CONCEPTOS BÁSICOS

By: Jared Isaías Monje Flores

14 DE AGOSTO DE 2023

CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERIAS

UdeG

Computación tolerante a fallas

En el mundo de la computación tolerante a fallas la comprensión de conceptos fundamentales como “Bug” y “Fallo” es esencial para garantizar la integridad y confiabilidad de los sistemas. Tanto los bugs como los fallos representan desafíos significativos en el desarrollo y operación de software y hardware, pero se diferencian en sus características y consecuencias.

Fault tolerance es la característica incorporada en el sistema que permite su buen funcionamiento incluso después de que ocurre una falla en alguno de sus componentes. Un diseño tolerante a fallas puede causar una reducción en el nivel de productividad o un mayor tiempo de respuesta. Sin embargo, se asegura de que todo el sistema no falle. Por lo tanto, en resumen, funciona como un mecanismo de afrontamiento a un sistema que tiene como objetivo la autoestabilización.

Al principio de la tecnología, los sistemas tolerantes a fallas se diseñaron para dar alarmas al usuario o al operador sobre la posible falla. Se suponía que el operador actuaría sobre la alarma y aclararía las cosas antes de que ocurriera una avería importante. Esto implicó la interferencia humana. Sin embargo, hoy las cosas han cambiado. Los sistemas, ya sean hardware o software, están diseñados para resolver problemas de forma independiente sin mucha interferencia humana, a menos que sea un problema importante que requiera atención inmediata.

La tolerancia a fallas se refiere a la capacidad de un sistema (computadora, red, clúster de nube, etc.) para continuar funcionando sin interrupción cuando uno o más de sus componentes fallan.

El objetivo de crear un sistema tolerante a fallas es prevenir las interrupciones que surgen de un solo punto de falla, asegurando la alta disponibilidad y la continuidad comercial de las aplicaciones o sistemas de misión crítica.

Un "fallo" se refiere a una desviación no deseada o inesperada del funcionamiento normal de un sistema o componente. Este término abarca una amplia gama de situaciones en las que un sistema no puede cumplir con su función prevista, lo que puede resultar en una degradación o interrupción de su operación. Los fallos pueden ser causados por una variedad de factores, incluidos errores de hardware, problemas de software, eventos imprevistos o condiciones extremas.

Los fallos pueden manifestarse de diversas formas, como bloqueos del sistema, respuestas incorrectas, caídas del rendimiento, pérdida de datos o incluso la interrupción completa del servicio. En el contexto de la computación tolerante a fallas, el enfoque se centra en desarrollar estrategias y técnicas para mitigar los

efectos adversos de los fallos y permitir que el sistema continúe funcionando de manera aceptable incluso en presencia de fallas.

Un "bug" sigue refiriéndose a un error o defecto en un programa o sistema, pero en este caso, se considera especialmente relevante debido a su potencial para afectar la confiabilidad y la capacidad de recuperación del sistema en caso de fallos.

En sistemas tolerantes a fallos, un bug puede tener repercusiones más significativas, ya que podría ser la causa subyacente de un posible fallo. Un bug mal gestionado podría debilitar los mecanismos de detección, recuperación o redundancia diseñados para lidiar con fallos. Por ejemplo, si un componente de software tiene un bug que impide que el sistema detecte ciertos fallos, podría resultar en una incapacidad para activar procesos de recuperación adecuados.

Por lo tanto, en el contexto de la computación tolerante a fallos, la identificación y corrección de bugs son de vital importancia. Es esencial asegurarse de que el software y los sistemas sean robustos y estén diseñados para manejar adecuadamente situaciones de fallos, incluso si estos fallos son causados por errores de programación.

La relación entre bugs y tolerancia a fallos destaca la necesidad de realizar pruebas exhaustivas, revisiones de código y análisis de riesgos cuidadosos durante el desarrollo de sistemas críticos para garantizar que los bugs no comprometan la capacidad del sistema para mantener la integridad y la disponibilidad en condiciones adversas.

La "latencia de un fallo" se refiere al tiempo que transcurre desde el momento en que ocurre un fallo en un sistema hasta que dicho fallo se hace evidente o se manifiesta de manera detectable. En otras palabras, es el intervalo de tiempo entre el momento en que ocurre el problema y el momento en que el sistema o sus usuarios perciben los efectos negativos del fallo.

La latencia de un fallo es un concepto importante en la ingeniería de sistemas y en la computación tolerante a fallos. Comprender y medir la latencia de los fallos es crucial para el diseño de sistemas que puedan detectar, recuperarse y responder eficazmente ante situaciones anómalas o problemas inesperados.

En algunos casos, una latencia de fallo corta es deseable, ya que significa que los sistemas pueden identificar rápidamente los problemas y tomar medidas para mitigar sus efectos. Sin embargo, en otros casos, una latencia de fallo más larga puede ser aceptable, especialmente si se trata de situaciones menos críticas o si existen sistemas redundantes que pueden asumir la carga en caso de fallo.

