Práctica 1: Eficiencia

Antonio Manuel Fernández Cantos

12 de octubre de 2015

Índice

1. Introducción

La práctica consiste en calcular la eficiencia **teórica y empírica** de un código en c++ y **realizar un ajuste de la curva de eficiencia teórica a la empírica.** Se utilizará la biblioteca **ctime** para poder obtener los resultados empíricos. Dentro de la biblioteca ctime tenemos la función **clock()** que devuelve el número de ticks que han transcurrido desde un momento determinado, es esta función la que usaremos para medir la diferencia de tiempo entre el inicio del algoritmo y su finalización.

2. Componentes utilizados

En el cálculo empírico, el algoritmo tardará más o menos en función de:

- **Hardware usado**:
  - CPU
  - RAM
  - HDD
- **Sistema Operativo**
- **Compilador (y sus opciones de compilación)**

- **Bibliotecas**

Todos estos componentes se tienen en cuenta cuando se obtiene el tiempo que tarda nuestro algoritmo en ejecutar todas las sentencias. Dependiendo de la potencia de nuestro ordenador y de las librerias usadas, el algoritmo tardará más o menos. Para la realización del cálculo empírico de los ejercicios, he usado los siguientes componentes:

- **Hardware usado**:
  - Procesador: 8x Intel(R) Core(TM) i7-3630QM CPU@2.40MHz
  - RAM: 6GB
  - CPU clock: 1200 MHz
  - HDD: 750GB
- **Sistema Operativo**: Ubuntu 14.04.3 LTS
- **Compilador**: GCC sin opciones de compilación
- **Bibliotecas**:
  - iostream (E/S)
  - ctime (Para medir el tiempo de ejecución de un algoritmo)
  - cstdlib (Para generar números pseudoaleatorios)

3. Ejercicio 6: Influencia del proceso de compilación

3.1. Introducción

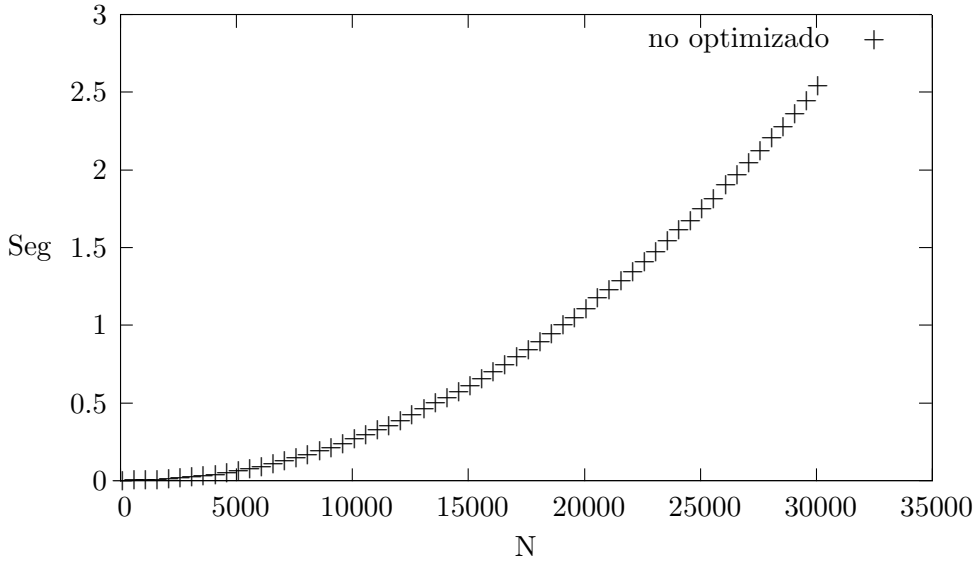
Retomaremos el ejecutable del **ejercicio 1** (ordenación por burbuja). Pero ahora compilaremos el programa con la opción -O3. La orden sería la siguiente:

```
1 g++ -O3 ordenacion.cpp -o ordenacion_optimizado
```

