Alexandria University
Faculty of Engineering

Computer Organization
Fall 2023
Assigned: 16/12/2023
Due: 23/12/2023

Matrix Multiplication vs. Cache Optimization

Introduction

Matrix multiplication is a fundamental operation in linear algebra and finds widespread application in various domains, including computer graphics, scientific computing, and machine learning. However, the performance of matrix multiplication can be significantly impacted by the behavior of the computer's cache memory hierarchy. Understanding how cache works and employing efficient coding techniques that exploit cache locality and cache blocking can greatly enhance the efficiency of matrix multiplication algorithms.

Cache memory is a small but fast memory unit located closer to the CPU than the main memory. It serves as a temporary storage space for frequently accessed data, allowing the CPU to access it much faster than retrieving data from the main memory. The cache operates on the principle of locality, which refers to the tendency of programs to access data that is spatially or temporally close to each other.

Matrix multiplication involves accessing elements from multiple matrices and performing a series of multiplications and additions. If the matrices are large, the data access pattern can exhibit poor locality, leading to frequent cache misses. Cache misses occur when the requested data is not found in the cache, requiring the CPU to fetch it from the main memory, incurring a significant performance penalty.

To mitigate the impact of cache misses on matrix multiplication,techniques are employed that optimize cache utilization. One such technique is cache blocking. This technique involves dividing the matrix multiplication into smaller blocks and performing computations on these blocks sequentially, taking advantage of the cache's limited size.

By operating on smaller blocks that fit into the cache, cache blocking exploits spatial locality, ensuring that the required data stays in the cache for subsequent operations. This reduces the frequency of cache misses as the algorithm accesses memory in a more predictable and efficient manner. Additionally, cache blocking promotes data reuse, as elements within a block are reused multiple times before being evicted from the cache.

Furthermore, optimizing the order in which matrix elements are accessed can also improve cache efficiency. This technique, known as loop interchange, rearranges the nested loops in the matrix multiplication algorithm to traverse the matrices in a manner that maximizes data locality. By accessing elements in a sequential and contiguous manner, loop interchange increases the likelihood of cache hits and reduces unnecessary cache evictions.

Experiment

For this experiment, you are required to plot a chart showing the relation between matrix sizes and block sizes when using a blocking multiplication algorithm. Run the matrix multiplication and calculate the time it took. Plot the time on the y-axis and block size on the x-axis. Repeat for the `m` sizes. Plot all the results on the same graph.

Note that when block size is 1 the blocking multiplication and normal multiplication are both the same

Submission

You should submit a report showing the obtained results. You should draw charts that signifies how blocking affects the performance of the matrix multiplication.