

Program Comparison

In this comparison, we compared the run times of two different programs with similar algorithms. The programs are implementations of matrix multiplication with the key difference being the indexing of the array and the array variables. In terms of time complexity, both programs have a n^2 run time, which is common for matrix multiplication. The primary performance difference is that one program takes advantage of the cache to speed up memory accesses and the other program purposefully avoid the benefits of the cache.

In more detail the following indexings are embedded within nested loops:

Program_0 = > **$y[i] += A[i*size + j] * x[j];$**

Program_1 => **$y[j] += A[j*size + i] * x[i];$**

In terms of locality, Program_0 has temporal, spacial, and spacial for the different parts of the statement and Program_1 has spacial, none, and temporal respectively. This indicated that Program_0 better utilizes the cache by implementing sequential memory accesses, which cache memory is made to optimize. The important piece of code for Program_1 is **$A[j*size + i]$** , which changes by the size variable, avoiding the spacial locality, whereas Program_0 is **$A[i*size + j]$** , which changes by 1 per iteration, making better use of the cache. To visualize the comparison, we compared run times based on the size of the matrix. Below is a chart illustrating that although both programs have the same run time complexity, Program_1 increases at a faster rate.

