CSC 656-01 Coding Assignment #3
Spring 2023
Name: Rutu Patel
SFID: 920073203

Analysis questions.

Please provide a brief (2-3 sentences maximum) answer to each of the following questions about the performance of your codes. Please make use of the concepts we discuss in class and in the P&H textbook for full credit:

What types of operations are more expensive and why, and which of the codes is performing a larger number of more expensive operations?

Explanation: In general, memory operations are more expensive than arithmetic operations, because memory access involves accessing data from much slower memory devices compared to performing arithmetic operations on a processor's registers.

Looking at the provided code, the Indirect sum operation is likely to involve a larger number of more expensive memory operations, because it is accessing non-contiguous elements in the input array and requires more memory access operations than the Direct and Vector sum operations, which can perform operations on contiguous blocks of memory.

Computational rate. Which of the 3 methods has the best computational rate (MFLOP/s)? Why?

Explanation: The Vector sum method has the best computational rate (MFLOP/s) with a value of 7996.978828. This is because the vector sum method utilizes vector instructions which can perform operations on multiple data elements

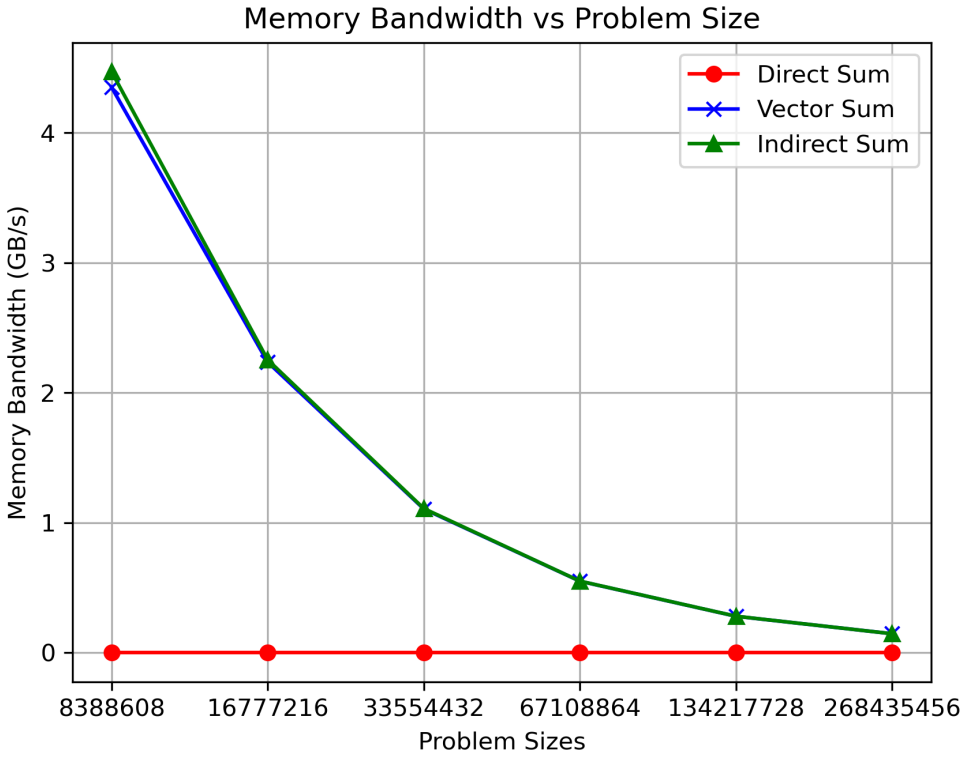
simultaneously, resulting in a faster computation compared to the direct sum and indirect sum methods which operate on data elements one at a time.

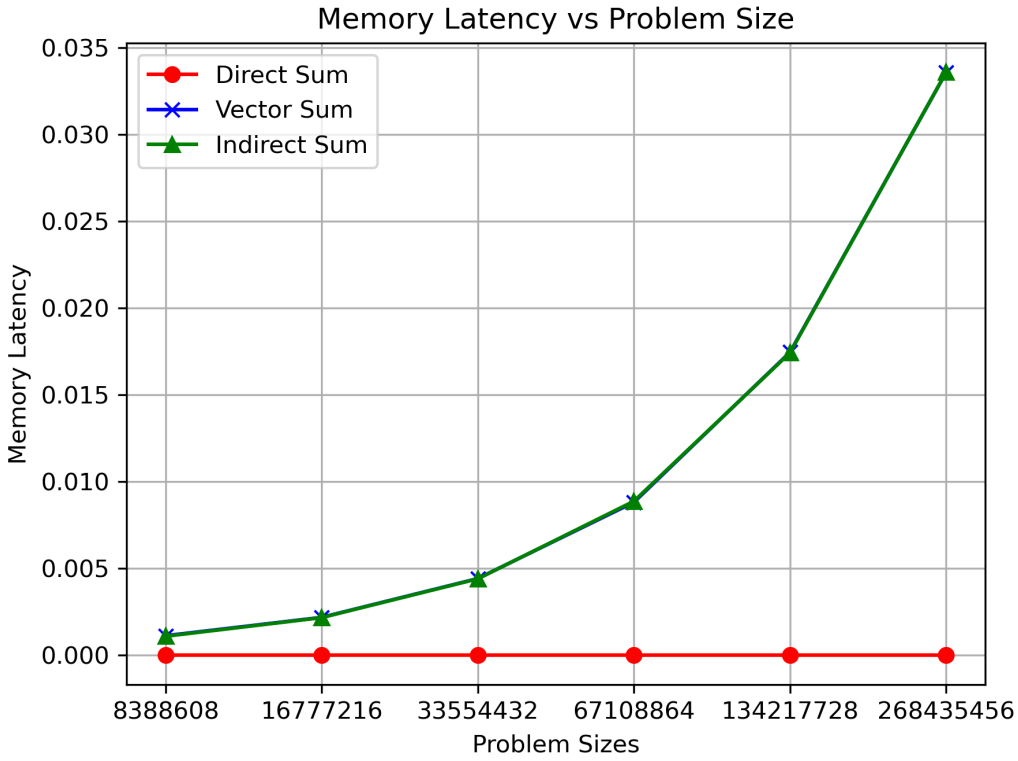
Memory bandwidth usage. Of the 2 methods vector sum and indirect sum, which has higher levels of memory bandwidth utilization? Why?

