Name: Ramon Arambula

Lab Topic: Performance Optimization (Lab #:5)

<u>Part 1 - Program Operation</u>

Question #1:
What is the total physical RAM installed in the system? (In MB)
Answer:
RAM: 16gb = 16,000 mb

Question #2:
With no applications running (beyond the web browser with this page), how much RAM is used by the native operating system? (e.g. Windows)
Answer:
OS: MacOS
RAM used by OS: 4.51 GB

Question #3:
With no applications running (beyond the web browser with this page), how much RAM is available?
Answer:
16GB - 4.51 GB = 10.49 GB of RAM available

Question #4:
Check the virtual machine configuration. How much RAM is currently allocated to Linux in your virtual machine?
Answer:
2048 mb is allocated to virtual machine

Question #5:
Try to increase your virtual machine memory allocation, if possible, to the maximum allowed based on your free RAM. Leave ~256MB free for the virtual machine program itself. Now how much RAM is allocated to Linux in your virtual machine?
Answer:
4096 mb of RAM is now being allocated to the virtual machine

Question #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.
Answer:
There is 2 GiB of memory available (since 1.8 are already being used by the virtual machine processes)

Question #7:
What is the code doing? (Describe the algorithm in a paragraph, focusing on the combine1() function.)
Answer:
The code is iterating through all the values in a vector and performing an operation on the element. The operation is either add or multiply, and it is updating the element value based on the operation and saving it in memory.

Question #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.
Answer:
500,000,000 is the maximum amount of elements that can be stored in the vector before swaps are used. It used 1907.35 mb of memory

Part 2 - Compiler Optimizations

Question #9:

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

Answer:

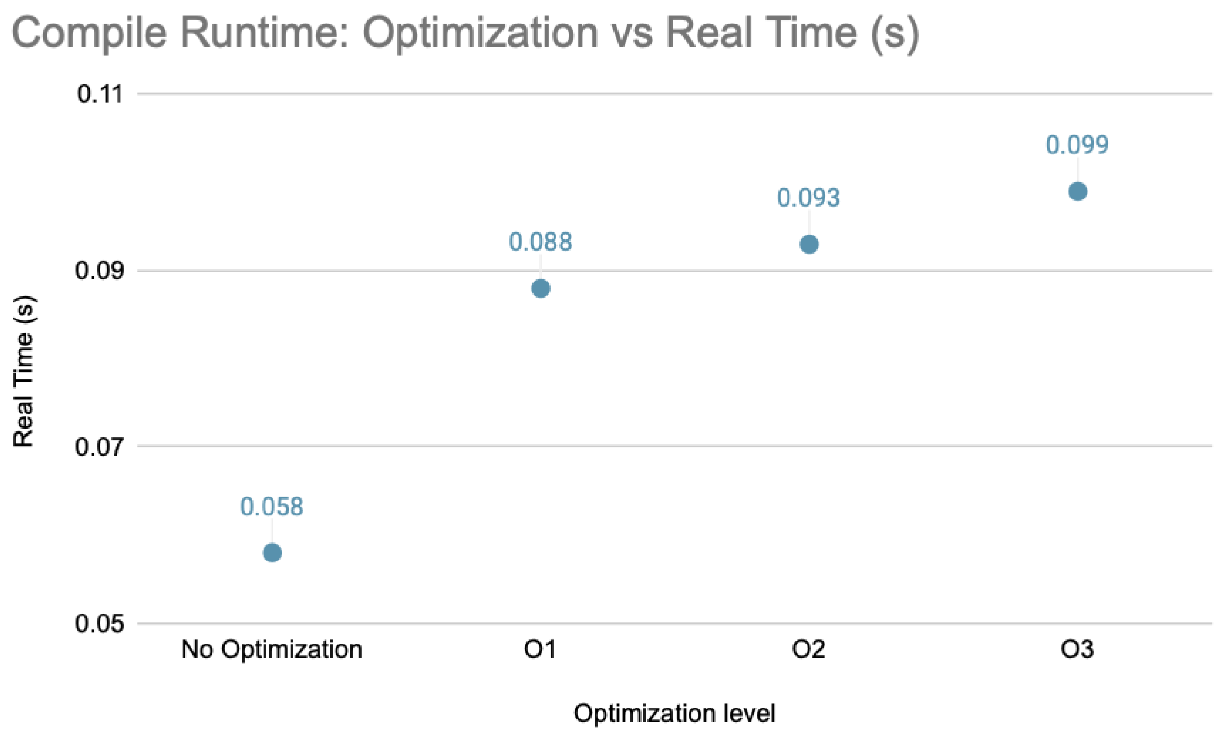
500.000.000 elements

Question #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.

