Minor Project (Open MP)

Nimoshika Jayaraman

UIN: 631000852

1. The C++ implementation in Rinverse.cpp is parallelized using the strategies mentioned.

```
module load intel
icc -qopenmp -o Rinverse.exe Rinverse.cpp
./Rinverse.exe 10 7 4
```

and experimented with different values.

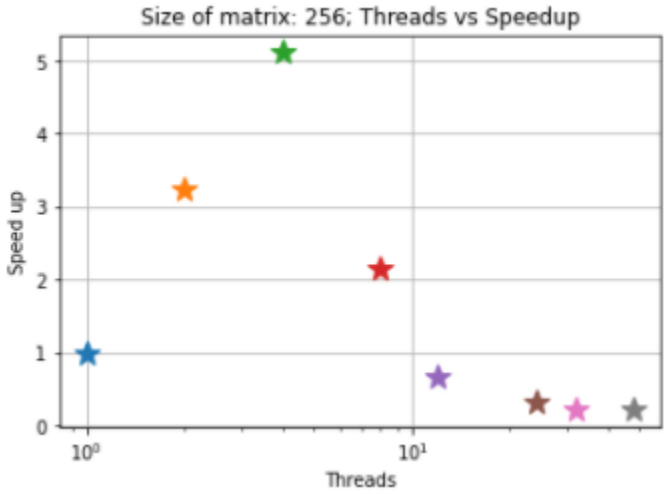
```
sbatch Rinverse.grace_job.sh
```

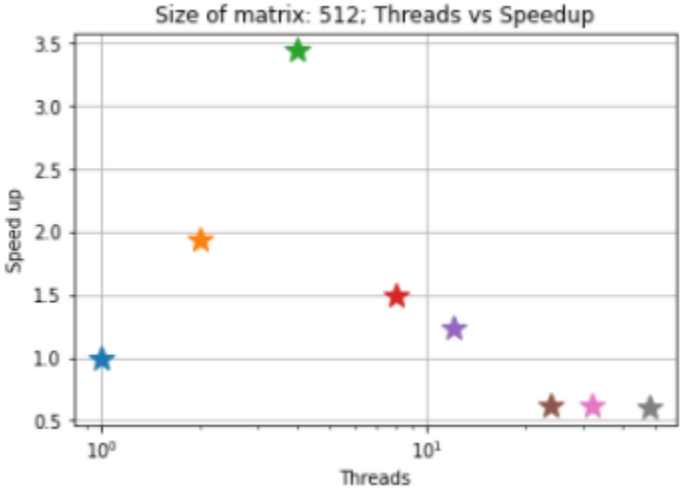
snippet:

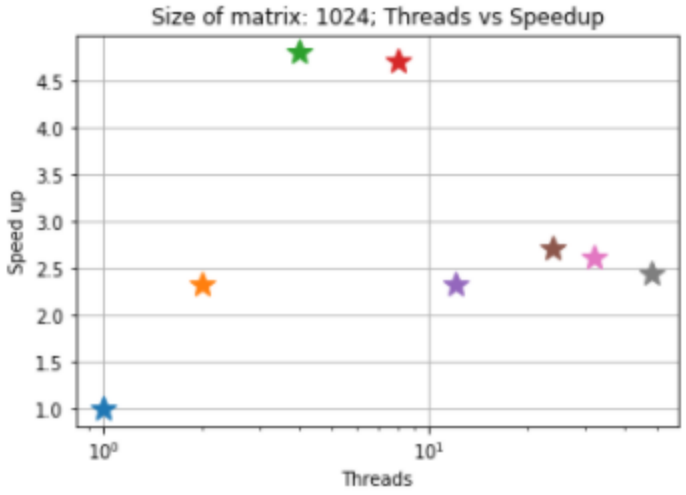
```
[nimoshika@grace1 ~]$ cd Minor-Project-735
[nimoshika@grace1 Minor-Project-735]$ module load intel
[nimoshika@grace1 Minor-Project-735]$ ..
[nimoshika@grace1 Minor-Project-735]$ cd ..
[nimoshika@grace1 ~]$ ls
HW1-735  HW2-735  HW3-735  Minor-Project-735
[nimoshika@grace1 ~]$ cd Minor-Project-735
[nimoshika@grace1 Minor-Project-735]$ module load intel
[nimoshika@grace1 Minor-Project-735]$ icc -qopenmp -o Rinverse.exe Rinverse.cpp
icc -o sort_list.exe sort_list.c -lpth./sort_list.exe 12 4
-bash: ./sort_list.exe: No such file or directory
[nimoshika@grace1 Minor-Project-735]$ icc -qopenmp -o Rinverse.exe Rinverse.cpp
[nimoshika@grace1 Minor-Project-735]$ ./Rinverse.exe 10 7 4
Matrix Size = 1024,  Leaf Matrix Size = 128, Error = 0, Execution Time =   0.2419
[nimoshika@grace1 Minor-Project-735]$ ./Rinverse.exe 10 7 0
Matrix Size = 1024,  Leaf Matrix Size = 128, Error = 0, Execution Time =   0.9448
[nimoshika@grace1 Minor-Project-735]$ ./Rinverse.exe 10 7 5
Matrix Size = 1024,  Leaf Matrix Size = 128, Error = 0, Execution Time =   0.2220
[nimoshika@grace1 Minor-Project-735]$ ./Rinverse.exe 10 7 6
Matrix Size = 1024,  Leaf Matrix Size = 128, Error = 0, Execution Time =   0.1517
[nimoshika@grace1 Minor-Project-735]$ ./Rinverse.exe 10 7 5
Matrix Size = 1024,  Leaf Matrix Size = 128, Error = 0, Execution Time =   0.1715
[nimoshika@grace1 Minor-Project-735]$ ./Rinverse.exe 10 7 1
Matrix Size = 1024,  Leaf Matrix Size = 128, Error = 0, Execution Time =   0.9035
[nimoshika@grace1 Minor-Project-735]$ ./Rinverse.exe 10 7 32
Matrix Size = 1024,  Leaf Matrix Size = 128, Error = 0, Execution Time =   0.0828
[nimoshika@grace1 Minor-Project-735]$ ./Rinverse.exe 10 7 48
Matrix Size = 1024,  Leaf Matrix Size = 128, Error = 0, Execution Time =   0.1030
[nimoshika@grace1 Minor-Project-735]$ ./Rinverse.exe 10 7 20
Matrix Size = 1024,  Leaf Matrix Size = 128, Error = 0, Execution Time =   0.0744
[nimoshika@grace1 Minor-Project-735]$ ./Rinverse.exe 10 7 25
Matrix Size = 1024,  Leaf Matrix Size = 128, Error = 0, Execution Time =   0.0747
[nimoshika@grace1 Minor-Project-735]$ sbatch Rinverse.grace_job.sh
```

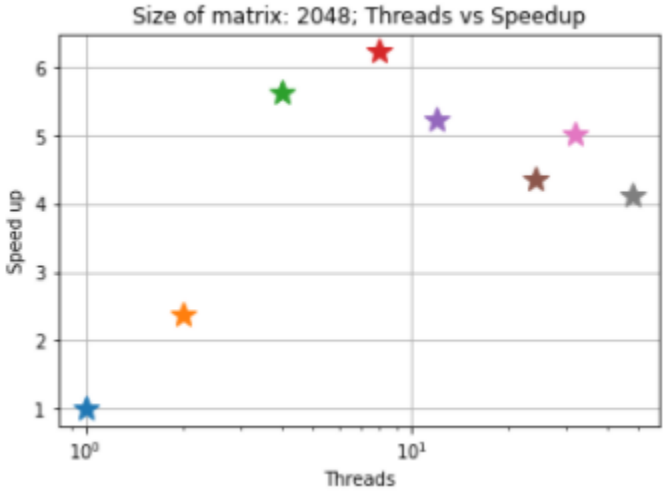
2. The speedup and efficiency obtained by the implemented routine on up to 48 processors is shown below:

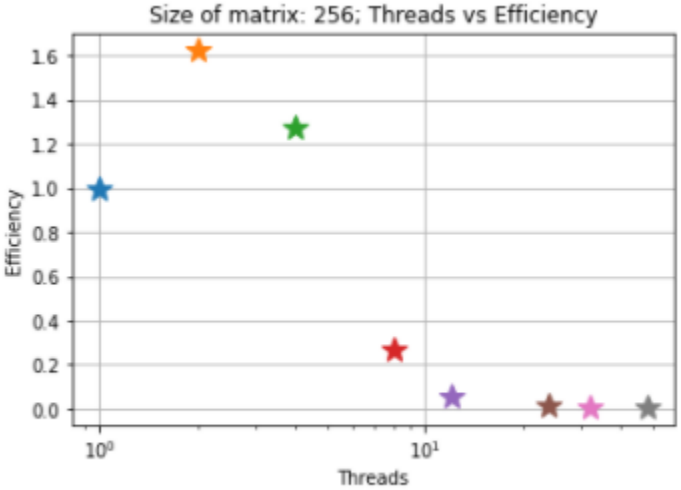
   Matrix Size: 256, 512, 1024 and 2048 is used to experiment with leaf matrix size of 2.