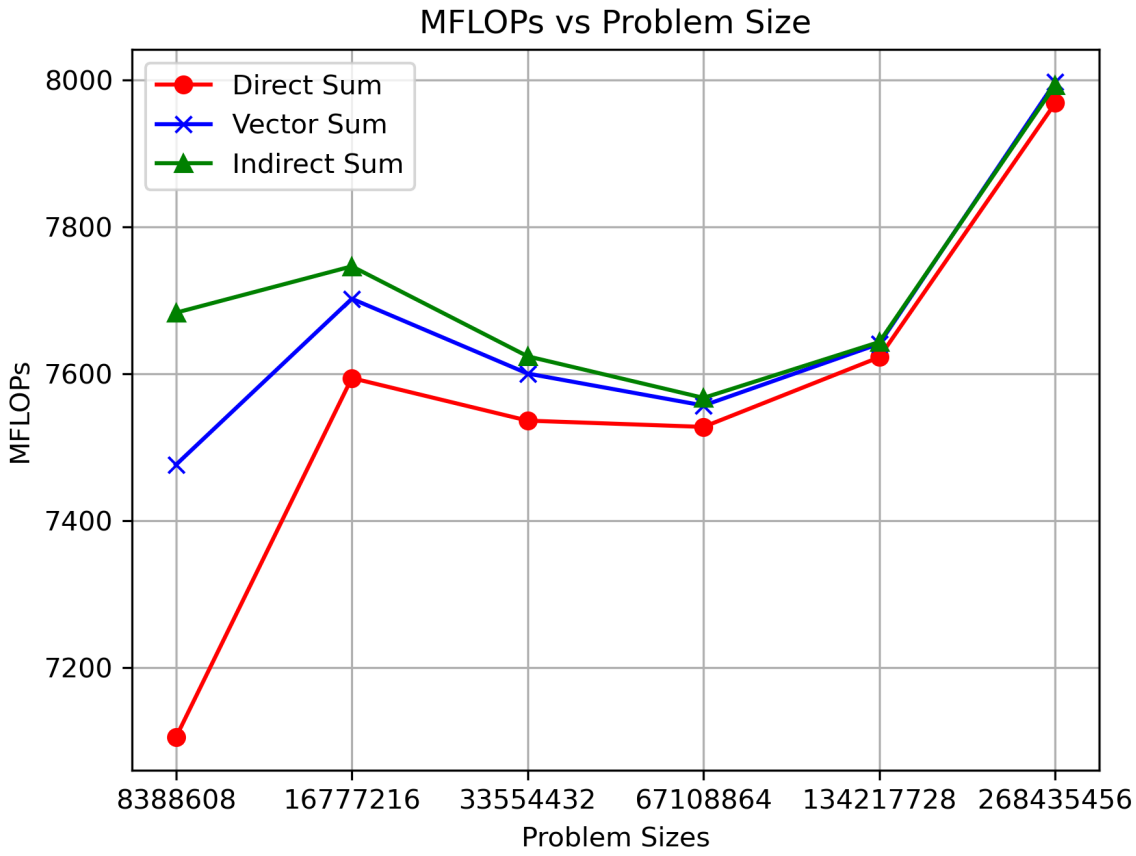
Explanation: The method with higher levels of memory bandwidth utilization is the vector sum method. This is because the vector sum method accesses memory sequentially and makes use of vector instructions that allow for parallelism in memory accesses. On the other hand, the indirect sum method accesses memory randomly, leading to a higher number of cache misses and lower memory bandwidth utilization.

Memory latency. Of the 2 methods vector sum and indirect sum, which shows lower levels of memory latency? Why?

Explanation: The method with lower levels of memory latency is the vector sum method. This is because the vector sum method uses contiguous memory accesses, which allow for efficient use of the CPU's memory hierarchy, including the cache. On the other hand, the indirect sum method accesses memory non-contiguously, which can result in cache misses and higher memory latency.


Memory Bandwidth vs Problem Size
Direct Sum
Vector Sum
Indirect Sum
Memory Bandwidth (GB/s)
0
1
2
3
4
8388608
16777216
33554432
67108864
134217728
268435456
Problem Sizes


Memory Latency vs Problem Size
Direct Sum
Vector Sum
Indirect Sum
Memory Latency
0.000
0.005
0.010
0.015
0.020
0.025
0.030
0.035
8388608
16777216
33554432
67108864
134217728
268435456
Problem Sizes


MFLOPs vs Problem Size
Direct Sum
Vector Sum
Indirect Sum
8000
7800
7600
7400
7200
MFLOPs
8388608
16777216
33554432
67108864
134217728
268435456
Problem Sizes