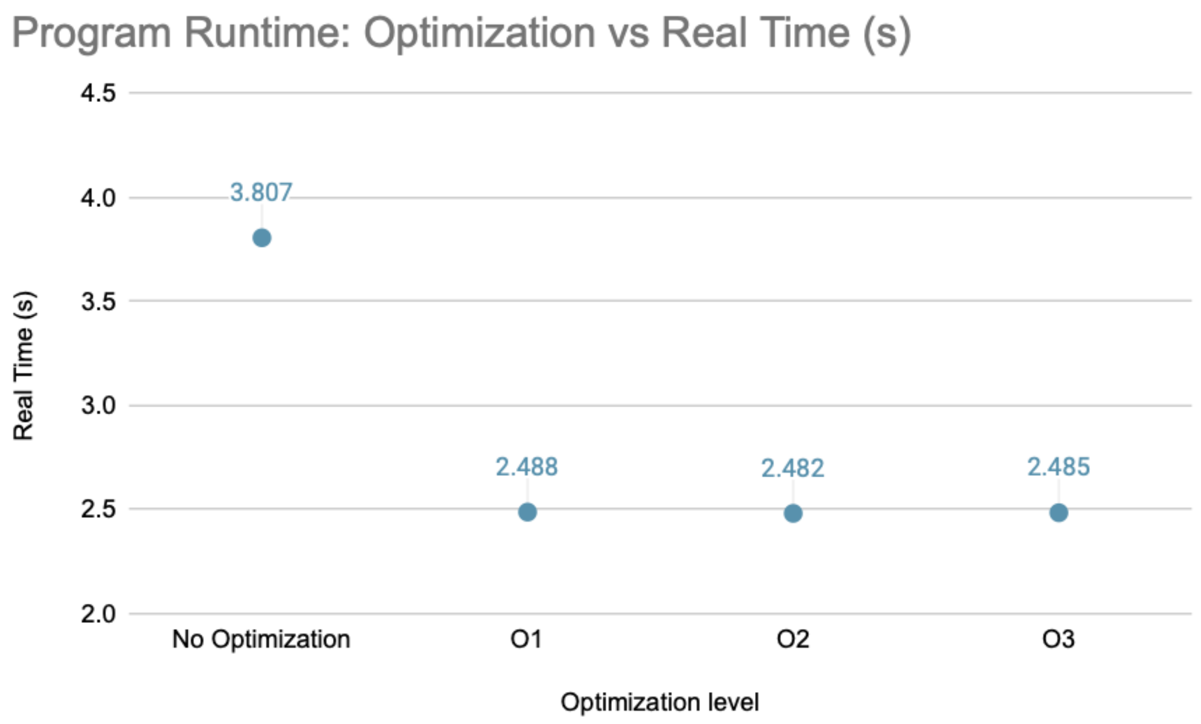
Answer:

Real time	No optimization	-O1	-O2	-O3
Compile time	0.058s	0.088s	0.093s	0.099s



Question #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.
Answer:

