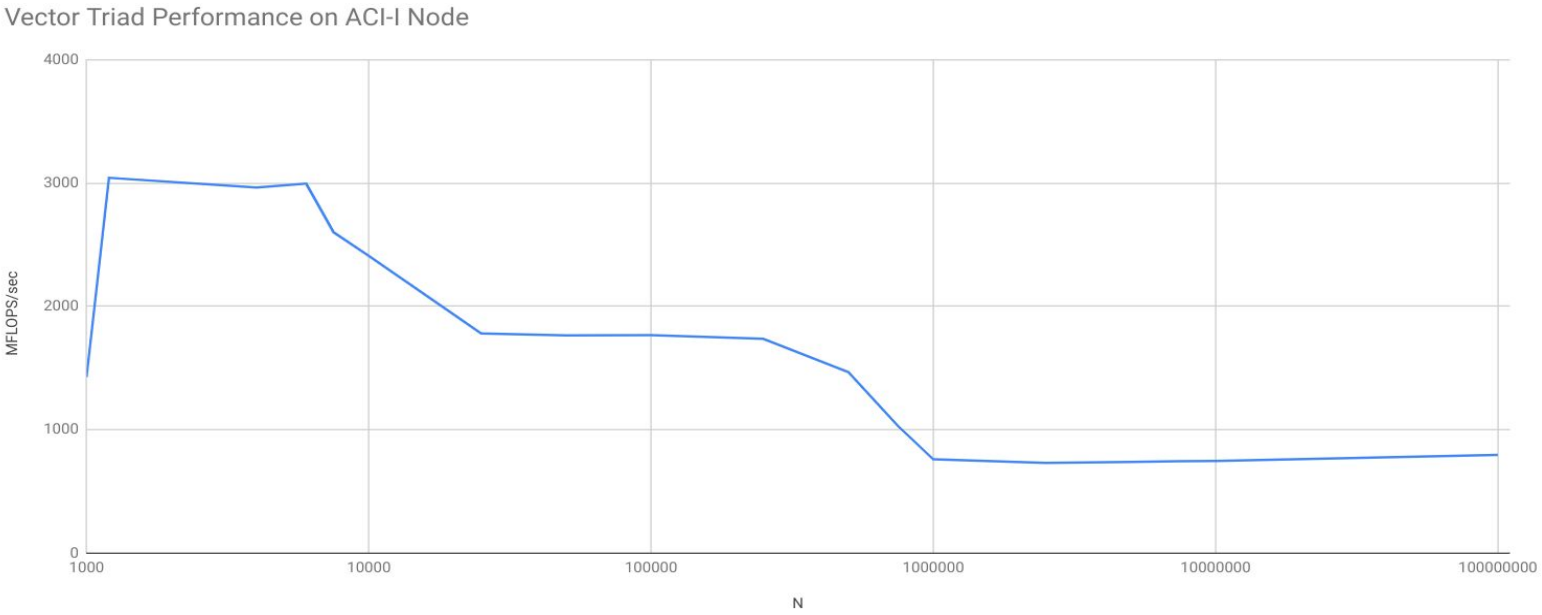
Brian Nguyen

01/21/2019

CMPSC 450

Homework 1 - Memory Read Benchmark



The purpose of the memory read benchmark assignment is to understand how computer performance (mega floating-point operations per second) is affected by varying amounts of data and iterations (N and R) from the vector triad program. By testing different combinations of N and R values, we can see how the hardware of the clusters react.

We mainly focus on the relationship between N and MFLOPs because we want to see the effect of increasingly big data on performance. N represents the size of the four arrays, which means the larger the value of N, the larger amount of bytes that are allocated, amount of

operations, and amount of stores and loads. These things and their optimizations are essentially the primary focus of this course, so learning how they translate to performance is the objective.

The reason for a need to change values of R is to have it act as a stopwatch. For example, for the lower amounts of N, iterations of the vector triad program would take microseconds. This could lead to hard-to-measure results after calculating MFLOPs since it is dependent on time spent. Therefore, by adjusting R, we can make the benchmark run for longer in an attempt to get a more accurate result. In my tests, I tried to make sure that all of the tests lasted for 0.5 seconds or greater.

I ran the vector triad program on both the ACI-i and ACI-b nodes for my benchmark results using a single core. In my tests, I chose 18 different values of N ranging between 1,000 to 100,000,000. For very small loop lengths, such as my first data point of N=1000, poor performance can be seen. However, as loop size increases after that, performance sees its highest points with MFLOPs of around 3000. After those peaks, performance decreases for all other data points. It then begins to converge to around 700-800 MFLOPs from N=1,000,000 to N=100,000,000. The reason for these drops in performance is due to the change in use of different types of memory, starting with registers down to the use of main memory. The different types of memory down the chain become larger, but much slower. For example, my assumption for the sudden drop in performance around N=1,000,000 is that main memory had to be used. This type of memory is much slower than what was being used for around N=5,000, which I would think is L1 cache. Therefore, in the middle portion of the graph between the two drops, I believe L2 cache is being used. From trying extraordinarily high values of N such as over one billion, I was unable to get a proper MFLOPs calculation, most likely due to the full use of main memory.