RC: Actividades Tema 1

ACTIVIDAD 1

Imagine que en una asignatura de programación se le pide que escriba un programa que implante una calculadora científica. Por falta de tiempo, de momento se ha escrito un programa que es capaz de manejar únicamente números enteros y los operadores '=' (para mostrar el resultado), '+' (suma), '*' (multiplicación), '-' (resta), '/' (división) y '%' (módulo). No se comprueba si el valor facilitado en los argumentos es coherente con la operación solicitada. Posteriormente se extiende ligeramente el programa para que también admita números reales como argumentos, pero se sigue sin comprobar la coherencia entre operadores y argumentos. Ahora tenemos un programa que todavía no está depurado y que es claramente incompleto. ¿Qué cree que ocurrirá si algún usuario utiliza esta versión del programa y realiza las operaciones que se listan a continuación? Justifique si en cada situación se generará: (i) un resultado correcto, (ii) un defecto, (iii) un error, (iv) un fallo.

a) El usuario introduce la siguiente secuencia: 456 + 23 =

Un resultado correcto.

b) El usuario introduce la siguiente secuencia: 345 / 0 =

Es un defecto que se manifestará como un error o un fallo.

c) El usuario introduce la siguiente secuencia: 345.23 % 2.0 =

Es un defecto de programación que producirá un error dado que la calculadora no acepta reales.

A partir de los ejemplos que acaba de analizar, justifique si el diseño y desarrollo cuidadoso de un programa puede reducir las situaciones que conduzcan a la aparición de defectos.

Una cuidadosa programación si que aumentará la robustez de la aplicación, pero se debería de tener en cuenta que pueden aparecer defectos que no están previstos.

ACTIVIDAD 2

El modelo de replicación pasivo fue especificado inicialmente bajo la exigencia de que en todo momento solo haya una réplica primaria y que todas las operaciones sean ejecutadas únicamente por esa réplica. Las modificaciones generadas durante la ejecución de una operación deben ser propagadas posteriormente a las réplicas secundarias.

Según esos requisitos, el número mínimo de réplicas que deberá tener un proceso replicado bajo el modelo pasivo depende del número de fallos simultáneos que pueda haber (que representaremos mediante la variable "f") y del modelo de fallos que se asuma. Asumiremos una red dedicada con topología completa (existe un enlace de comunicación entre cada par de procesos) y que cada proceso se ubica en una máquina distinta. El término "enlace" se referirá a una vía de comunicación entre dos procesos, considerando también el encaminamiento necesario.

Por ejemplo, si asumimos el modelo de fallos de "caída y enlace" el número mínimo de réplicas para soportar "f" fallos simultáneos es "f+2". Para demostrarlo, asumamos que solo existieran f+1 réplicas y observaremos cómo se incumple el requisito de tener un solo primario en el sistema:

- Esas f réplicas podrían dividirse en dos conjuntos: A (formado por una sola réplica) y B (formado por las f restantes). Consideremos dos situaciones:
    - La única réplica de A cae (es decir, falla y permanece parada a partir de entonces). Entonces se elegirá un primario de entre las réplicas de B.
    - Todas las réplicas de B caen. Entonces se elegirá la única réplica de A como primario.
- Podría llegar a darse una tercera situación: Todos los enlaces que unen a la única réplica de A con cada una de las réplicas de B perdieran todos los mensajes enviados a través de ellos. Hay "f" enlaces, por lo que esto también sería una situación con "f" fallos simultáneos. En este caso, la réplica de A creería que todas las réplicas de B han fallado, por lo que esa réplica adoptaría el rol primario. A su vez, todas las réplicas de B también creerían que la única réplica de A ha fallado. Por ello, una de las réplicas de B sería elegida como primario. Como resultado, tendríamos dos primarios y esta situación no sería admisible. Resulta necesario tener al menos una réplica más.

Analice cuántas réplicas se necesita tener como mínimo para soportar "f" fallos simultáneos bajo los siguientes modelos de fallos:

1. Parada.

Con disponer de f+1 replicas sería suficiente para que el sistema siga funcionando, dado que si fallan f nodos, quedaría uno que actuaría de nodo primario.

2. Omisión de recepciones.

Se necesitaría disponer de f+2 nodos para que al menos dós replicas más para que se comuniquen con el nodo principal y evitar tener dos nodos principales. Esta situación se explica también en el enunciado.

3. Justifique por qué bajo este modelo pasivo no se podrá soportar el modelo de fallos arbitrarios.

Porque solo hay un nodo que envía la información (Nodo primario) y por lo tanto no se comprueba si la respuesta es valida o es un error dada la confianza que tiene los nodos secundarios.