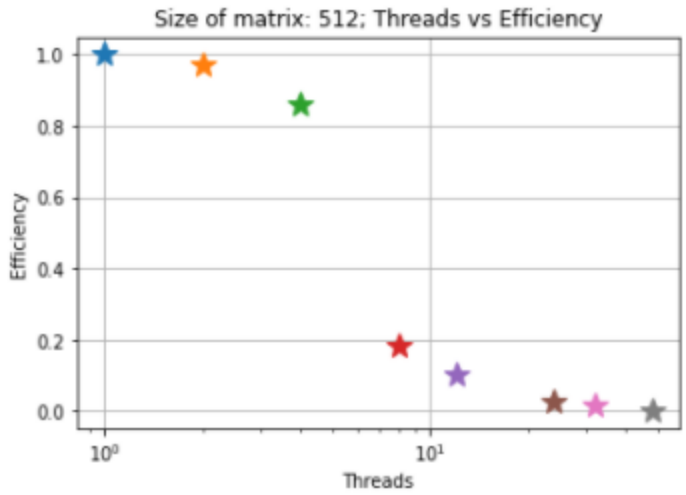


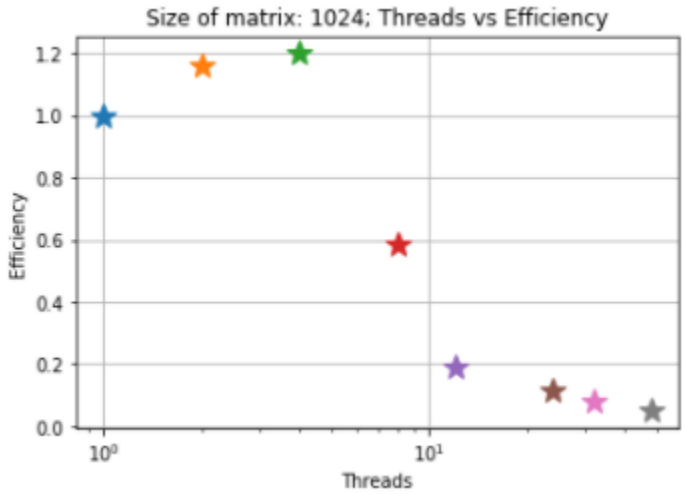


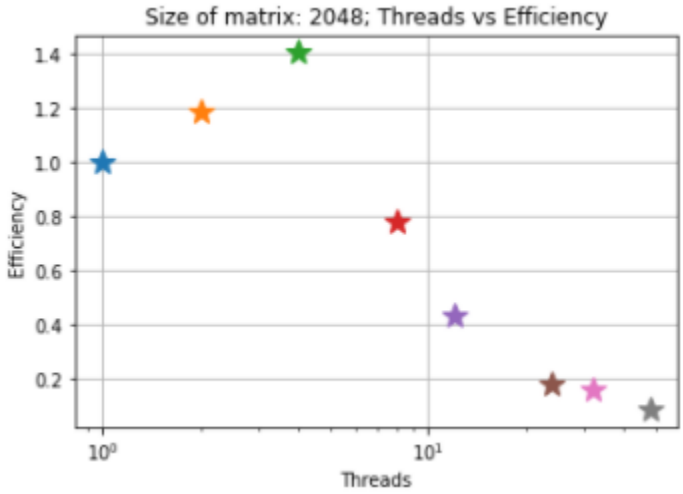

Size of matrix: 1024; Threads vs Speedup
Speed up
Threads
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
10^0
10^1


Size of matrix: 2048; Threads vs Speedup
Speed up
Threads
1
2
3
4
5
6
10^0
10^1


Size of matrix: 256; Threads vs Efficiency
Efficiency
Threads
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
10^0
10^1







3. Performance of the code was experimented using different values of Leaf_matrix_size. When the leaf matrix size is increased by 2,8

and 32, the execution time is decreasing. As it gets fine grained, the execution time is decreasing and makes the code work better but beyond certain value like 32 even when the leaf node is increased, the execution time does not decrease, and the parallelism is not that effective for leaf matrix size greater than 32. When this value is changed for different matrix sizes, the time keeps decreasing as the size increases but beyond a point it keeps increasing. Here a iterative routine is used and combined into a recursive routine.