Coordinación y consenso

Sistemas Distribuidos

Objetivos

- Explicar el concepto de coordinación
- Revisar y explicar las consideraciones de fallos
- Explicar el concepto de exclusión mutua distribuida
- Revisar dos algoritmos de exclusión mutua

Conceptos

En esta unidad se abordarán temas y algoritmos que comparten un fin común en los sistemas distribuidos, el cual es:

> *"dado un conjunto de proceso, coordinar sus acciones o ponerse de acuerdo en uno o más valores compartidos aún en presencia de fallos"*

Conceptos

Los procesos en un Sistema Distribuido necesita:

- Coordinar sus acciones
- Ponerse de acuerdo en la actualización de valores.

Formas de coordinación y consenso

- Acceso a recursos: exclusión mutua distribuida
- Selección de coordinadores: algoritmos de elección
- Toma de decisiones: condiciones para lograr el consenso

Conceptos

Se debe tener en cuenta la presencia de fallos:

- En un sistema distribuido pueden fallar tanto los procesos como los canales de comunicación; esto es, pueden apartarse de lo que se considera el comportamiento correcto y deseable.

Fallos

Consideraciones sobre los fallos

- Los fallos pueden ser por omisión o arbitrarios.
  - **Omisión**: los procesos o canales no realizan las acciones que se supone deben hacer.
  - **Arbitrarios** (bizantinos): la peor semántica de fallo posible, en la que puede ocurrir cualquier tipo de error.

- Ej. un proceso carga valores equivocados, responder con un valor incorrecto, contenido corrompido, repartir mensajes inexistentes, entrega de mensajes duplicados, etc.

Fallos

Consideraciones sobre los fallos

- El fallo de un proceso no significa que el resto del sistema no puedan comunicarse: propiedad de tolerancia a fallos

- Un **proceso correcto** es uno que no muestra fallos en ningún momento de la ejecución que se esté considerando.

Fallos

Detección de fallos

- Un **detector de fallos** es un servicio al que recurren para saber si un proceso determinado ha fallado.

  - Detector de fallo **no fiable**: no es necesariamente exacto (sospechoso / no sospechoso) : manejo de probabilidades

  - Detector de fallo **fiable**: detecta de forma exacta (fallido / correcto)

Fallos

Detección de fallos

- **Detección inexacta**: sospechar de un proceso que no ha fallado.
- **Detección incompleta**: no sospechar de un proceso que ha fallado.

El término «**sospechar**» es muy subjetivo.

- Problema: ¿Cuándo un proceso realmente ha fallado?

Fallos

Algoritmo para la detección de fallos no fiable

- Cada proceso *p* envía " *p* está aquí" al resto de los procesos y lo hace cada *T* unidades de tiempo.
- El detector de fallos utiliza una estimación del tiempo máximo de transmisión de un mensaje de *D* segundos.
- Si el detector de fallos local en otro proceso *q* no recibe un mensaje " p está aquí" dentro de los *T+D* segundos desde el último, entonces:
  - informa a *q* que *p* es *Sospechoso*.

Fallos

Detección de fallos no fiable hacia uno fiable

- Al ser una estimación, las respuestas del detector no fiable son consideradas como indicios (sospechas), con una probabilidad de que sea exacto.

- Para hacerlo fiable se debe elegir D de tal forma que no sea una estimación sino un límite absoluto en los tiempos de transmisión del mensaje. *D = 3±1 segundos*
  - Cada T=5 segundos más D : 2 - 4 segundos llegará un mensaje de "p está aquí". El mensaje se deberá recibir entre 7-9 segundos
  - La ausencia de un mensaje «p está aquí» dentro de T+D segundos lleva al detector a concluir que p se ha caído (es fallido).

Fallos

Elección de T

- Demasiado pequeño, sobrecargará la red
- Demasiado grande, detectará procesos sospechosos con lentitud

Detector de fallos en un sistema distribuido práctico y real

- En la mayoría de los sistemas distribuidos se tendrán estimaciones en los tiempos máximos de transmisión.
- Se debe detectar siempre el fallo, aunque no sea de manera perfecta.
- Detectores con propiedades bien definidas pueden aportar soluciones prácticas.