ACTIVIDAD 3

La actividad 2 justifica que cuanto más severo es un modelo de fallos (los fallos bizantinos son el modelo más severo posible) mayor dificultad existe para el desarrollador de aplicaciones robustas. Como consecuencia, parece sensato que se asuma un modelo de fallos lo más relajado posible (por ejemplo, el modelo de parada). Eso facilitaría las tareas de los responsables del diseño y desarrollo de aplicaciones. Sin embargo, el sistema subyacente también necesitará realizar cierto esfuerzo para asegurar que las situaciones de fallo reales "cuadren" con el modelo asumido. En este caso, para quien deba desarrollar el sistema operativo distribuido y/o el middleware, resultaría mucho más cómodo proporcionar un modelo de fallos de omisión general que un modelo de fallos de parada.

Describa alguna situación en la que resulte difícil proporcionar una imagen coherente con el modelo de fallos de parada. Esto es, se pide describir un escenario en el que haya un fallo en uno o más componentes y, para los demás componentes correctos, eso no "cuadre" con la parada de los componentes defectuosos.

Si estamos en el caso donde la conexión entre dos nodos (que están conectados directamente) tienen un retraso importante a la hora de comunicarse o se genera un fallo por omisión y la razón de fallo es causado por el medio de comunicación, sería difícil aplicar un modelo de parada. En este caso se sospecharía de los dos nodos, en lugar de una solo y por lo tanto se debería de estudiar más en detalle el caso, dado que resultará difícil saber si se tiene que hacer una parada y en cual de los dos nodos se debería de realizar.

ACTIVIDAD 4

Los servicios robustos o confiables (del inglés “dependable”) deben ser fiables, disponibles, seguros (“safe” y “secure”) y mantenibles. Como hemos visto, la “mantenibilidad” se refiere a la capacidad de recuperación rápida.

Imagine que cierto servicio distribuido se ha replicado para asegurar su fiabilidad y disponibilidad. Con ello se espera que pueda soportar las situaciones de fallo. Como ya habrá podido observar, garantizar la disponibilidad también implicará una buena “mantenibilidad” pues cuando falle alguna réplica ésta deberá volver lo antes posible a proporcionar servicio.

Asuma que este servicio se ha replicado utilizando el modelo pasivo (que se explicará con detenimiento en otras unidades de esta misma asignatura pero que ya ha sido introducido en la Actividad 2). Explique qué secuencia de acciones debería llevarse a cabo para recuperar una réplica (primaria o secundaria) bajo este modelo de replicación. Preste especial atención a los mecanismos de detección de fallos (que deberá describir) y a las acciones necesarias para transferir aquel estado “perdido” durante el intervalo de fallo. Justifique si el modelo de fallos que se haya podido asumir influirá en el procedimiento a utilizar para recuperar una réplica.

Si estamos con un modelo pasivo donde todos los fallos se pueden reducir a fallos de parada menos los fallos bizantino, el mecanismo que se va a utilizar para detectar un fallo será mediante la sospecha.

En el caso en el que se sospecha de una replica primaria, esta se debería de reiniciar y sustituir por una de las replica secundarias más actualizada y el resto de replicas deberían de actualizar su información sobre cual es la nueva máquina primaria. La máquina al reiniciarse se incorporará como una máquina secundaria. Los pasos de como se va a incorporar serán los mismos que para las máquinas secundarias que se explicarán a continuación.

En el caso de que se sospeche de una máquina secundaria, esta se debería de reiniciar y a la hora de realizarse la incorporación, primero se debería estudiar el estado de la máquina que puede ser de amnesia, amnesia parcial, pausa o parada.

En el caso de que la máquina se reinicia en un estado de amnesia, significa que no depende de las peticiones que atendió antes de fallar y se reinicia con un estado preestablecido.

En el caso de que la máquina se reinicia en un estado de amnesia parcial, significa que la máquina recupera parte de la información que tenía antes de fallar, pero también tiene unos valores preestablecidos.

En el caso de que la máquina se reinicia en un estado de pausa, significa que la máquina es capaz de recuperar toda la información que tenía antes de fallar.

En el caso de que la máquina se reinicia en un estado de parada, significa que ya no es posible reiniciarla y permanecerá parada.

En el caso de que el estado de la máquina al reiniciarse es de parada, primero se iniciará una nueva máquina con valores preestablecidos para sustituir esa máquina si es posible.

