Universidad de Guadalajara

Centro Universitario de Ciencias Exactas e Ingenierías

Computación Tolerante a Fallas

Maestro: Michel Emanuel Lopez Franco

Juan Antonio Perez Juarez
Carrera: INCO
Código: 215660996

Ejercicio 01 Conceptos básicos.

Objetivo:

Conocer los conceptos básicos en sistemas tolerantes a fallas.

Desarrollo:

Contesta las siguientes preguntas:

¿Qué son los sistemas tolerantes a fallas?

En informática, la tolerancia a fallos o conmutación por error (en inglés: failover) se refiere a la capacidad de un sistema de seguir funcionando, aún en caso de producirse algún fallo en el sistema. Observar que los fallos pueden ser no intencionados (por ejemplo, caídas de sistemas, fallos en el cableado, fallo hardware) o intencionados por alguna parte no confiable del sistema.

El nivel de tolerancia a fallos dependerá de las técnicas utilizadas para conseguirlo. No obstante, nunca será absoluta ya que siempre hay algún tipo de fallo masivo que produciría un error irrecuperable. Cada sistema hay que diseñarlo (diseño de tolerancia a fallos) de forma que los esfuerzos realizados para mitigar cierto tipo de fallos compensen los perjuicios que provocaría no tolerar ese tipo de fallo.

Hay distintas estrategias para conseguir un sistema lo más tolerante de fallos posible. Las más importantes son:

- Redundancia

Una de las formas en la que las redes confiables proporcionan redundancia es mediante la implementación de una red conmutada por paquetes. La conmutación por paquetes divide el tráfico en paquetes que se enrutan a través de una red compartida. Un solo mensaje, como un correo electrónico o una transmisión de vídeo, se divide en múltiples bloques de mensajes, llamados paquetes. Cada paquete tiene la información de dirección necesaria del origen y el destino del mensaje. Los routers dentro de la red conmutan los paquetes según la condición de la red en ese momento. Esto significa que todos los paquetes en un mismo mensaje pueden tomar distintas rutas para llegar a destino.

Ejemplos de uso de esta estrategia son:

- Uso de códigos detectores y correctores de error.

Tener módulos pasivos que hacen exactamente lo mismo que otros activos de forma que puedan sustituirlo y evitar que el sistema se caiga por el fallo de un ese elemento.
Redundancia modular. Consiste en tener un número normalmente impar (para evitar luego empates) de módulos que hacen la misma función aunque pueden implementarla de forma diferente. Luego hay un módulo (el cual puede tener a su vez redundancia modular) que evalúa las salidas de dichos módulos y toma como resultado global el resultado que devuelve la mayoría de los módulos redundantes.

- Replicación

Para evitar que un fallo produzca la pérdida de la información almacenada en un sistema se suele replicar esa información en más de un soporte físico (redundancia), o en un equipo o dispositivo externo a modo de respaldo. De esta forma, si se produce alguna falla que pueda ocasionar pérdida de datos, el sistema debe ser capaz de restablecer toda la información, recuperando los datos necesarios a partir de algún medio de respaldo disponible.

En esto se basa el sistema de almacenamiento en RAID (Redundant Array of Independent Disks). Los sistemas RAID (a excepción de RAID 0) se basan en la técnica mirroring («en espejo»), que permite la escritura simultánea de los datos en más de un disco del array.
En sistemas distribuidos es frecuente replicar la información para conseguir que sean tolerantes a los fallos. Para hacer que dicha información sea consistente en todo el sistema distribuido se implementan protocolos de consenso

- Autocorrección

Esta estrategia es la que hacen los navegadores de internet. Cuando el navegador de internet envía una solicitud HTTP al servidor WEB este responde con el contenido del sitio en formato estandarizado HTML o XHTML, si este código viene con errores (el estándar no se cumple), entonces el navegador es libre de elegir qué hacer con él, ya sea no mostrar el contenido con problemas, intentar corregirlo o simplemente mostrarlo en texto plano. Normalmente lo que hacen es intentar corregirlo.

¿Qué es una falla?

Fallo es un estado o situación en la que se encuentra un sistema formado por dispositivos, equipos, aparatos y/o personas en el momento que deja de cumplir la función para la cual había sido diseñado.”

Hay que evitar esta situación siempre que queramos diseñar un sistema altamente fiable, competitivo y fuerte. Para ello hay que adelantarse a dicho estado o situación mediante métodos matemáticos y científicos

¿Qué es un error?

En estadística, un error (o residuo) no es un "error" en el sentido de un "descuido", sino más bien una diferencia entre un valor calculado, estimado o medido y el valor verdadero, especificado o teóricamente correcto.