La latencia de un fallo puede variar según el tipo de sistema, la naturaleza del fallo y las estrategias de detección y recuperación implementadas. Para sistemas en tiempo real o aplicaciones críticas, minimizar la latencia de los fallos puede ser esencial para garantizar un rendimiento confiable y una respuesta oportuna ante eventos adversos.

La "latencia de un error" se refiere al tiempo transcurrido desde que un error se introduce en un sistema o proceso hasta que ese error se detecta y se corrige. Es el lapso entre el momento en que se comete un error y el momento en que se toma conciencia de su existencia.

En el contexto del desarrollo de software y la ingeniería de sistemas, la latencia de un error es un factor crítico que puede influir en la calidad y confiabilidad del sistema final. Una latencia de error corta implica que los errores se identifican rápidamente después de su introducción, lo que facilita su corrección y minimiza el impacto en el funcionamiento del sistema. Por otro lado, una latencia de error larga podría permitir que los errores pasen desapercibidos durante más tiempo, lo que podría generar problemas más graves o dificultar su identificación y solución posterior.

La detección temprana de errores y la reducción de la latencia de errores son objetivos clave en el desarrollo de software y en la gestión de sistemas en general. Estrategias como la realización de pruebas exhaustivas, revisiones de código, análisis estático y monitorización constante pueden contribuir a reducir la latencia de errores y mejorar la calidad del software y los sistemas.

En resumen, la latencia de un error se refiere al tiempo que transcurre desde la introducción de un error hasta su detección, y es un concepto importante en la búsqueda de sistemas confiables y de alta calidad.

everRun Enterprise y Stratus Redundant Linux, la plataforma operativa que potencia Stratus' ztC Edge replican todos los datos escritos en el disco (para cargas de trabajo de alta disponibilidad) y utilizan un motor único de checkpointing para replicar continuamente los datos en los estados de la memoria y la CPU (para cargas de trabajo tolerantes a fallos). Todas las operaciones de E/S se ponen en cola hasta que se completan y verifican los puntos de control. Algoritmos propios ajustan dinámicamente la frecuencia de los puntos de control, basándose en el tipo y la cantidad de cambios de datos y el rendimiento de E/S. Si/cuando un nodo falla, se utiliza una pausa de dos segundos para evitar escenarios de cerebro dividido, lo que resulta en un tiempo de recuperación de menos de cinco segundos, por debajo del umbral TCP/IP para poner en cola y reenviar las solicitudes.

En el intrigante ámbito de la computación tolerante a fallos, hemos explorado en detalle los conceptos de fallos, bugs y la crucial latencia de errores. Los fallos, que indican desviaciones indeseadas en el funcionamiento de sistemas, pueden originarse por diversos motivos, desde fallos de programación hasta situaciones extremas en el entorno. En contraposición, los bugs, que representan errores y defectos en el software, pueden actuar como los desencadenantes silenciosos que generan fallos, enfatizando la necesidad de una minuciosa depuración y pruebas.

La latencia de errores, el lapso entre la introducción y la detección de un error, ha adquirido una importancia crucial al dar forma a la fiabilidad y eficacia de los sistemas. Una latencia breve permite reacciones y correcciones rápidas, mitigando el impacto de errores en cadena. Por otro lado, una latencia prolongada podría permitir que los problemas arraiguen y se propaguen, afectando negativamente la estabilidad y seguridad del sistema.

La computación tolerante a fallos busca fortalecer la resiliencia de los sistemas, implementando estrategias como redundancia, detección temprana y recuperación. La comprensión y aplicación adecuada de estos conceptos, desde la identificación y corrección de bugs hasta la gestión de la latencia de errores, se han convertido en fundamentos esenciales para crear sistemas sólidos y confiables, capaces de resistir desafíos cambiantes y mantener un rendimiento aceptable incluso en circunstancias adversas. En última instancia, la comprensión de estos conceptos contribuye al progreso continuo de la tecnología y a la construcción de sistemas capaces de afrontar incertidumbres con éxito.

ciberseg1922. (2021, December 16). *Tolerancia a fallos, qué es y técnicas*. Ciberseguridad. https://ciberseguridad.com/guias/prevencion-proteccion/tolerancia-fallos/

Conozca los servidores con tolerancia a fallos | ¿Qué es la tolerancia a fallos?Stratus. (2022, April 19). Stratus | Zero-Touch Edge Computing. https://www.stratus.com/es/fault-tolerant/

What is Fault Tolerance and How it Works? | vSphere | VMware. (2022, August 9). VMware. https://www.vmware.com/es/products/vsphere/fault-tolerance.html

Conozca los servidores con tolerancia a fallos | ¿Qué es la tolerancia a fallos?Stratus. (2022, April 19). Stratus | Zero-Touch Edge Computing. https://www.stratus.com/es/fault-tolerant/