Computación Tolerante a Fallas

Sección D06

Failure Injection

Fernández Venegas David Guadalupe

Código: 216437352

Profesor: López Franco Michel Emanuel

La inyección de fallos (Failure Injection) es una técnica crucial en el ámbito de la computación y la informática, utilizada para evaluar la resiliencia y la robustez de sistemas y aplicaciones ante situaciones adversas. Este enfoque consiste en introducir fallos deliberadamente en un sistema para observar cómo responde y se recupera, permitiendo identificar debilidades y mejorar su diseño y funcionalidad.

¿En qué consiste la inyección de fallos?

La inyección de fallos implica la introducción intencional de errores, defectos o fallos en un sistema de software o hardware. Estos fallos pueden ser de diversos tipos, incluyendo errores de hardware, fallos de red, errores de programación, condiciones extremas de carga o incluso ciberataques simulados. El objetivo es simular condiciones que podrían ocurrir en entornos de producción para estudiar el comportamiento del sistema bajo estrés.

¿Cómo se realiza la inyección de fallos?

La implementación de la inyección de fallos se puede llevar a cabo mediante varias técnicas:

1. **Inyección de Fallos a Nivel de Software**: Utiliza herramientas y scripts para introducir fallos en el código, como excepciones no manejadas, fugas de memoria o tiempos de espera prolongados.
2. **Inyección de Fallos a Nivel de Hardware**: Simula fallos físicos, como desconexiones de discos duros, pérdida de energía o fallos en componentes de memoria.
3. **Inyección de Fallos en la Red**: Introduce problemas de conectividad, como pérdida de paquetes, latencia elevada o desconexiones intermitentes.
4. **Inyección de Fallos Mediante Servicios de Nube**: Servicios como Chaos Monkey de Netflix, que apagan instancias de servidores al azar para probar la resiliencia de sus aplicaciones en la nube.

¿Por qué se utiliza la inyección de fallos?

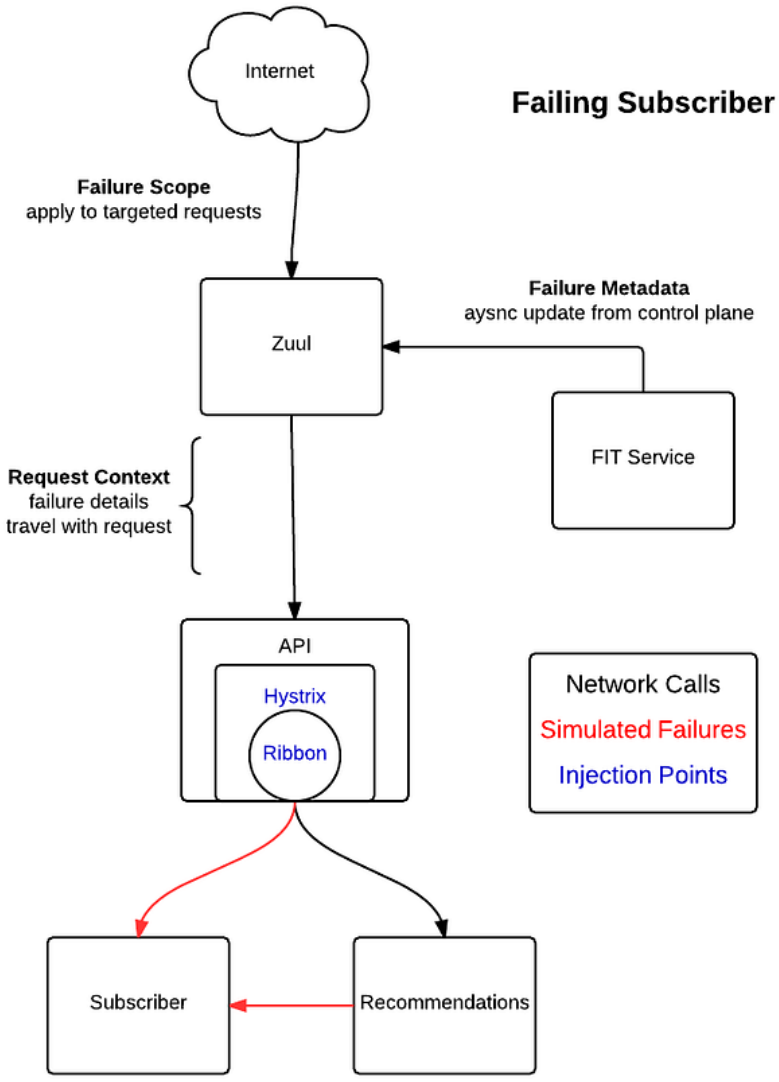
El uso de la inyección de fallos es fundamental por varias razones:

1. **Identificación de Vulnerabilidades**: Permite descubrir puntos débiles en el sistema que podrían causar fallos catastróficos en situaciones reales.
2. **Mejora de la Resiliencia**: Ayuda a diseñar sistemas que puedan recuperarse rápida y eficientemente de fallos, asegurando una mayor disponibilidad y fiabilidad.
3. **Validación de Mecanismos de Recuperación**: Verifica que las estrategias de recuperación y redundancia funcionen correctamente bajo condiciones de fallo.
4. **Preparación para Situaciones Reales**: Proporciona experiencia práctica y datos valiosos para la preparación ante desastres, mejorando las respuestas a incidentes reales.

