

Lab Report

ECPE 170 – Computer Systems and Networks – Fall 2021

Name: Dustin Schuette

Lab Topic: Performance Optimization (Lab #: 06)

I am booting directly to linux, not using a virtual machine.

(6) Boot Linux. With no applications running in Linux, how much RAM is available inside the virtual machine? The "System Monitor" program should report that information. This is the space that is actually available for our test application.

4.6GB

(7) What is the code doing? (Describe the algorithm in a paragraph, focusing on the `combine1()` function.)

The function receives a vector of numbers and can either multiply all of them or add all of them.

(8) What is the largest number of elements that the vector can hold WITHOUT using swap storage (virtual memory), and how much memory does it take? Be sure to leave enough memory for Firefox and LibreOffice, since you'll need those when running this lab as well.

1 000 000 000 elements (testing with resource monitor)

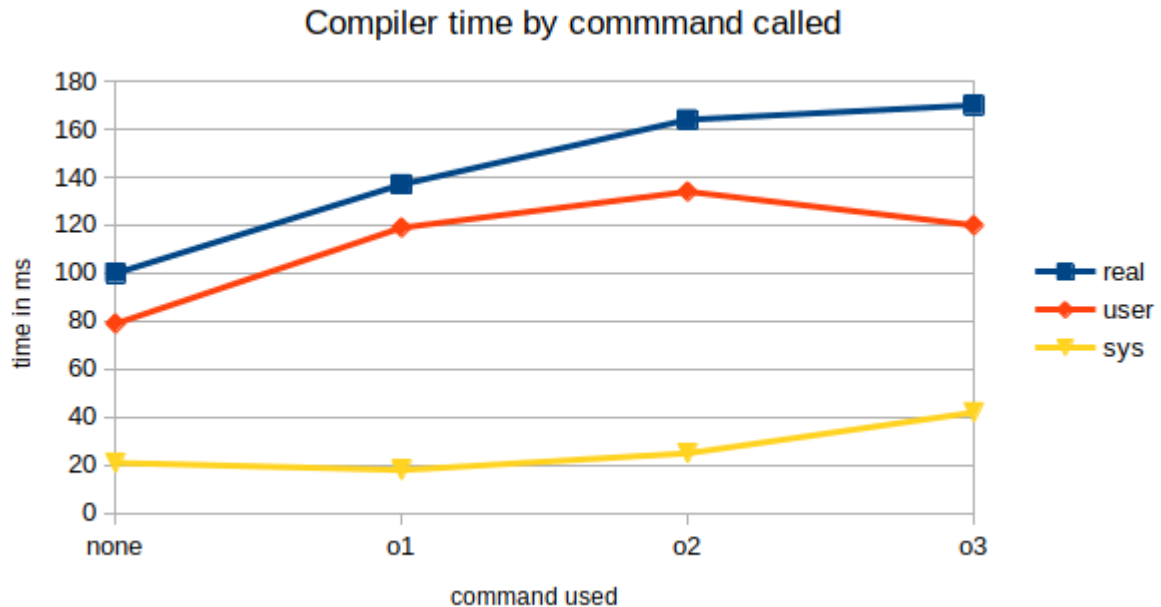
(9) What vector size are you using for all experiments in this lab?

1 000 000 000

(10) How much time does the compiler take to finish with (a) no optimization, (b) with -O1 optimization, (c) with -O2 optimization, and (d) with -O3 optimization?

Report the Real time, which is the "wall clock" time. Create both a table and a graph in LibreOffice Calc.

	real	user	sys
none	100	79	21
o1	137	119	18
o2	164	134	25
o3	170	120	42



(11) How much time does the program take to finish with (a) no optimization, (b) with -O1 optimization, (c) with -O2 optimization, and (d) with -O3 optimization? Report the Real time, which is the "wall clock" time. Create both a table and a graph in LibreOffice Calc.

Note: No credit will be given for sloppy graphs that lack X and Y axis labels, a legend, and a title.

(12) After implementing each function, benchmark it for a variety of data types and mathematical operations. Fill in the table below as you write each function.

(13) Using LibreOffice Calc, make two graphs:

Graph 1: Create a single graph that shows the data in the table created, specifically the four time columns. You don't need to plot vector size.

Graph 2: For FP (float) multiply only, plot a line graph that shows the speed-up of combine2(), combine3(), combine4(), combine5x2(), combine5x3(), and combine6() over combine1() for the vector size tested in Question 12. Plot speed-up on the y axis and function names on the x-axis.

Note that the speed-up of program A over program B is defined as (TB/TA) where TB is the execution time for program B and TA is the execution time for program A.

Note: No credit will be given for sloppy graphs that lack X and Y axis labels, a legend (for graph 1), and a title.

config	vector size (elements)	vector size (MB)	Time for Int Add	Time for Int Multiply	Time for Float add	Time for float mult
combine1()	1 billion	3814.70 MB	7.084	7.03	7.167	7.053
combine2()	1 billion	3814.70 MB	6.085	6.04	6.012	5.897

combine3()	1 billion	3814.70 MB	3.046	3.238	3.264	3.26
combine4()	1 billion	3814.70 MB	2.936	3.228	3.292	3.244
combine5x2()	1 billion	3814.70 MB	2.219	3.253	3.229	3.217
combine5x3()	1 billion	3814.70 MB	2.17	3.226	3.218	3.232
combine6()	1 billion	3814.70 MB	2.2	3.231	3.222	3.246

(14) As a reminder, you should be using version control to track your code, and ensure that the final code is checked in along with your report PDF.

