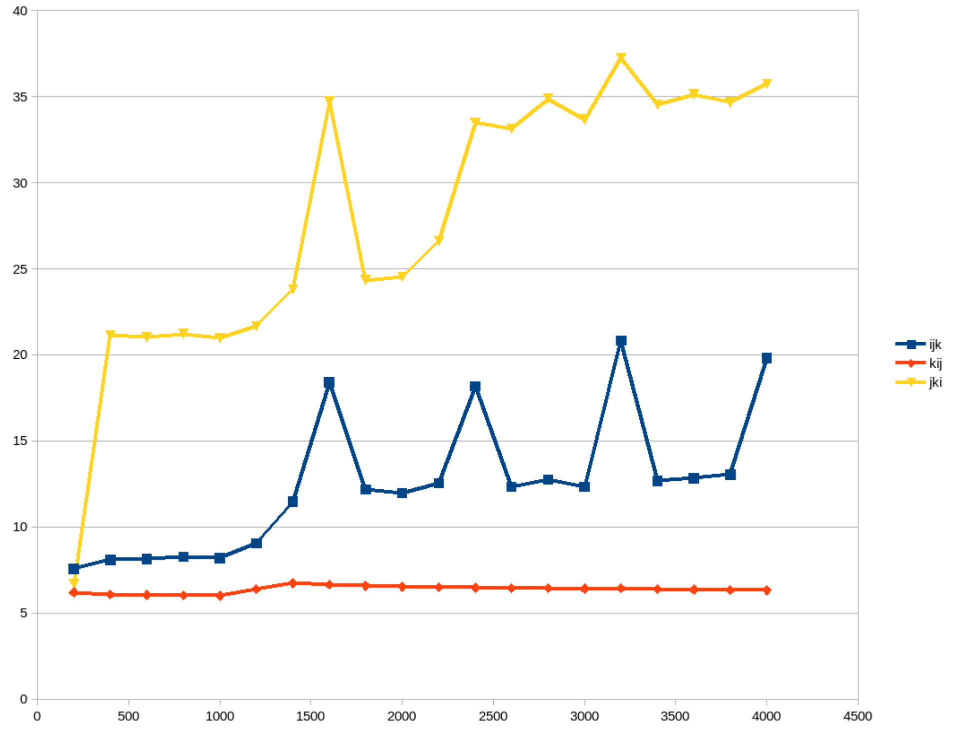
Anirudh Sriram
Collaborator: Anand Sanmukhani

Anirudh Sriram

Collaborator: Anand Sanmukhani

Task 1

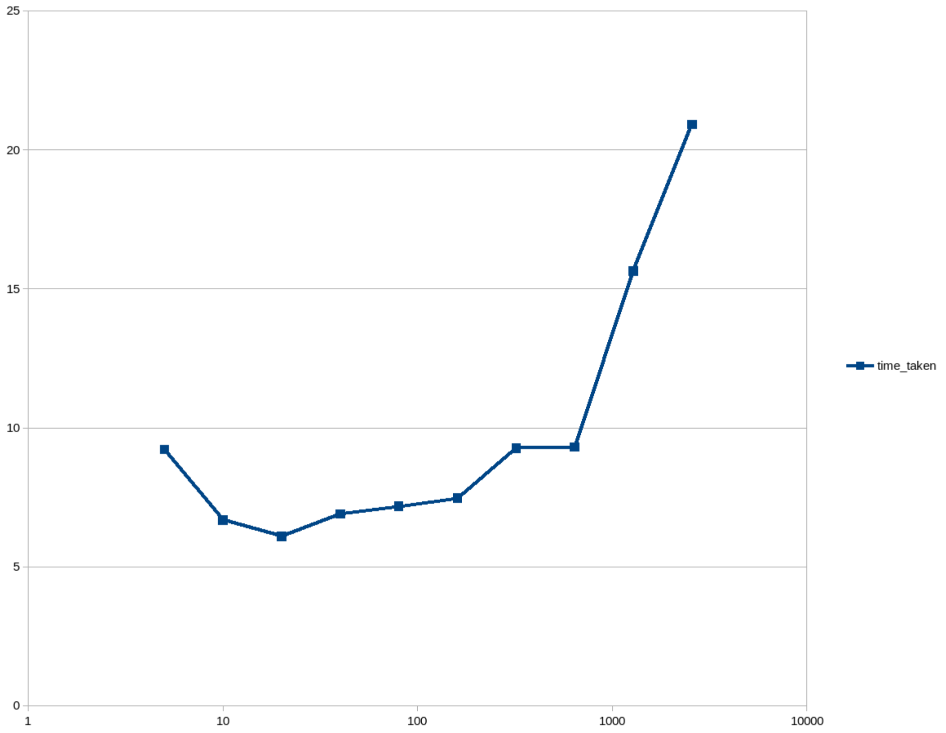
As seen in the slides, the most efficient version of MMM was kij, followed by ijk, and then the worst performance was for ikj, An important transition for the kij graph is when length increases from 1000, where it can be seen that the efficiency goes down. An important transition for graph ijk is also when length increases from 1000, as efficiency can be seen to go down. Since ikj, takes all vertical reads, it has the worst performance.



