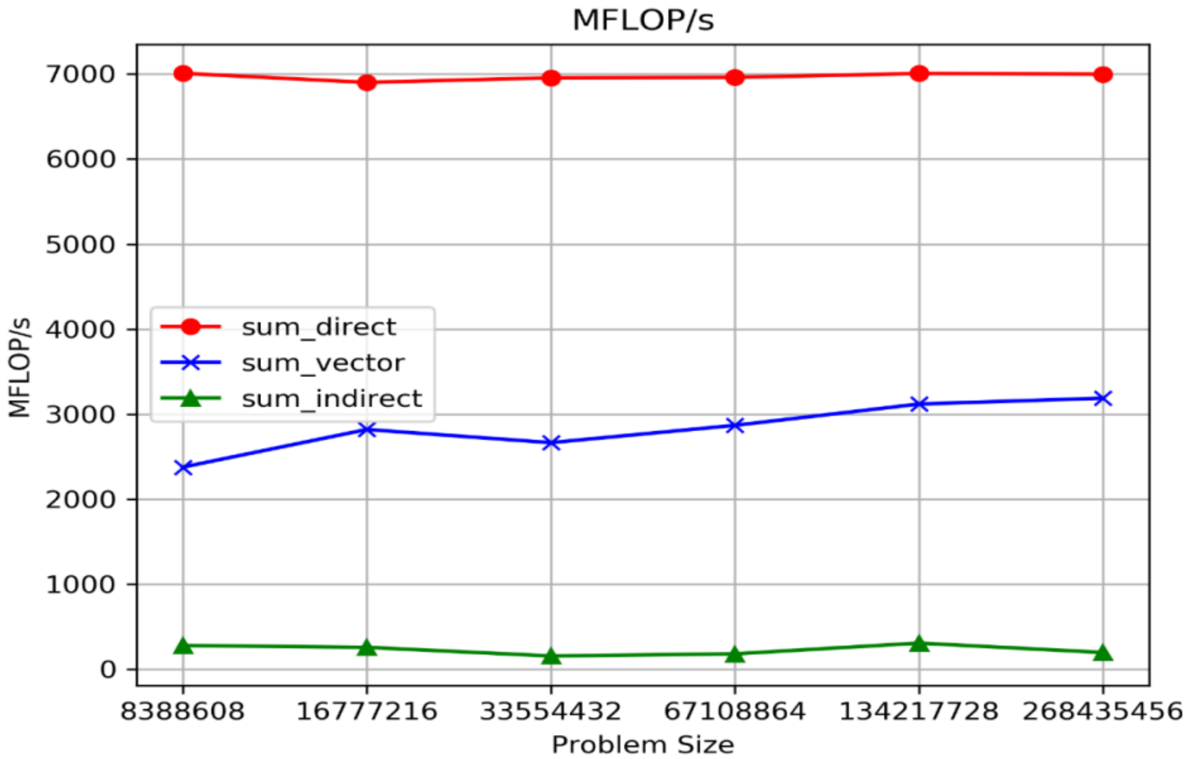
Nghi Nguyen

Oct 29 2024

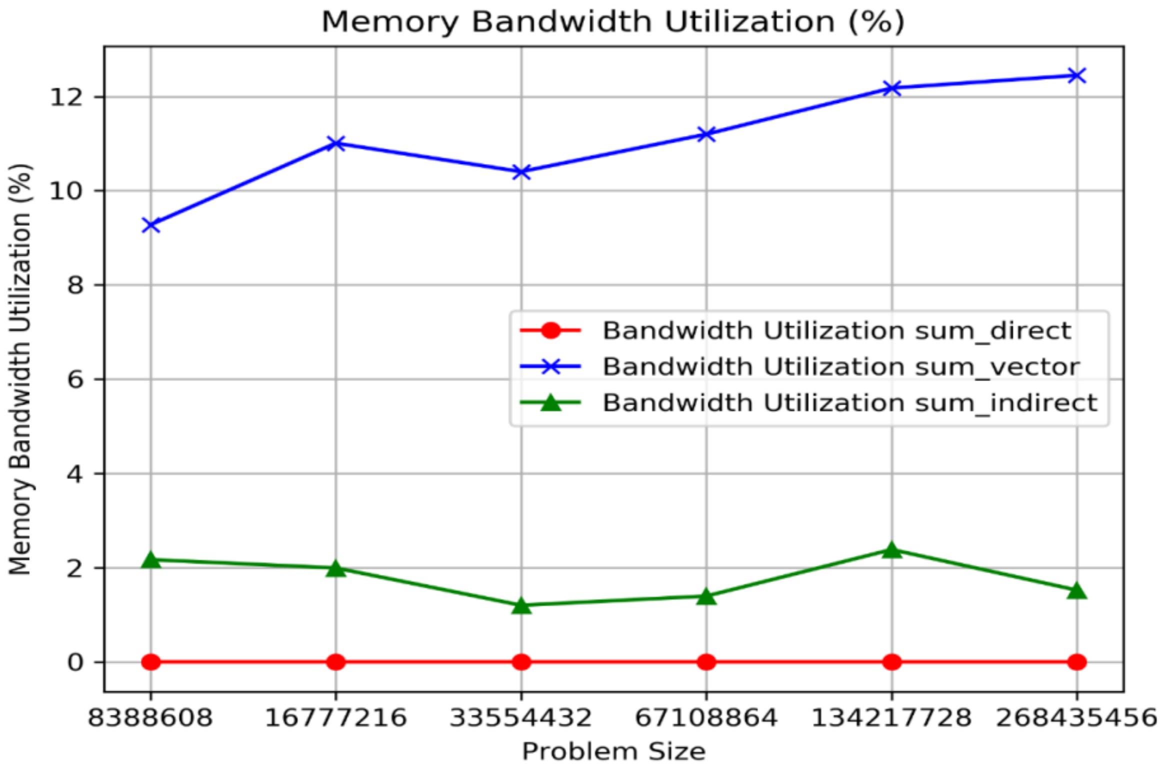
CSC 656.01

Coding Project #2 – Performance Analysis Fundamentals

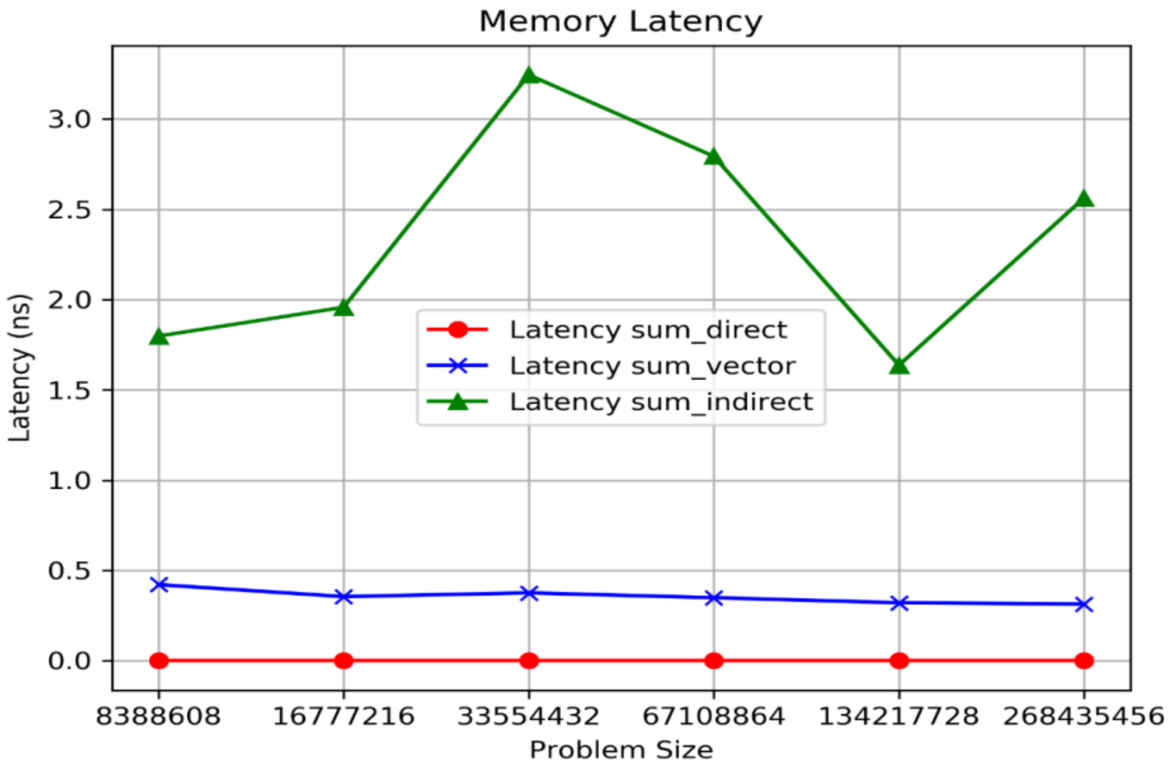
1. What types of operations are more expensive and why, and which of the codes is performing a larger number of more expensive operations?
    - Memory access operations are more expensive than arithmetic operations due to higher latency. Indirect sum is performing a larger number of more expensive operations due to the additional memory accesses needed for indirect indexing.



2. Computational rate. Which of the 3 methods has the best computational rate (MFLOP/s)? Why?
    - The method with the best computational is sum_direct. This is because it involves fewer memory accesses and simple arithmetic operations so its performance is not dragged down by memory latency.



3. Memory bandwidth usage. Of the 2 methods vector sum and indirect sum, which has higher levels of memory bandwidth utilization? Why?
    - The vector sum method has a higher level of memory bandwidth utilization compared to the indirect sum method. This is because the vector sum involves sequential memory accesses result in better utilization of memory bandwidth



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

    - The indirect sum method shows higher levels of memory latency compared to the vector sum method. Indirect accesses can result in more cache misses and delays, while the vector sum's contiguous accesses benefit from cache locality, reducing latency.