INSTITUTO TECNOLÓGICO DE NUEVO LEÓN

Ingeniería En Sistemas Computacionales

Lenguajes y Autómatas II

U3

Tema: Optimización

Proyecto 3

Profesor.
Juan Pablo Rosas Baldazo

Presenta.

Luis Fernando Dominguez Perez

14480482

Índice

Introducción

En este documento veremos sobre la optimización que es, para que sirve y cuáles son los diversos tipos de optimización.

Además de saber tener la noción de cuanto es el costo que implica tener este tipo de optimización, para la mejora del código, asi como la mejora de optimización gracias a herramientas de entendimiento del flujo de datos

1. Tipos de Optimización

La optimización busca mejorar la forma en que un programa utiliza los recursos. Las optimizaciones se realizan en base al alcance ofrecido por el compilador. La optimización va a depender del lenguaje de programación y es directamente proporcional al tiempo de compilación; es decir, entre más optimización mayor tiempo de compilación.

La optimización es un proceso que tiende a minimizar o maximizar alguna variable de rendimiento, generalmente tiempo, espacio, procesador, etc.

Desafortunadamente no existen optimizador que hagan un programa más rápido y que ocupe menor espacio.

La optimización se realiza reestructurando el código de tal forma que el nuevo código generado tenga mayores beneficios. La mayoría de los compiladores tienen una optimización baja, se necesita de compiladores especiales para realmente optimizar el código. Cada optimización está basada en una función de coste y en una transformación que preserve el significado del programa.

Los diversos tipos de optimización son los siguientes:

1.1 Locales

La optimización local se realiza sobre módulos del programa. En la mayoría de las ocasiones a través de funciones, métodos, procedimientos, clases, etc. La característica de las optimizaciones locales es que sólo se ven reflejados en dichas secciones.

La optimización local sirve cuando un bloque de programa o sección es crítico, por ejemplo: la E/S, la concurrencia, la rapidez y confiabilidad de un conjunto de instrucciones. Como el espacio de soluciones, es más pequeño la optimización local es más rápida.

1.2 Bucles

Los ciclos son una de las partes más esenciales en el rendimiento de un programa dado que realizan acciones repetitivas, y si dichas acciones están mal realizadas, el problema se hace N veces más grandes. La mayoría de las optimizaciones sobre ciclos tratan de encontrar elementos que no deben repetirse en un ciclo.

El problema de la optimización en ciclos y en general radica en que es muy difícil saber el uso exacto de algunas instrucciones. Así que no todo código de proceso puede ser optimizado. Otro uso de la optimización puede ser el mejoramiento de consultas en SQL o en aplicaciones remotas (sockets, E/S, etc.).

1.3 Globales

La optimización global se da con respecto a todo el código. Este tipo de optimización es más lenta, pero mejora el desempeño general de todo programa. Las optimizaciones globales pueden depender de la arquitectura de la máquina.

1.4 Mirilla

La optimización de mirilla trata de estructurar de manera eficiente el flujo del programa, sobre todo en instrucciones de bifurcación como son las decisiones, ciclos y saltos de rutinas. La idea es tener los saltos lo más cerca de las llamadas, siendo el salto lo más pequeño posible.

Se recorre el código buscando combinaciones de instrucciones que pueden ser reemplazadas por otras equivalentes más eficientes.

2. Costos

Los costos son el factor más importante a tomar en cuenta a la hora de optimizar ya que en ocasiones la mejora obtenida puede verse no reflejada en el programa final, pero si ser perjudicial para el equipo de desarrollo. La optimización de una pequeña mejora tal vez tenga una pequeña ganancia en tiempo o en espacio, pero sale muy costosa en tiempo en generarla.

2.1 Costo de ejecución (memoria, registros, pilas)

Los costos de ejecución son aquellos que vienen implícitos al ejecutar el programa. En algunos programas se tiene un mínimo para ejecutar el programa, por lo que el espacio y la velocidad de los microprocesadores son elementos que se deben optimizar para tener un mercado potencial más amplio. Las aplicaciones multimedia como los videojuegos tienen un costo de ejecución alto por lo cual la optimización de su desempeño es crítico, la gran mayoría de las veces requieren de procesadores rápidos (e.g. tarjetas de video) o de mucha memoria. Otro tipo de

aplicaciones que deben optimizarse son las aplicaciones para dispositivos móviles. Los dispositivos móviles tienen recursos más limitados que un dispositivo de cómputo convencional razón por la cual, el mejor uso de memoria y otros recursos de hardware tiene mayor rendimiento.

2.2 Criterios para mejorar el código

La mejor manera de optimizar el código es hacer ver a los programadores que optimicen su código desde el inicio, el problema radica en que el costo podría ser muy grande ya que tendría que codificar más y/o hacer su código más legible.

Los criterios de optimización siempre están definidos por el compilador. Muchos de estos criterios pueden modificarse con directivas del compilador desde el código o de manera externa.

2.3 Herramientas para el análisis del flujo de datos

Existen algunas herramientas que permiten el análisis de los flujos de datos. La optimización al igual que la programación es un arte y no se ha podido sistematizar del todo.

Entre las herramientas más importantes están:

- **Depurador:** permite correr otros programas, permitiendo al usuario ejercer cierto control sobre los mismos a medida que estos se ejecutan, y examinar el estado del sistema.
- **Desamblador:** programa de computadora que traduce el lenguaje máquina a lenguaje ensamblador, la operación inversa de la que hace el ensamblador, está dirigido a un lenguaje de alto nivel.
- **Diagrama de flujo:** herramienta de modelización que permite describir, de un sistema, la transformación de entradas y salidas.
- **Diccionario de datos:** listado organizado de todos los elementos de datos que son pertinentes para el sistema, con definiciones precisas y rigurosas que le permite al usuario y al proyectista del sistema tener una misma comprensión de las entradas y salidas.

Conclusión:

La optimización n muchos casos no se toma en cuenta ya hasta que el problema de rendimiento esta presente, con esto se conlleva una mayor costo y trabajo, para esto sirve la optimización la cual en lo que pude apreciar se debe tener en claro desde un principio para optimizar costos y saber que herramientas utilizar las cuales se adecuen a lo que ocupa el desarrollador.

Conceptos

Bifurcacion: es la creación de un proyecto en una dirección distinta de la principal u oficial tomando el código fuente del proyecto ya existente.

Bibliografía

S.A, S.F, Un. VII. Optimización, https://ingarely.files.wordpress.com/2012/11/unidad-vii.pdf

José M. García Carrazco, S.F, La optimización: una mejora en la ejecución de programas, http://ditec.um.es/~jmgarcia/papers/ensayos.pdf

Gabriela Fernández Espinoza, 13 noviembre 2013, optimización, Mis tareas, http://gaferz.blogspot.mx/2013/11/tipos-de-optimizacion.html

Juan Carlos Olivares Rojas, S.F, Unidad VII Optimización, http://dsc.itmorelia.edu.mx/~jcolivares/courses/ps207a/ps2_u7.pdf

Juan José Sánchez, 19 enero 2014, 3.2.1 Costo de ejecución. (Memoria, registros, pilas), https://prezi.com/m-ft53psccpy/321-costo-de-ejecucion-memoria-registros-pilas/

Claudia Dávila, 21 octubre 2014, 3.2.3 herramientas para el análisis del flujo de datos, https://prezi.com/4dtcp9qnkjbk/323-herramientas-para-el-analisis-del-flujo-de-datos/