Fallos

Posible propuesta conocer el valor de D:

- Elección de tiempos máximos de transmisión **adaptativos** mediante un **aprendizaje supervisado** que utilice las variables: tráfico, retraso observados en la red, horario, medio de transmisión, etc.

- La idea es aprender el tiempo estimado, dado un conjunto con valores observados de las características, y que el detector **calcule** cuál es el tiempo máximo de transmisión para nuevos valores de las características.

Exclusión mutua distribuida

- **Sección crítica (SC)**: porción de código que permite el acceso a un recurso compartido por varios procesos

- **Exclusión mutua distribuida**: el acceso a la sección crítica se basa en paso de mensajes.

«Si una colección de procesos comparten recursos, entonces es necesaria una exclusión mutua para prevenir interferencias y asegurar la consistencia »

Algoritmos para la exclusión mutua

Protocolo a nivel de aplicación

- entrarSC() // entre a la sección crítica, bloqueo del proceso si SC ocupada
- accesoRecursos() // acceso a los recursos compartidos en la sección crítica.
- salirSC() // salida de la sección crítica, pueden entrar otros procesos.

Algoritmos para la exclusión mutua

Requisitos

- EM1: Seguridad
  - A lo sumo un proceso puede estar ejecutándose a la vez en la SC
- EM2: Pervivencia
  - Las peticiones de entrada/salida de la SC al final son concedidas sin interbloqueos.
- EM3: Ordenación
  - Si una petición para entrar en la SC ocurrió “antes que” otra, entonces la entrada en la SC se garantiza en ese orden

Algoritmos para la exclusión mutua

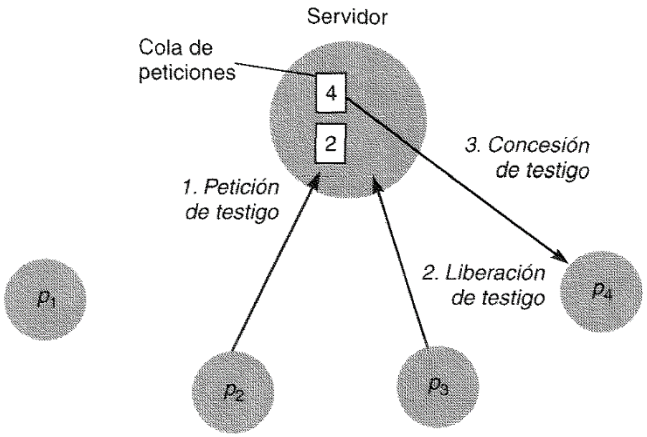
Criterios de evaluación del rendimiento de los algoritmos

- Ancho de banda consumido
  - Proporcional al número de mensajes enviados en cada operación de entrada y salida de la SC
- Retraso del cliente
  - Cuánto tarda en entrar y salir de la SC
- Retraso en la sincronización
  - Tiempo que pasa entre que un cliente sale de la SC y el siguiente que esté esperando entra.

