

Perfiles

Análisis dinámico

- Un perfilador o profiler reúne información sobre la conducta de un programa durante su ejecución
- Es toda aquella investigación del comportamiento de algún programa computacional
- Los programas analizadores de rendimiento en Unix se remontan al menos a 1979
- ATOM es una plataforma para convertir un programa en su propio profiler

Procesos para el desarrollo de software

- Tipo de análisis de software que supone la ejecución del programa
- Para que este análisis dinámico sea efectivo, el programa que será analizado se debe de ejecutar con los suficientes casos de prueba
- Las Pruebas de software son las investigaciones empíricas y técnicas para poder proporcionar información objetiva e independiente sobre la calidad del producto
- Dichas pruebas presentan información sobre la eficacia del producto a las personas responsables de este mismo

En cascada

- Se denomina de este modo, ya que a cada salida de una etapa cae en la siguiente
- Las etapas se llevan a cabo una a continuación de la otra

Desarrollo iterativo y creciente

- Tiene las mismas etapas que la de cascada
- La etapa de relevamiento se divide en distintos subconjuntos
- Cada uno de estos subconjuntos se construye de la misma forma que con el ciclo de vida en cascada

Desarrollo ágil

- Es un proceso iterativo e incremental
- Se caracteriza por contar con iteraciones cortas y por no tener fases lineales

Pruebas estáticas

- Son el tipo de pruebas que se realizan sin ejecutar el código de la aplicación

Pruebas dinámicas

- Se requieren de la ejecución de la aplicación
- Permiten el uso de técnicas de caja negra y caja blanca con mayor amplitud, debido a su naturaleza dinámica
- Pruebas de caja blanca
 - Se realiza sobre las funciones internas de un módulo
- Pruebas de caja negra
 - Técnica donde se busca la verificación de las funcionalidades del software o aplicación analizada

Cobertura de código

- Mide el grado en que el código fuente de un programa ha sido comprobado con pruebas de software
- Sirve para determinar la calidad del test que se lleve a cabo y para determinar las partes críticas del código que no han sido comprobadas y las partes que ya lo fueron
- Se puede utilizar como técnica de optimización dentro de un compilador optimizador para llevar a cabo una eliminación de código muerto

Código inalcanzable

- Aquel que nunca podrá ser ejecutado porque no existe ningún camino dentro de las estructuras de control en el resto del programa para llegar a ese código
- Suele referirse a este tipo de código como código muerto
- Se considera indeseable por
 - Ocupar memoria innecesaria
 - Generar almacenamiento innecesario en la caché de instrucciones del CPU
 - Pierde tiempo y esfuerzo en mantener y documentar una pieza de código que nunca se ejecuta

Compilador optimizador

- Es un compilador que trata de minimizar ciertos atributos de un programa informático con el fin de aumentar la eficiencia y rendimiento
- Las optimizaciones del compilador se aplican generalmente mediante una secuencia de transformaciones de optimización

Fuzzer

- Es un programa que intenta descubrir vulnerabilidades de seguridad enviando una entrada arbitraria a una aplicación
- Estos generan fallos y los envían a una aplicación

Profiler

- Puede proporcionar distintas salidas, como una traza de ejecución o un resumen estadístico de los eventos observados
- Es utilizado durante el desarrollo de software como método para la depuración y optimización de los algoritmos
- Se pueden clasificar según la forma de recopilación de datos que utilicen

¿Cómo implementar un perfilador?

- Primeramente, debemos de tener en cuenta que la arquitectura si hablamos de un sistema, es un insumo fundamental para poder estudiar su performance
- Es el esqueleto del sistema e influye en los atributos de calidad a evaluar a través de elementos tales como: la interacción entre componentes y su distribución física
- La arquitectura va a determinar dónde hay más procesamiento y dónde están los motores de la aplicación
- Posteriormente, necesitamos implementar herramientas para generación automática de carga
- Estas permiten ejecutar una y otra vez las mismas o diferentes pruebas, incluso con diferentes objetivos
- Son muy útiles para simular diferentes escenarios variando la cantidad de usuarios
- completadas las etapas anteriores, tendremos todo lo necesario para realizar las ejecuciones y los reportes pertinentes
- La diferencia entre los tiempos base y los tiempos obtenidos en las diferentes pruebas brindará un indicativo de la sensibilidad de la misma a la concurrencia
- La ejecución de los escenarios definidos se realiza de manera gradual
- Las herramientas de análisis de programas son extremadamente importantes para comprender el comportamiento del programa
- Los arquitectos informáticos necesitan estas herramientas para evaluar el rendimiento de los programas en las nuevas arquitecturas
- Los escritores de software necesitan herramientas para analizar sus programas e identificar secciones críticas de código
- Los escritores de compiladores a menudo usan tales herramientas para averiguar qué tan bien se está desempeñando su programación
- Algunos perfiladores operan por muestreo

Las etapas para implementar un perfilador son

- Análisis de requerimientos
- Automatización
- Armado del ambiente de pruebas
- Ejecución y reportes

Los diferentes tipos de arquitecturas son

- Monolítica
 - El software se estructura en grupos funcionales muy acoplados
 - No hay distribución, ni a nivel físico (hardware) ni a nivel lógico (software)
 - Todo se ejecuta en una máquina, generalmente por un único usuario
- Cliente - servidor
 - Formada por un programa que realiza peticiones a otro programa para obtener cierta respuesta
 - Uno de los clientes más utilizados, es el navegador web
- En capas
 - Cada una de sus capas tiene responsabilidades definidas y encapsula un conjunto de funcionalidades relacionadas
- Cloud computing
 - Las aplicaciones del sistema se ejecutan en ambientes remotos, virtualizados y accesibles a través de Internet

Ejemplos de perfiladores utilizados en la industria son

- SystemTap
 - Herramienta de trazado y sondas que permite a los usuarios monitorizar y analizar en detalle las actividades del sistema operativo
 - Proporciona información similar a la salida de herramientas como netstat, top, ps y iostat
 - proporciona un análisis más preciso y profundo de las actividades del sistema y de su conducta
- OProfile
 - Es otra herramienta de monitorización de rendimiento de todo el sistema
 - Utiliza hardware dedicado de monitorización de rendimiento de procesos para recuperar información sobre el kernel y los ejecutables del sistema
 - También puede utilizarse con Eclipse a través del complemento de Eclipse OProfile
 - Se enfoca en la identificación de problemas con procesos de CPU limitada
- Valgrind
 - analiza su aplicación al ejecutarla en una CPU sintética e instrumentando el código de aplicación existente mientras se ejecuta
 - El nivel de instrumentación varía según la herramienta Valgrind en uso y sus parámetros
 - Puede servir para usar su aplicación sin recompilar