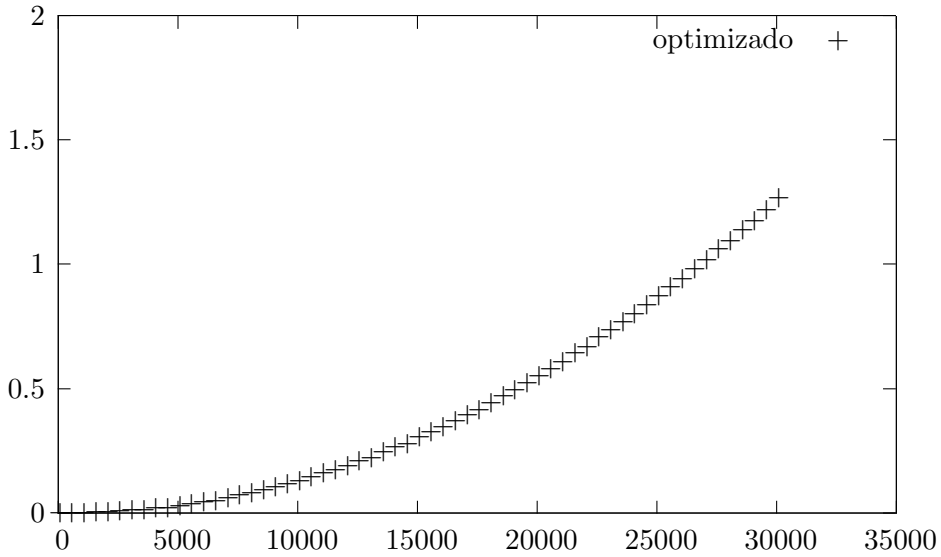
Posteriormete compararemos la curva obtenida en el ejercicio 1 con la curva obtenida con el código optimizado.

3.2. Comparativa

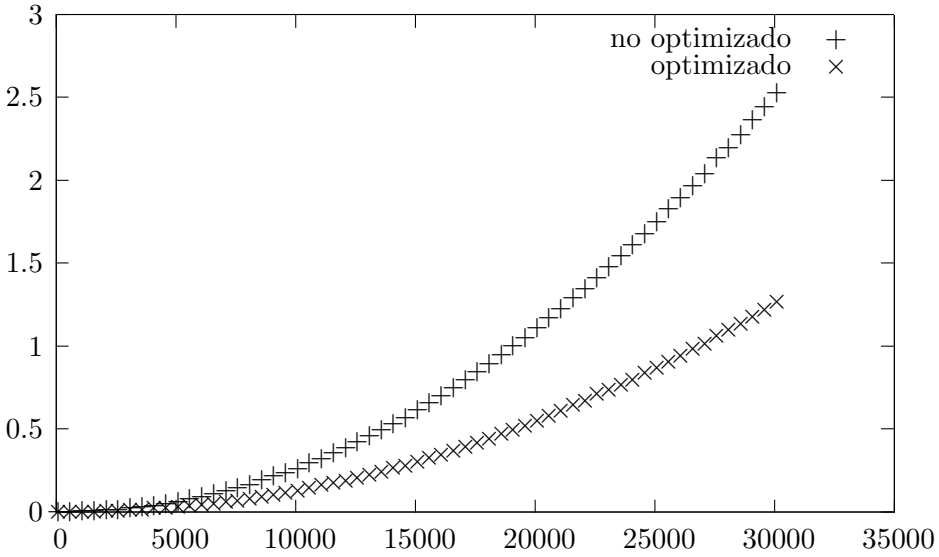
Retomando el ejercicio 1, la gráfica que obtuve fue la siguiente:



A continuación obtenemos la eficiencia empírica con las opciones en la compilación de optimización del código. La gráfica que obtenemos es la siguiente:



Como se puede apreciar, el código con opciones de optimizado en tiempo de compilación tarda menos tiempo en ejecutar la misma cantidad de datos que el código sin opciones de optimizado. Vamos a juntar las dos curvas en una misma gráfica y podremos apreciarlo mejor:



En la gráfica vemos lo que comentabamos anteriormente, el tiempo a la hora de ejecutar el mismo tamaño de vector es **menor** con las opciones de optimizado.