FIT: Failure Injection Testing. By Netflix

Cómo funciona FIT

La simulación de fallas en FIT comienza cuando el servicio FIT envía metadatos de simulación de fallas a Zuul. Las solicitudes que coinciden con el alcance del fallo en Zuul son decoradas con fallos específicos. Esto puede implicar un retraso adicional en una llamada de servicio o una falla al alcanzar la capa de persistencia. Cada punto de inyección revisa el contexto de la solicitud para determinar si hay una falla aplicable para ese componente específico. Si se detecta una falla, el punto de inyección simula dicha falla de manera adecuada. A continuación, se presenta un esquema de una falla simulada, que muestra algunos de los puntos de inflexión donde se puede inyectar la falla.



Alcance de la falla

Es crucial limitar el alcance potencial de los fallos para asegurar que solo afecten a los objetivos previstos. Para ello, se utiliza Zuul, que ofrece capacidades avanzadas para inspeccionar y gestionar el tráfico. Antes de reenviar una solicitud, Zuul verifica un almacén local de metadatos FIT para determinar si dicha solicitud debe verse afectada. Si es así, Zuul adorna la solicitud con un contexto de error, que luego se propaga a todos los servicios dependientes.

En la mayoría de las pruebas de fallos, Zuul se emplea para aislar las solicitudes afectadas a una cuenta de prueba específica o a un dispositivo particular. Una vez validado en ese nivel, se amplía el alcance a un pequeño porcentaje de solicitudes de producción. Si las pruebas de fallos siguen mostrando resultados favorables, se incrementa gradualmente el caos hasta alcanzar el 100%.

Puntos de inyección

Netflix utiliza varios componentes clave para aislar fallos y definir alternativas. Entre estos componentes se encuentran Hystrix, que ayuda a definir alternativas; Ribbon, que facilita la comunicación con dependencias; EVCache, que maneja el almacenamiento en caché de datos; y Astyanax, que se ocupa de los datos persistentes. Cada una de estas capas actúa como un punto de inyección ideal para introducir fallos. Estas capas interactúan con el contexto FIT para determinar si una solicitud específica debe verse afectada. El comportamiento del fallo se implementa en la capa correspondiente, que decide cómo emular el fallo de manera realista, ya sea suspendiendo la operación durante un período de retraso, devolviendo un error 500, lanzando una excepción, etc.

Escenarios de falla

Para recrear interrupciones pasadas o probar proactivamente la pérdida de una dependencia, es crucial entender qué podría fallar para construir una simulación efectiva. Se utiliza un sistema interno que rastrea las solicitudes en todo el ecosistema de Netflix para identificar todos los puntos de inyección a lo largo del camino. Estos puntos se utilizan para crear escenarios de falla, que son conjuntos de puntos de inyección que deberían o no fallar. Un ejemplo de escenario es el conjunto mínimo de servicios críticos necesarios para la transmisión, conocido como escenario de servicios críticos. Otro ejemplo podría ser la pérdida de un servicio individual, incluyendo sus capas de persistencia y almacenamiento en caché.

Pruebas automatizadas

Las herramientas de prueba de fallos son tan efectivas como su implementación. Los equipos de prueba de dispositivos de Netflix han desarrollado una automatización que habilita fallos, inicia la aplicación de Netflix en un dispositivo, navega por varias listas, selecciona un video y comienza la reproducción. Inicialmente, se valida que este proceso

funcione correctamente solo con los servicios críticos disponibles. Actualmente, se está ampliando este enfoque para identificar cada dependencia involucrada durante el proceso y fallar sistemáticamente en cada una de ellas de manera individual. La ejecución continua de estas pruebas ayuda a identificar vulnerabilidades a medida que se introducen.

Estrategias de resiliencia

FIT ha demostrado ser una herramienta valiosa para cerrar la brecha entre las pruebas aisladas y los ejercicios caóticos a gran escala, permitiendo que dichas pruebas sean de autoservicio. Es una de las muchas herramientas utilizadas para construir sistemas más resilientes. El alcance del problema va más allá de las simples pruebas de fallos; se requiere una variedad de técnicas y herramientas, como el diseño para fallos, una mejor detección y diagnóstico más rápido, pruebas automatizadas periódicas y la implementación de protecciones de barrera. Si este enfoque resulta interesante, Netflix está buscando excelentes ingenieros para unirse a los equipos de confiabilidad, arquitectura de nube y API.

Bibliografía

- http://chaostoolkit.org/reference/usage/install/
- https://www.gremlin.com/get-started/?ref=hero
- https://github.com/netflix/chaosmonkey
- https://principlesofchaos.org/
- https://netflixtechblog.com/fit-failure-injection-testing-35d8e2a9bb2