Task 2

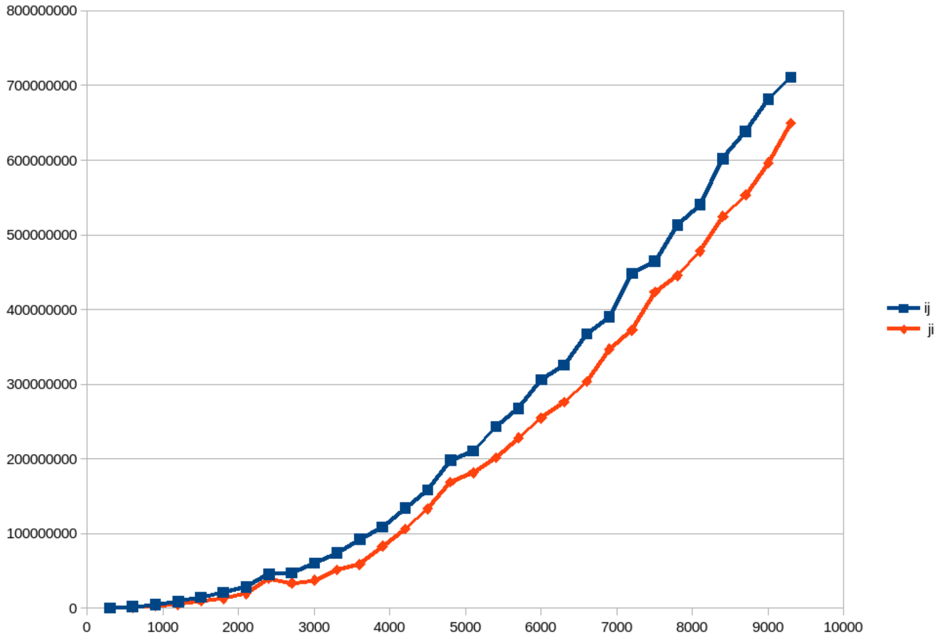
Using the given blocked code, we do matrix multiplication. We took matrix size to be 2400, and varied the block size to check the time. We found that the peak efficiency is achieved when block size is 20, after which the efficiency starts to decrease.

Anirudh Sriram
Collaborator: Anand Sanmukhani



Task 3_1

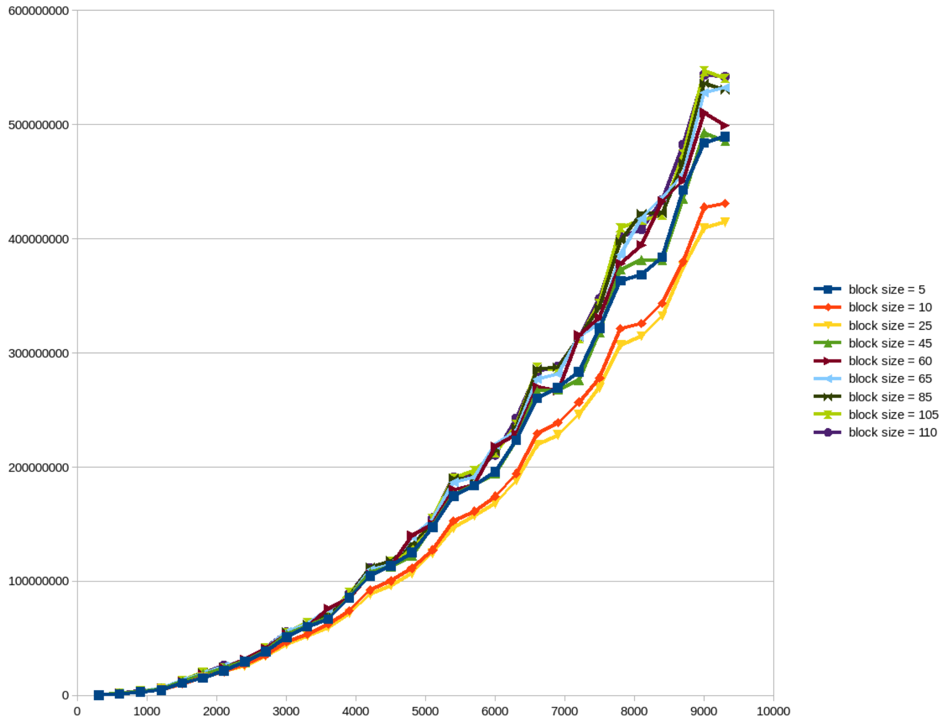
For transpose we found ji to be slower than ij, since ij takes in horizontal reads, whereras ji takes vertical reads.



