CS6868: Concurrent Programming
Spring 2014
Assignment 1: Due 22 February 2014, 11:59 pm

Problem 1:Square of Square Matrix

Approaches Used

Iterative Program.

This approach is similar to classical Matrix multiplication. Only difference sequential one is outermost for loop is paralleled

Recursive Block Divide and Conquer

In this method,We take a matrix and divide into 4 blocks. Each block is recursively multiplied and final result is then calculated from partial products by adding. The recursive process continue till block size reduced below some threshold (In implementation threshold = 8x8)

Using Transpose

This approach uses the row major representation of multidimensional array in C/C++.By taking transpose of 2 matrix (here same matrix),we can convert matrix into column major order.

$$A[i,j] = \sum_k A[i,k]*A[k,j]$$

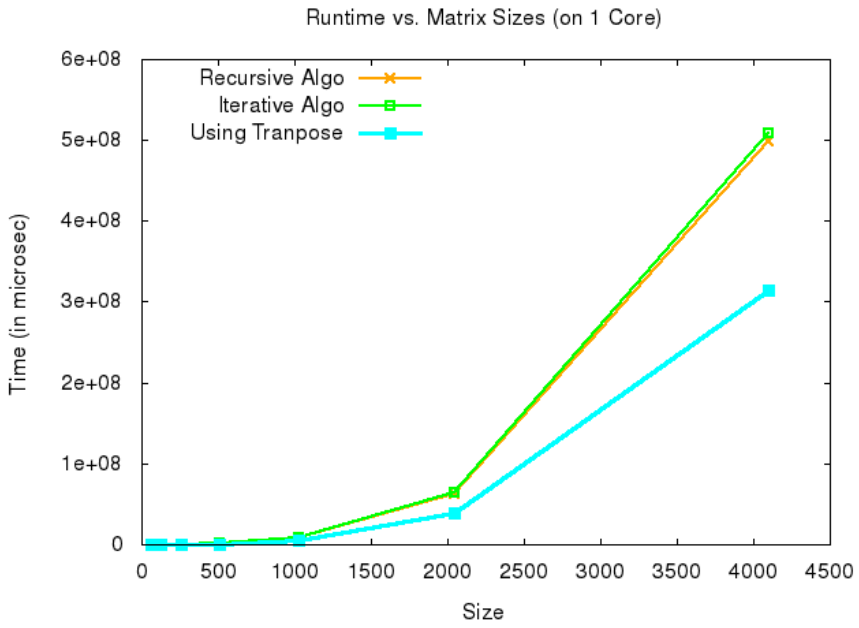
now become

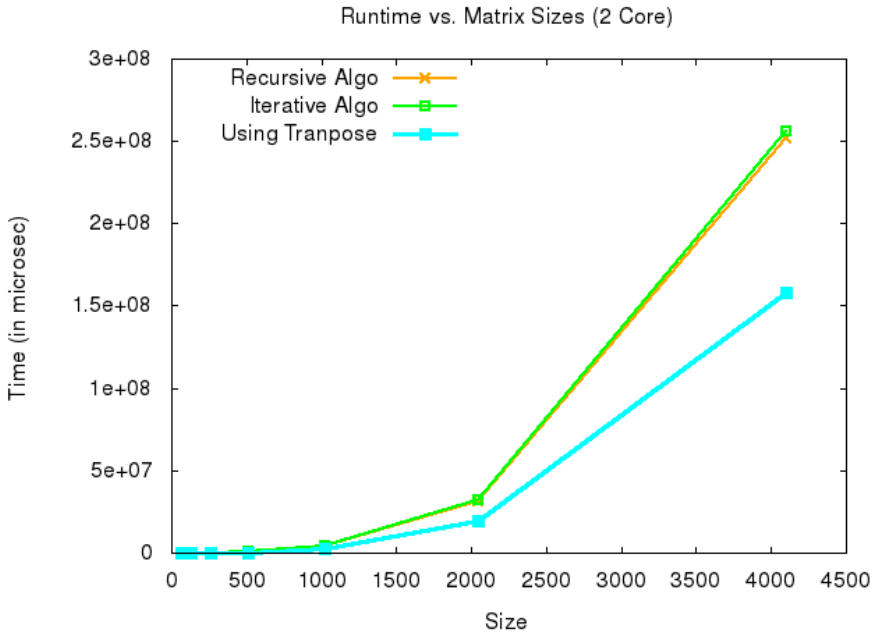
$$A[i,j] = \sum_k A[i,k]*A^T[j,k]$$

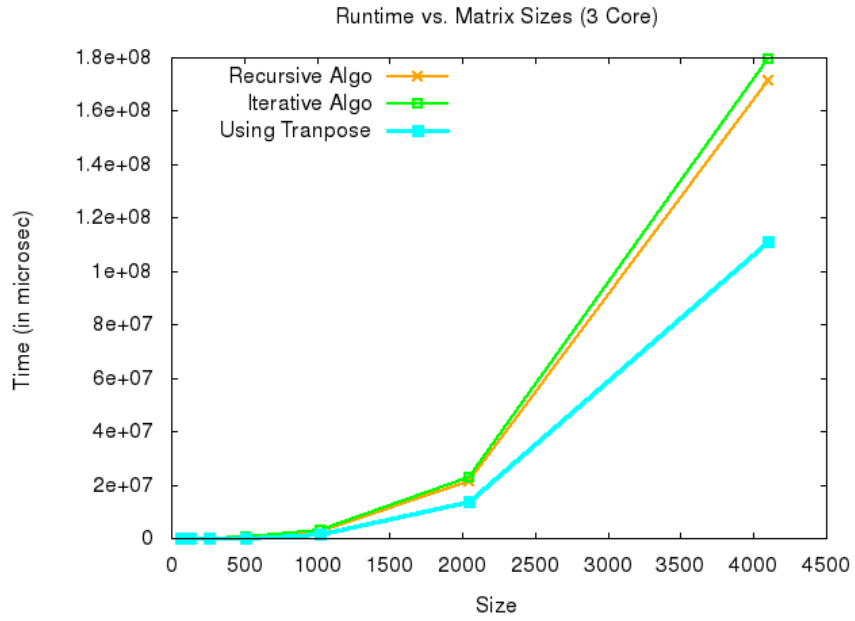
which increase cache locality.

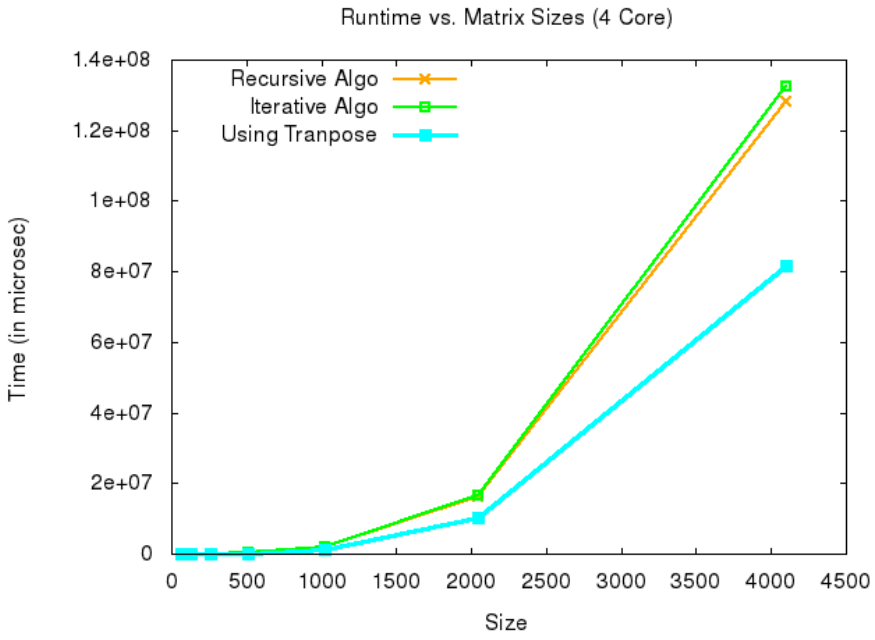
Comparison of Different Methods

Best Approach is the 3rd Approach(Using Transpose) followed by Iterative Approach and Recursive Approach respectively.

