Lab Report

ECPE 170 – Computer Systems and Networks – Spring 2021

Name: Kaung Khant Pyae Sone

Lab Topic: Performance Optimization   (Lab #: 6)

Question #1:
What is the total physical RAM installed in the system? (In MB)
Answer:
16000 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:
4380 MB

Question #3:
With no applications running (beyond the web browser with this page), how much RAM is available?
Answer:
11846 MB

Question #4:
Check the virtual machine configuration. How much RAM is currently allocated to Linux in your virtual machine?
Answer:
4096 MB

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:
8697 MB

Question #7:
What is the code doing? (Describe the algorithm in a paragraph, focusing on the combine1() function.)
Answer:
The code takes in a pointer to a vector **v** and a pointer to the variable with an unknown data type **dest**. **dest** is then set to **IDENT**, which is 0 for addition and 1 for multiplication. The for loop then retrieves every element in vector **v**, then performs either addition or multiplication on **dest** using the value retrieved.

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:
2100000000

Question #9:
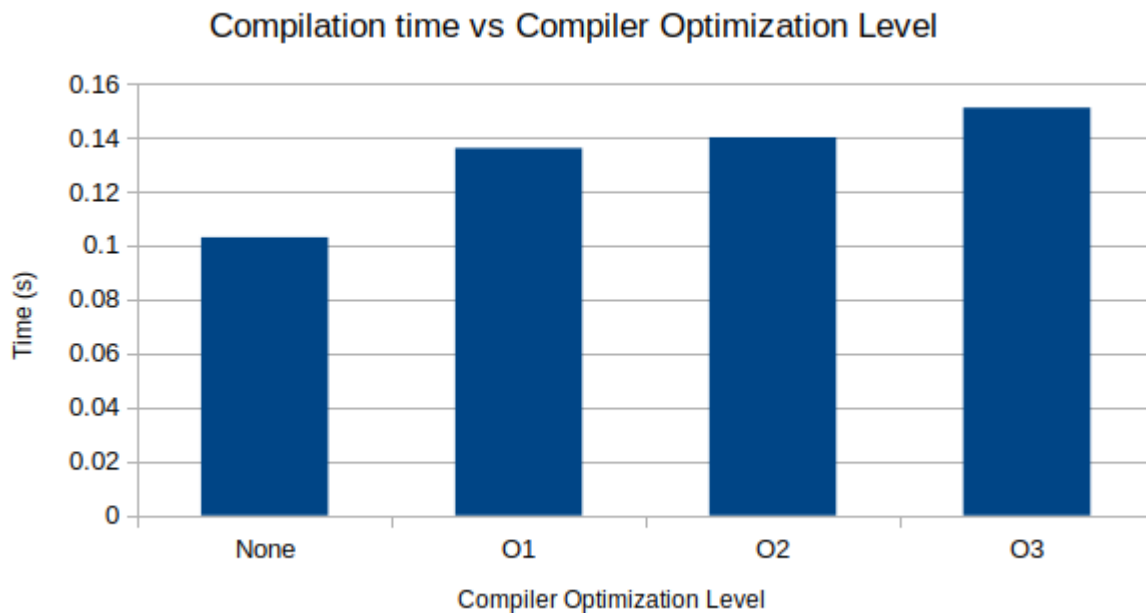What vector size are you using for all experiments in this lab?
Answer:
2100000000

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:

Optimization	Real Time (s)
None	0.103
O1	0.136
O2	0.14
O3	0.151

Compilation time vs Compiler Optimization Level

Time (s) vs Compiler Optimization Level

| Compiler Optimization Level	Time (s)
None	~0.103
O1	~0.136
O2	~0.14
O3	~0.151

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.
Answer:
At vector size of 2100000000,