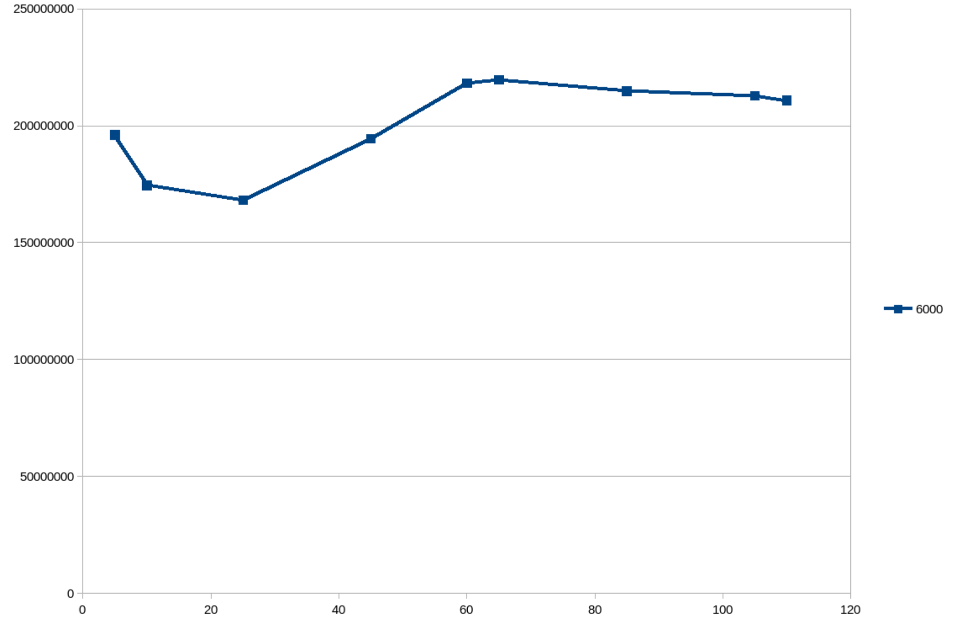
Anirudh Sriram
Collaborator: Anand Sanmukhani

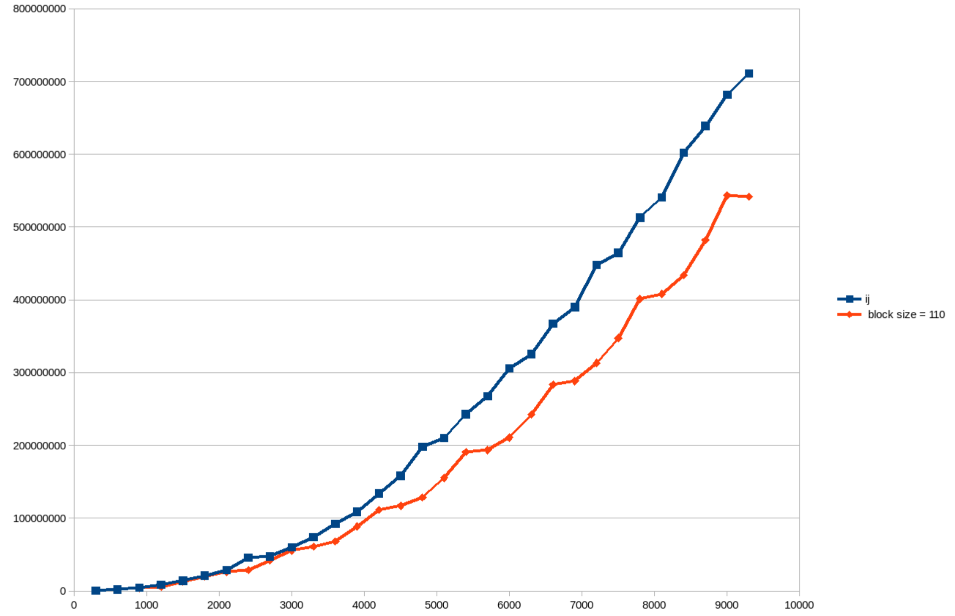
Task 3_2

For the blocked version of the code we implemented by working on smaller block sizes , inside the larger block sizes. We transposed the relevant elements inside the smaller blocks. For the elements which had to be transposed with elements outside their blocks, we manually transposed them. The results can be seen for various block sizes. We also focused on one matrix size of 6000 , to focus more closely on the variation of time with block size. As it can be seen after about a block size of 20, for a matrix size of 6000, the efficiency starts to go down. The blocking version is more efficient than the non-blocking version, since in the blocking version, once a particular block is used, it is not referenced again, whereas in the non-blocking version, the same elements may be brought into the cache multiple times. The third graph shows the difference in performance between blocking and non-blocking versions.



Anirudh Sriram
Collaborator: Anand Sanmukhani


250000000
200000000
150000000
100000000
50000000
0
0
20
40
60
80
100
120
6000


ij
block size = 110