A continuación la máquina recibirá la información sobre el conjunto de máquinas secundarias y a su vez el resto de máquinas secundarías actualizarán su información. Este ultimo paso se realiza por todas las máquinas indiferente del estado que tenga.

ACTIVIDAD 5

Imagine un determinado servicio distribuido en el que se utilizan servidores replicados para garantizar su disponibilidad. ¿Cómo afectará a dicha disponibilidad una situación de particionado de la red? ¿Se podrá seguir garantizando? En caso de que no se pueda, justifique por qué. En caso de que se pueda, explique si tendrá algún efecto sobre (la calidad de) el servicio proporcionado.

Si estamos en una situación de particionado de red, existen dos posibilidades de resolver este problema para garantizar la máxima disponibilidad.

La primera posibilidad es que la aplicación que se ha desarrollado permita trabajar son el sistema particionado.
Eso puede generar problemas en cuanto a la consistencia de los datos si no se trata de comprobar y sincronizar la información que contiene cada una de las replicas. Lo que se garantiza con este sistema es que se tendrá una alta disponibilidad.

La segunda posibilidad es utilizar solo aquel subgrupo que tenga al menos más de la mitad de los procesos preconfigurados. De esta manera se mantiene la consistencia de manera más fácil, pero se reduce la disponibilidad.

ACTIVIDAD 6

Justifique qué dependencias llega a haber entre los cinco factores de la confiabilidad/robustez de un servicio distribuido: fiabilidad, disponibilidad, mantenibilidad, “safety” y “security”. Por ejemplo... ¿Puede haber un sistema fiable no disponible? ¿Se puede tener una buena mantenibilidad y una disponibilidad baja? ¿Puede un sistema inseguro (“unsafe”) ser fiable o ser altamente disponible?

Fiabilidad (“reliability”) y seguridad (“safety”):

Si se tiene una mayor seguridad, entonces la probabilidad de que un sistema no falle de manera catastrófica durante un intervalo de tiempo aumenta y eso incrementa a su vez la fiabilidad.

Fiabilidad (“reliability”) y disponibilidad (“availability”):

Si se tiene una mayor disponibilidad, entonces la fiabilidad del sistema aumenta ya que hay mayores probabilidades de que el sistema se funcional en un instante t.

Fiabilidad (“reliability”) y mantenibilidad (“maintainability”):

Si se tiene una mayor fiabilidad del sistema, entonces la mantenibilidad será alta también ya que no se deberán de realizar un numero elevado de recuperaciones para mantener el sistema en funcionamiento. En caso de el que sistema falle y se realiza la recuperación, a mayor mantenibilidad mayor fiabilidad habrá en el sistema ya que se recuperará de manera más rápida.

Fiabilidad (“reliability”) y seguridad (“security”):

Si se tiene una mayor seguridad, la fiabilidad del sistema aumentará ya que habrá menos o preferiblemente ninguna intrusión en el sistema que afecte la información o el sistema.

Seguridad (“safety”) y disponibilidad (“availability”):

Si se tiene una mayor seguridad, la posibilidad de que el sistema falle de manera catastrófica se reducen y eso aumenta la disponibilidad del sistema.

Seguridad (“safety”) y mantenibilidad (“maintainability”):

Si se tiene una mayor seguridad, la posibilidad de que el sistema falle de manera catastrófica se reducen y eso a su vez disminuye la necesidad de realizar el mantenimiento. En caso de que se tenga que realizar un mantenimiento, este no afecta la seguridad del sistema.

Seguridad (“safety”) y seguridad (“security”):

Si se tiene una mayor seguridad (“security”), se disminuye la posibilidad de que haya una intrusión en el sistema y eso aumenta la seguridad (“safety”) del sistema, dado que habrán menos posibilidades de que se produzca un fallo catastrófico por culpa de la intrusión.

Disponibilidad (“availability”) y mantenibilidad (“maintainability”):

Si se tiene mayor disponibilidad, la posibilidad de que el sistema falle y se tenga que recuperar disminuye. En cambio si se tiene que realizar realizar una recuperación, está afecta la disponibilidad del sistema y por lo tanto cuanto menor es el tiempo de mantenibilidad, mayor será la disponibilidad.

Disponibilidad (“availability”) y seguridad (“security”):

Si se tiene mayor seguridad, aumentará la disponibilidad del sistema dado que al no tener menos o ningún intrusos en el sistema, menor es la posibilidad de tener que realizar una parada del sistema.

Mantenibilidad (“maintainability”) y seguridad (“security”):

Si se tiene mayor seguridad, se disminuye la necesidad de realizar paradas para recuperar el sistema en caso de una intrusión no deseada.