurando que el sistema aún pueda funcionar y alcanzar consenso incluso con fallos en los nodos.

Parte 2: Tolerancia a Fallos y Análisis de Rendimiento

- Pruebas de Tolerancia a Fallos:
    - Simule escenarios de fallos donde los nodos fallan o se vuelven inactivos aleatoriamente.
    - Mida el tiempo necesario para recuperarse de fallos y continuar con la replicación de registros.
    - Asegúrese de que el sistema pueda aún lograr consenso en presencia de fallos.
- Análisis de Rendimiento:
    - Analice el rendimiento de tu implementación bajo diferentes condiciones de red (por ejemplo, variación en latencia, pérdida de mensajes, etc.).
    - Analice el tiempo necesario para alcanzar el consenso a medida que aumenta el número de nodos.
    - Compare el tiempo necesario con y sin fallos de nodos.

Parte Adicional: Tolerancia a Fallos Bizantinos (Opcional)

- Implementar Tolerancia a Fallos Bizantinos:
    - Modifique su algoritmo de consenso para tolerar nodos *bizantinos* (donde los nodos pueden actuar de forma maliciosa o enviar datos incorrectos), asegurando que el sistema aún pueda alcanzar consenso incluso cuando algunos nodos se comportan de forma arbitraria.
- Análisis:
    - Analice cómo los nodos bizantinos afectan el proceso de consenso.
    - Proporcione una comparación de rendimiento entre el algoritmo tolerante a fallos tradicional y la versión tolerante a fallos bizantinos.

Entregas

Deberan entregar:

- Un archivo PDF con un informe que incluya:
    - el diseño de cada uno de los algoritmos solicitados
    - los análisis requeridos en cada caso y
    - descripciones y resultados de los test realizados.
- Código fuente de todas las implementaciones.
- Un archivo README con instrucciones sobre cómo ejecutar tu código.

Fechas

El plazo para la entrega del trabajo: 1/12/2024

Recursos

1. Paxos Made Simple: http://lamport.azurewebsites.net/pubs/paxos-simple.pdf
2. Raft Consensus Algorithm: https://raft.github.io/raft.pdf
3. Raft Visualization Tool: http://thesecretlivesofdata.com/raft/
4. Practical Byzantine Fault Tolerance: https://pmg.csail.mit.edu/papers/osdi99.pdf