Optimization	Real Time (s)
None	21.487
O1	10.911
O2	10.106
O3	10.164

Program execution time vs Compiler optimization level
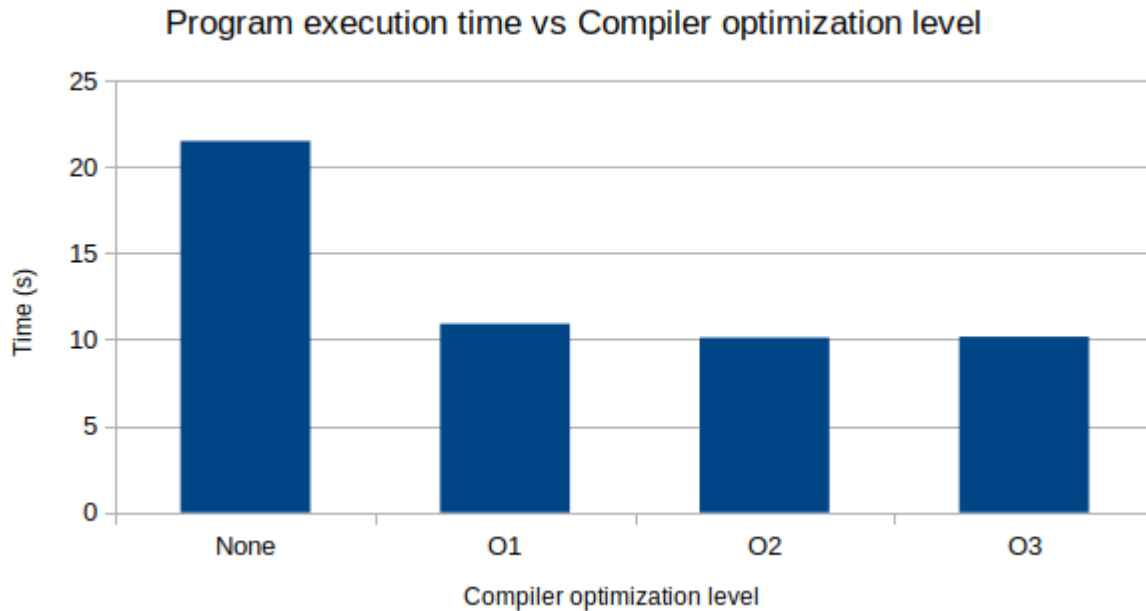
Question #12:
After implementing each function, benchmark it for a variety of data types and mathematical operations. Fill in the table below as you write each function.
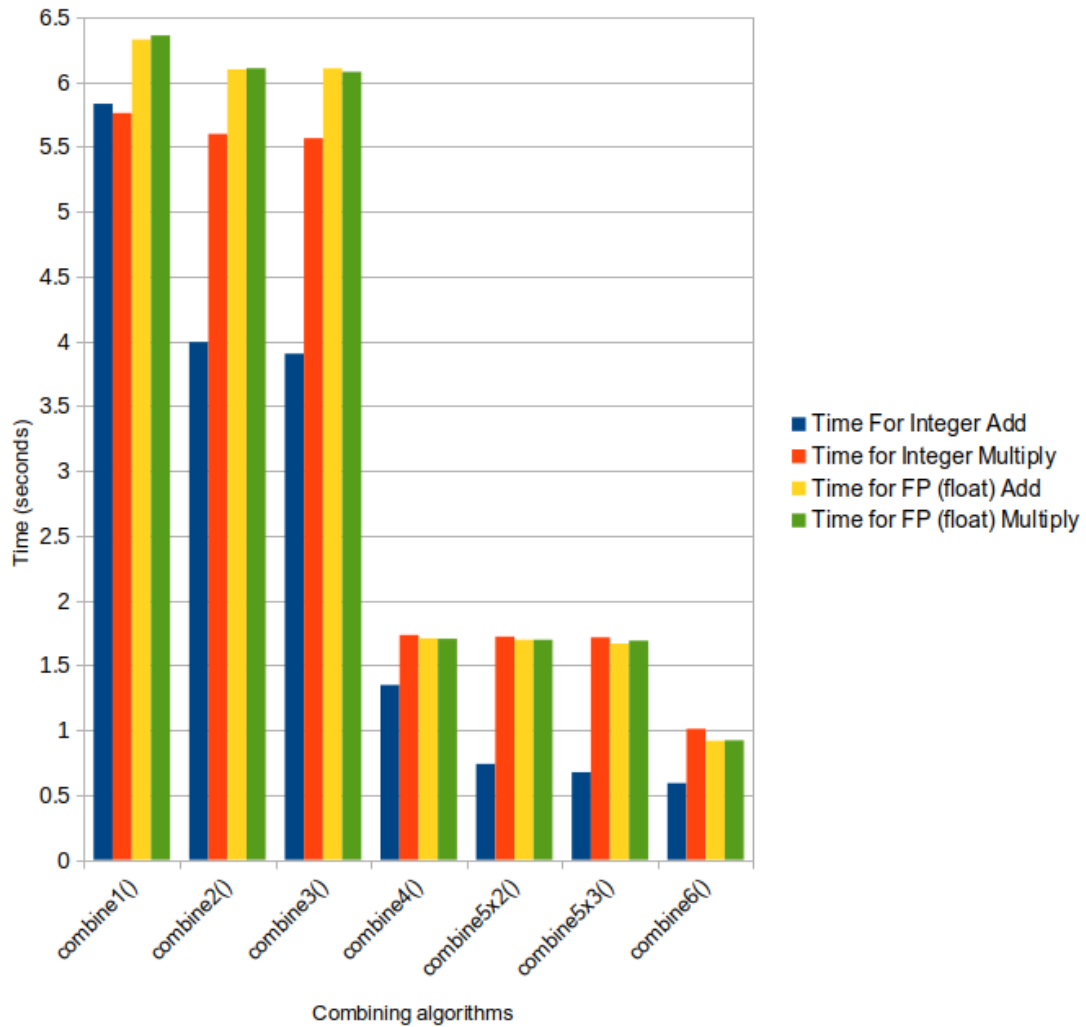**Answer:**

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()			5.832s	5.761s	6.328s	6.360s
combine2()			3.994s	5.600s	6.098s	6.107s
combine3()			3.905s	5.567s	6.106s	6.079s
combine4()	2100000000	8010.86	1.349s	1.735s	1.709s	1.705s
combine5x2()			0.740s	1.722s	1.698s	1.698s
combine5x3()			0.676s	1.716s	1.668s	1.691s
combine6()			0.591s	1.011s	0.918s	0.922s

Configuration	Speedup vs combine1()
combine2()	1.041
combine3()	1.046
combine4()	3.730
combine5x2()	3.746
combine5x3()	3.761
combine6()	6.898

Benchmarking results for different combining algorithms
Run at O1 optimization, vector size of 2100000000 occupying 8010.86 MB

Legend:
- Time For Integer Add
- Time for Integer Multiply
- Time for FP (float) Add
- Time for FP (float) Multiply

Y-axis: Time (seconds)
X-axis: Combining algorithms (combine1(), combine2(), combine3(), combine4(), combine5x2(), combine5x3(), combine6())

Speedup of combine algorithms compared to combine1()

Y-axis: Speedup factor
X-axis: Combine algorithm (combine2(), combine3(), combine4(), combine5x2(), combine5x3(), combine6())