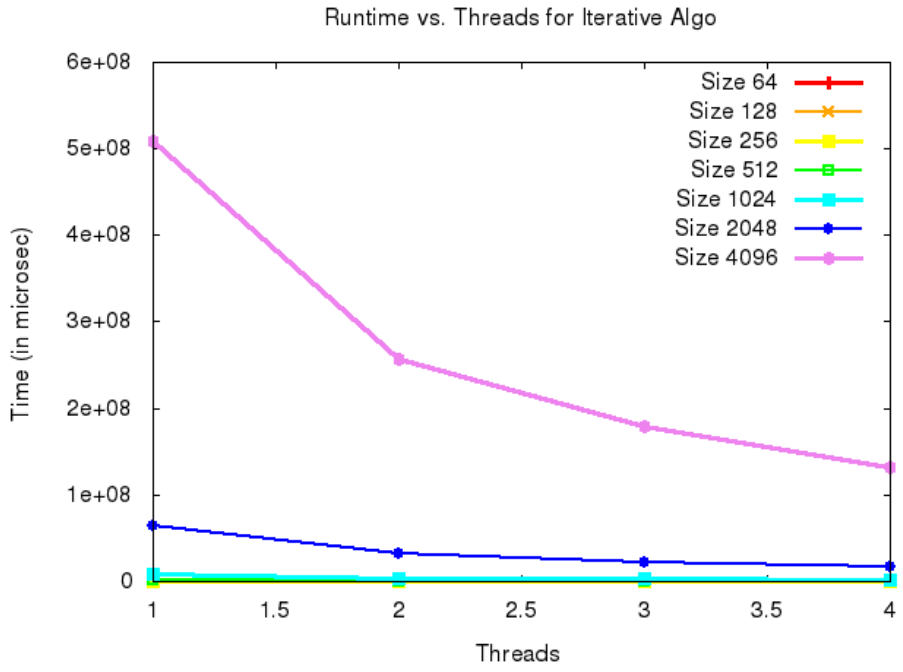


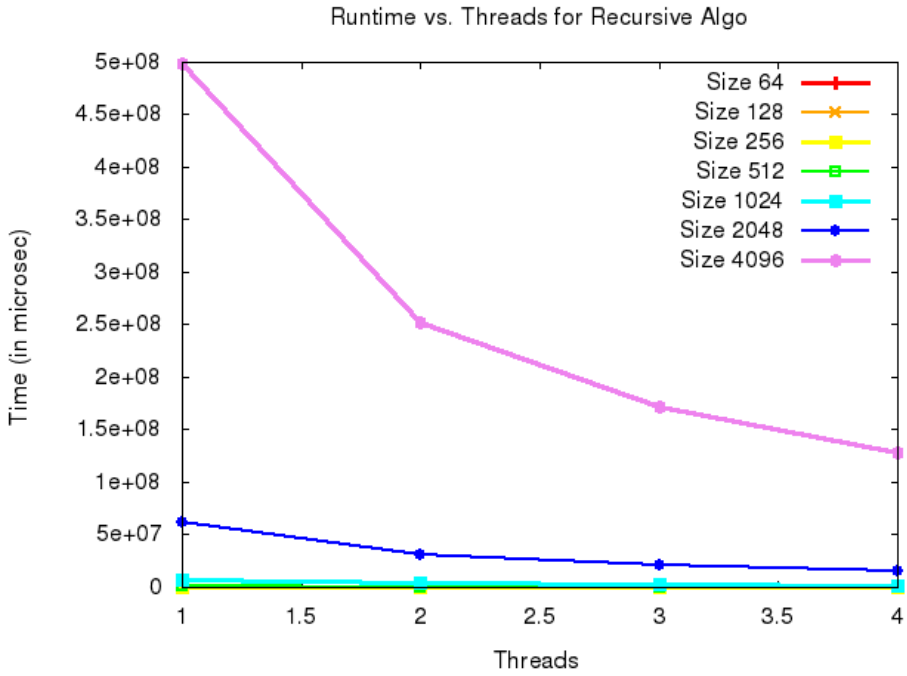


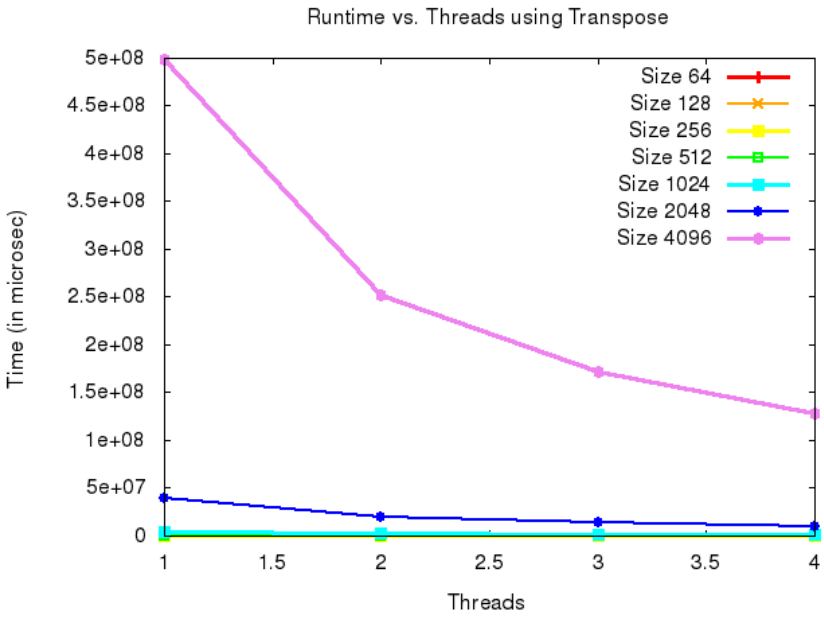

Runtime vs. Matrix Sizes (3 Core)
Recursive Algo
Iterative Algo
Using Tranpose
Time (in microsec)
Size
1.8e+08
1.6e+08
1.4e+08
1.2e+08
1e+08
8e+07
6e+07
4e+07
2e+07
0
0
500
1000
1500
2000
2500
3000
3500
4000
4500


Runtime vs. Matrix Sizes (4 Core)
Recursive Algo
Iterative Algo
Using Tranpose
Time (in microsec)
Size
1.4e+08
1.2e+08
1e+08
8e+07
6e+07
4e+07
2e+07
0
0
500
1000
1500
2000
2500
3000
3500
4000
4500