En la ciencia y la ingeniería en general, un error se define como una diferencia entre el rendimiento o comportamiento deseado y el real de un sistema u objeto. Esta definición es la base de operación para muchos tipos de sistemas de control, en los cuales el error se define como la diferencia entre un punto de referencia y el valor del proceso. Un ejemplo de esto sería el termostato en un sistema de calefacción doméstico: la operación del equipo de calefacción es controlada por la diferencia (el error) entre la configuración del termostato y la temperatura del aire detectada. Otro enfoque está relacionado con considerar una hipótesis científica como verdadera o falsa, dando lugar a dos tipos de errores: Tipo 1 y Tipo 2. El primero ocurre cuando una hipótesis verdadera se considera falsa, mientras que el segundo es lo contrario (una hipótesis falsa se considera verdadera).

Los ingenieros buscan diseñar dispositivos, máquinas y sistemas de tal manera que mitiguen o, preferiblemente, eviten los efectos del error, ya sea intencional o no. Dichos errores en un sistema pueden ser errores latentes de diseño que pueden pasar desapercibidos durante años, hasta que surge el conjunto adecuado de circunstancias que los activan. Otros errores en los sistemas diseñados pueden surgir debido a errores humanos, que incluyen sesgos cognitivos. La ingeniería de factores humanos se aplica a menudo a los diseños con el fin de minimizar este tipo de errores haciendo que los sistemas sean más tolerantes o indulgentes con los errores.

¿Que es la latencia de un fallo?

La latencia de un fallo se refiere al período de tiempo durante el cual un error o defecto en un sistema existe pero no se manifiesta o no es detectable. Es un intervalo en el que el fallo está presente en el sistema de manera latente, sin causar efectos visibles o inmediatos en el rendimiento o comportamiento del sistema.

Este tipo de fallos latentes pueden permanecer ocultos durante un largo tiempo hasta que se dan las condiciones específicas que los activan y

provocan un fallo observable. Una vez que se activa, el fallo puede tener consecuencias negativas, como un mal funcionamiento del sistema, errores de cálculo, o incluso un fallo completo.

Un ejemplo de latencia de un fallo podría ser un error de programación en un software que no se manifiesta hasta que una función específica se utiliza bajo ciertas condiciones poco comunes. Otro ejemplo podría ser un componente mecánico defectuoso en una máquina que solo falla bajo determinadas cargas o en circunstancias ambientales particulares.

¿Qué es la latencia de un error?

Es el periodo de tiempo en el que un error está presente pero puede que no afecte visualmente ni en la experiencia al usuario o al operador de un sistema.
Aspectos clave de la latencia de un error:
Tiempo de Manifestación: La latencia de un error puede variar en función de diversos factores, como la naturaleza del sistema, el tipo de error y las condiciones operativas. Algunos errores pueden manifestarse inmediatamente después de su aparición, mientras que otros pueden permanecer ocultos durante largos períodos.

Detección Tardía: La latencia puede influir en la dificultad de detectar errores. Un error con alta latencia puede pasar desapercibido hasta que se producen circunstancias específicas que activan el fallo.
Impacto: Durante el período de latencia, el error puede no causar ningún efecto visible, pero una vez que se activa, puede llevar a fallos graves, problemas de rendimiento o mal funcionamiento del sistema.

Software: En un programa informático, un error en el código puede no causar problemas hasta que se realicen ciertas operaciones o se procesen ciertos datos, lo que puede llevar a fallos en situaciones específicas.

Hardware: En un componente mecánico, un defecto interno puede no manifestarse hasta que el componente sea sometido a condiciones de operación extremas o a un uso prolongado.

Sistemas de Control: En un sistema de control industrial, un error en un sensor o actuador puede no ser evidente hasta que el sistema se someta a condiciones operativas específicas.

Bibliografía:

colaboradores de Wikipedia. (2022, 6 junio). Tolerancia frente a fallos. Wikipedia, la Enciclopedia Libre. https://es.wikipedia.org/wiki/Tolerancia_frente_a_fallos

colaboradores de Wikipedia. (2021, 22 enero). Fallo en producto o sistema. Wikipedia, la Enciclopedia Libre. https://es.wikipedia.org/wiki/Fallo_en_producto_o_sistema#:~:text=Fallo%20es%20un%20estado%20o,la%20cual%20hab%C3%ADa%20sido%20dise%C3%B1ado.%E2%80%9D

Nielsen, J., & Budiu, R. (2012). Mobile usability. Wiley.

Sommerville, I. (2016). Software engineering (10th ed.). Pearson.