Real time	No optimization	-O1	-O2	-O3
Program time	3.807s	2.488s	2.482s	2.485s

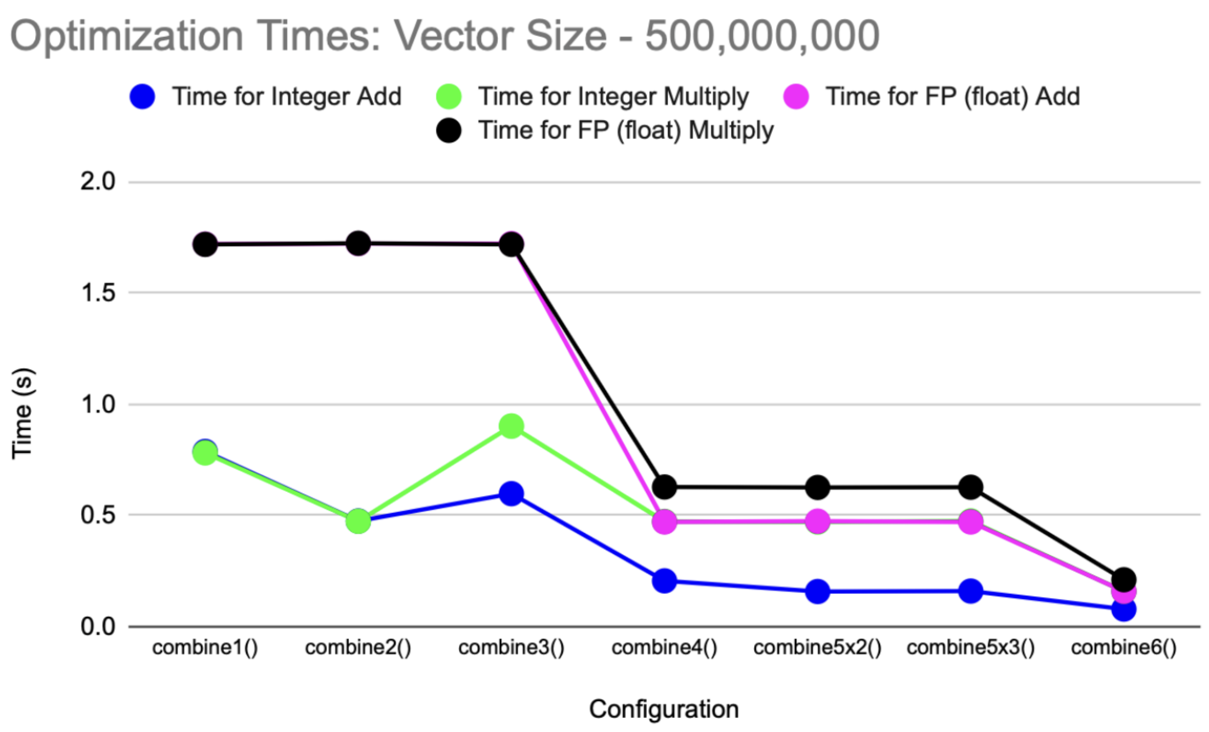


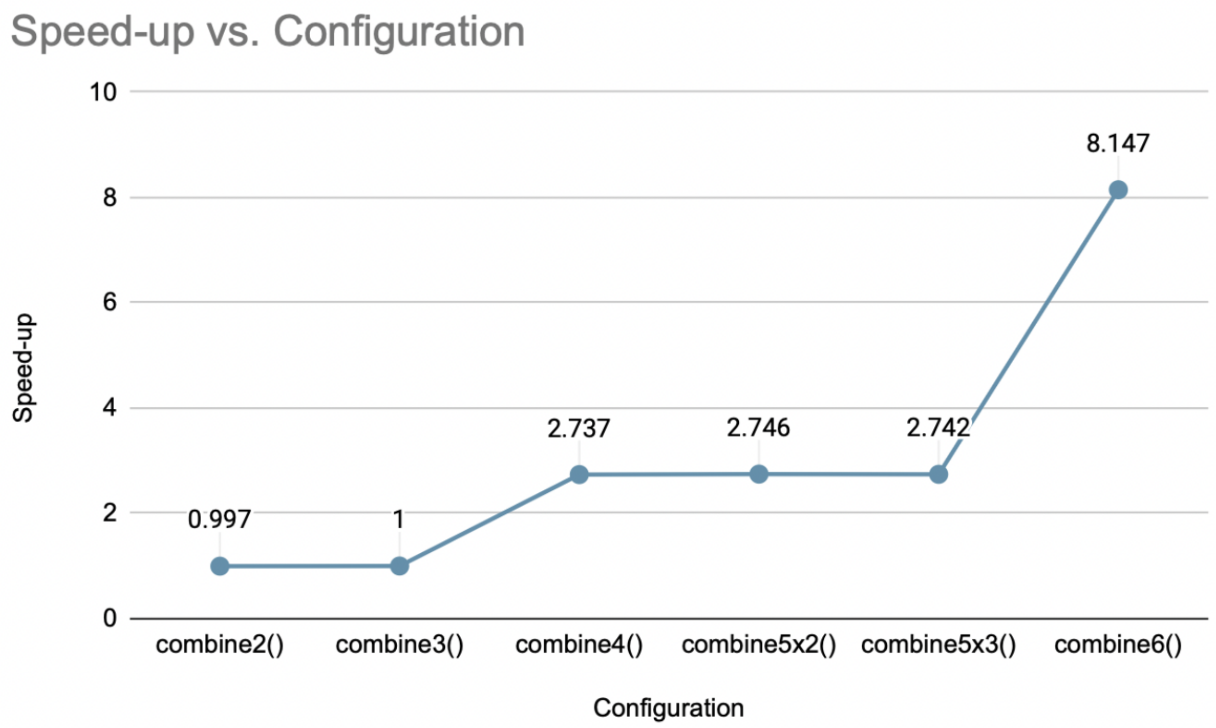
<u>Part 3 - Code Optimizations</u>
Size of 500,000,000
Question #12:
After implementing each function, benchmark it for a variety of data types and mathematical operations. Fill in the table below as you write each function.
Answer:

Configuration	**Vector Size (elements)**	**Vector Size (MB)**	**Time for Integer Add**	**Time for Integer Multiply**	**Time for FP (float) Add**	**Time for FP (float) Multiply**
combine1()	500,000,000	1907.35 MB	0.788s	0.782s	1.720s	1.719s
combine2()	500,000,000	1907.35 MB	0.475s	0.474s	1.723s	1.724s
combine3()	500,000,000	1907.35 MB	0.598s	0.902s	1.722s	1.719s
combine4()	500,000,000	1907.35 MB	0.206s	0.473s	0.471s	0.628s
combine5x2()	500,000,000	1907.35 MB	0.158s	0.471s	0.474s	0.626s
combine5x3()	500,000,000	1907.35 MB	0.160s	0.475s	0.471s	0.627s
combine6()	500,000,000	1907.35 MB	0.079s	0.159s	0.159s	0.211s

Question #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.

Answer:




Speed-up vs. Configuration
10
8
6
4
2
0
Speed-up
8.147
2.737
2.746
2.742
0.997
1
combine2()
combine3()
combine4()
combine5x2()
combine5x3()
combine6()
Configuration