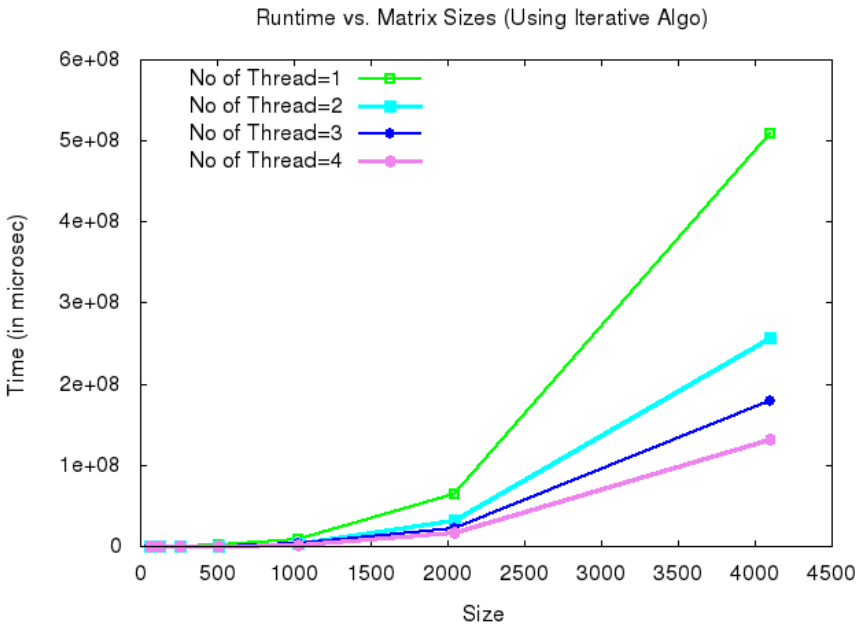
Comparison of Running Time Vs Threads

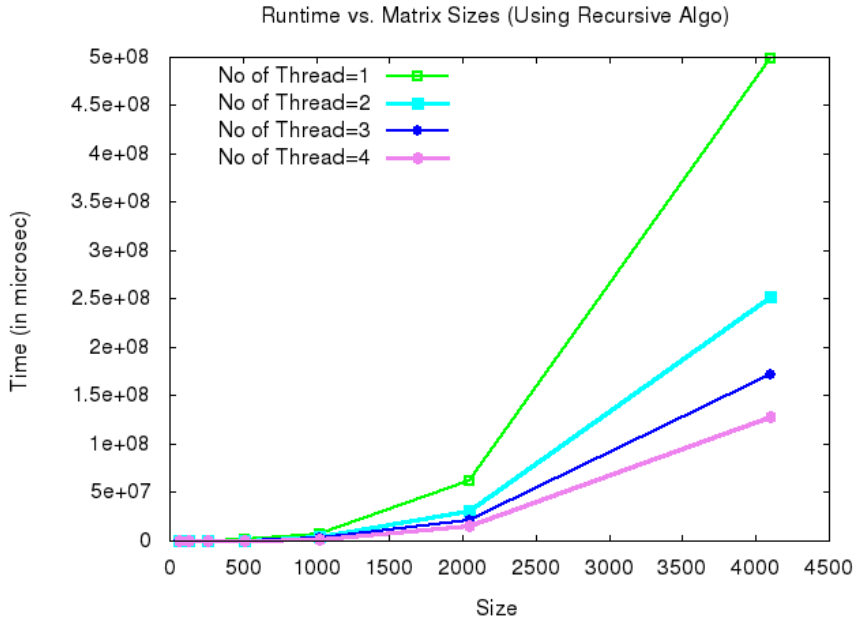


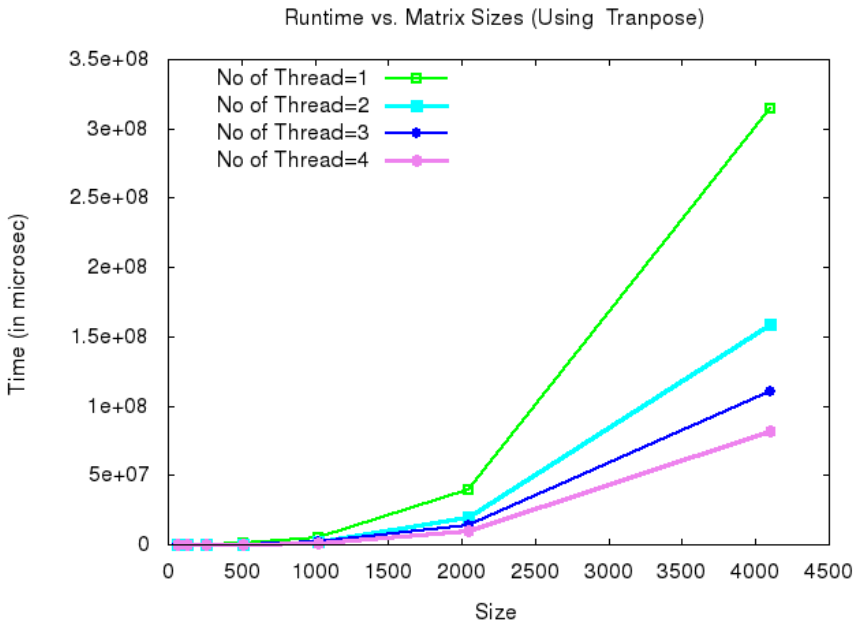




Comparison of Running Time Vs Threads




Runtime vs. Matrix Sizes (Using Recursive Algo)
No of Thread=1
No of Thread=2
No of Thread=3
No of Thread=4
Time (in microsec)
Size
5e+08
4.5e+08
4e+08
3.5e+08
3e+08
2.5e+08
2e+08
1.5e+08
1e+08
5e+07
0
0
500
1000
1500
2000
2500
3000
3500
4000
4500


Runtime vs. Matrix Sizes (Using Tranpose)
No of Thread=1
No of Thread=2
No of Thread=3
No of Thread=4
Time (in microsec)
Size
3.5e+08
3e+08
2.5e+08
2e+08
1.5e+08
1e+08
5e+07
0
0
500
1000
1500
2000
2500
3000
3500
4000
4500

Observation and Conclusion

- ➢ Running time of 64,128,256 are almost constant with respect to no of threads. ie,For small size input there is almost no parallelism.
- ➢ Recursive Program is much slower than iterative one.
- ➢ Even though no of operation is Approach 1 (*Iterative)* and Approach 3 *(using transpose)* are same,Approach 3 is faster due to better cache locality.
- ➢ Speedup Estimate of *recursive* program and one using *transpose* is same (128 processors: 121.60 - 128.00, 256 processors: 243.20 - 256.00) and slightly more than *Iterative* for higher number of processors (128 processors: 121.59 - 128.00, 256 processors: 231.51 - 256.00) [for input size 4096]