Algoritmos para la exclusión mutua

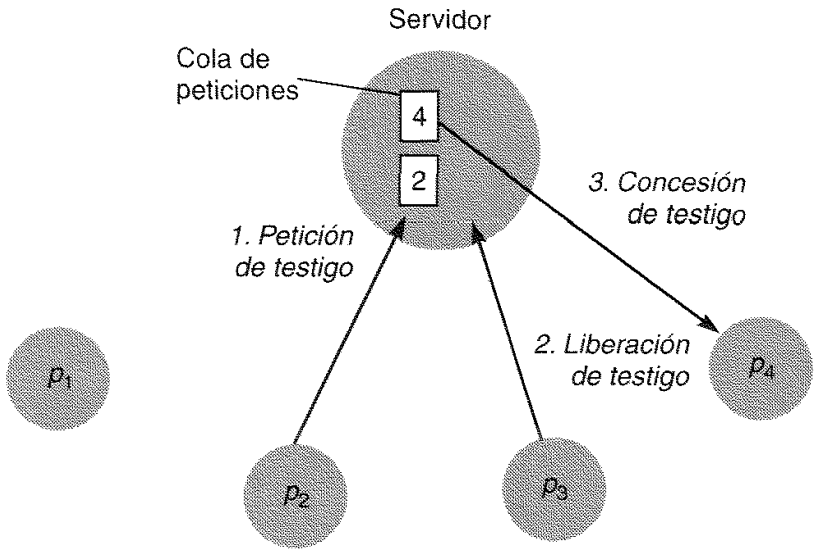
Algoritmo del servidor central

- Se emplea un servidor que da los permisos para entrar a la S.C mediante un testigo (*token*) que se envía por mensajes, el cual significa permiso.
- Si otro proceso tiene al testigo, el proceso solicitante se envía a una cola de peticiones.
- La atención en la cola se escoge mediante antigüedad.



Algoritmos para la exclusión mutua

Algoritmo del servidor central



- El P_3 está en la sección crítica.
- P_4 ya está en la cola de peticiones:
  1. P_2 solicita acceso y es puesto en la cola después de P_4 .
  2. P_3 librea el testigo
  3. Se concede el testigo al proceso más antiguo de la cola (P_4).

Algoritmos para la exclusión mutua

Evaluación del rendimiento del algoritmo del servidor central

- La entrada a la sección crítica conlleva 2 mensajes de entrada, (una petición y una concesión) y 1 mensaje de salida: una liberación.

- Retraso al proceso solicitante por la duración de este mensaje de ida y vuelta.

- El servidor puede convertirse en un cuello de botella para el rendimiento del sistema total.

Ejercicio

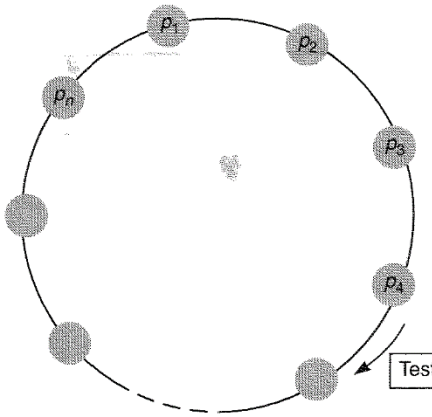
Realizar una corrida del algoritmo de servidor central considerando 5 procesos. Indicando el orden de atención y el número total de mensajes enviados.

- El testigo está libre
- P1 y P2 llegan solicitan testigo en ese orden.
- P3 y P4 llegan , en ese orden, mientras P2 tiene el testigo.
- P5 llega cuando P3 tiene el testigo.

Algoritmos para la exclusión mutua

Algoritmo basado en anillo

- Cada proceso tiene un canal de comunicación hacia el siguiente proceso en el anillo.
- *Testigo* circulando en una dirección.
- Si un proceso requiere entrar a la SC, espera el testigo y lo retiene y cuando sale de SC enviará el testigo hacia el vecino.
- Si no requiere entrar a SC, inmediatamente lo hace avanzar hacia su vecino.



Algoritmos para la exclusión mutua

Evaluación del rendimiento del algoritmo basado en anillo

- Consumo continuo de ancho de banda salvo si el proceso está en la SC.
- Los procesos siempre están enviando el testigo, aun cuando ningún proceso requiera entrar a SC.

Recapitulando…

- Explicar el concepto de coordinación
- Tipos de fallos
- Explicar “detector de fallos”, detector de fallos fiable, detector de fallos no fiable.
- Detección inexacta, detección incompleta.
- Explicar el algoritmo para la detección de fallos no fiable.
- Cómo seleccionar el valor de T más adecuado.
- Explicar “exclusión mutua distribuida”, “sección critica”.
- Tres requisitos para la exclusión mutua.
- Explicar algoritmo de servidor central
- Explicar algoritmo basado en anillo