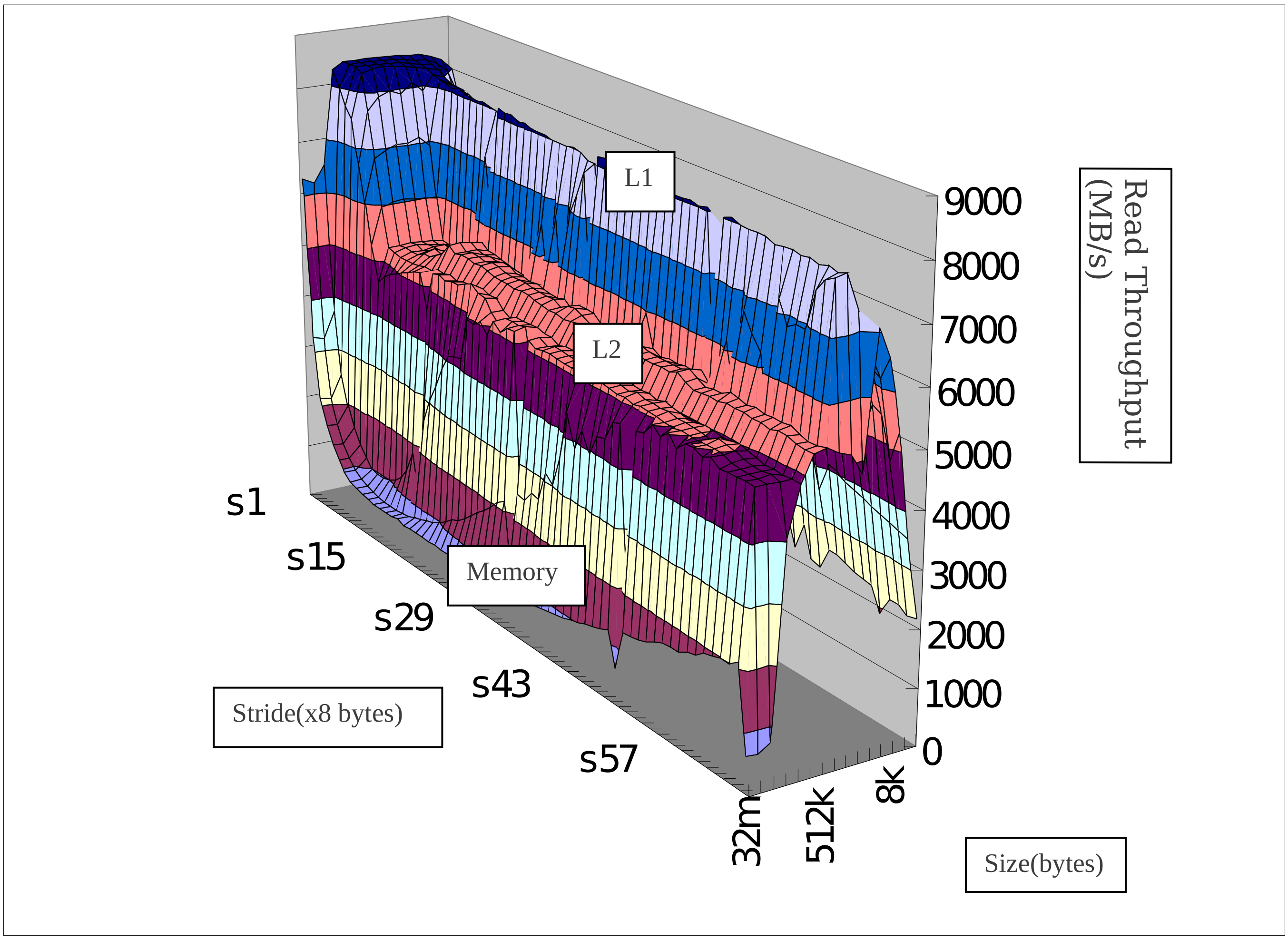
Memory Mountain Lab Report

Xiaohui Chen
xc2388



This is the lab report for memory mountain lab. The program code for the lab is shown on page 622 on the textbook. This program uses double arrays with different sizes and different strides to figure out the overall Read Throughput. In fact, the Read Throughput is a two-variable function of Size and Stride. When the program executes, it automatically prints out the value of Read Throughput with respect to specific Size and Stride Value. I will analyze the memory mountain graph thoroughly in this report.

This is the memory mountain graph according to the data from the lab. This 3D graph has 3 axes: Stride, Size and Read Throughput. When the size of data is fixed, the graph of Stride to Read Throughput indicates how spatial locality affects overall performance of CPU. While the Stride is fixed, the relationship between Size and Read Throughput reveals how temporal locality affects. Therefore, this graph shows the relationship between the performance to temporal and spatial locality.

The CPU of the computer I use is Intel Core 2 Duo E8500. It has 32kb L1 data cache and 6144kb L2 cache. The label of each cache can be seen in the graph. As a matter of fact, this 3D graph clearly indicates that L1 cache is faster than L2 cache, which is faster than memory.

The 3D graph shows that when the size of data is fixed, the wider of stride, the slower of the speed. The graph clearly shows how spatial locality increases the overall performance. This phenomenon is most obvious when size is huge because L1, L2 and memory are used with such huge data size. Consequently the overall performance is minimized. Note that for L1 and right part of L2 graph, the Stride-Read Throughput graph is nearly a horizontal line. This is because for small data sizes, only L1 is used. In the right part of L2, the data going to read is in L2 and so the effects of spatial locality are minimized.

When Stride is fixed, the Size-Read Throughput graph proves the effects of temporal locality on overall performance. As ridges of temporal locality shows, the increase in data sizes minimizes the overall performance. In sizes near 32kb and 6mb, the ridges of temporal locality drops down rapidly because at those values, the old data storage is full and a new cache or memory has to be used. Because of the effect of header, when data size is less than 8kb, the overall performance increases when data size increases. In general, the increase of data sizes decreases temporal locality and thus minimizes the overall performance.

This 3D graph shows how temporal locality and spatial locality affect the performance of the processor. Undoubtedly, programs with highest spatial and temporal locality run fastest. This rule clearly gives guideline to programmers. Program performance therefore is a curtail factor of computer programs, especially those with enormous data managements.