Venkata Shiva Krishna Reddy Avula
Email- vavula1@uncc.edu
Id- 800963515

1. A. Please find the code heat.cpp file in the rar.
   The partitioning was horizontal partitioning among nodes.
   B.and C.

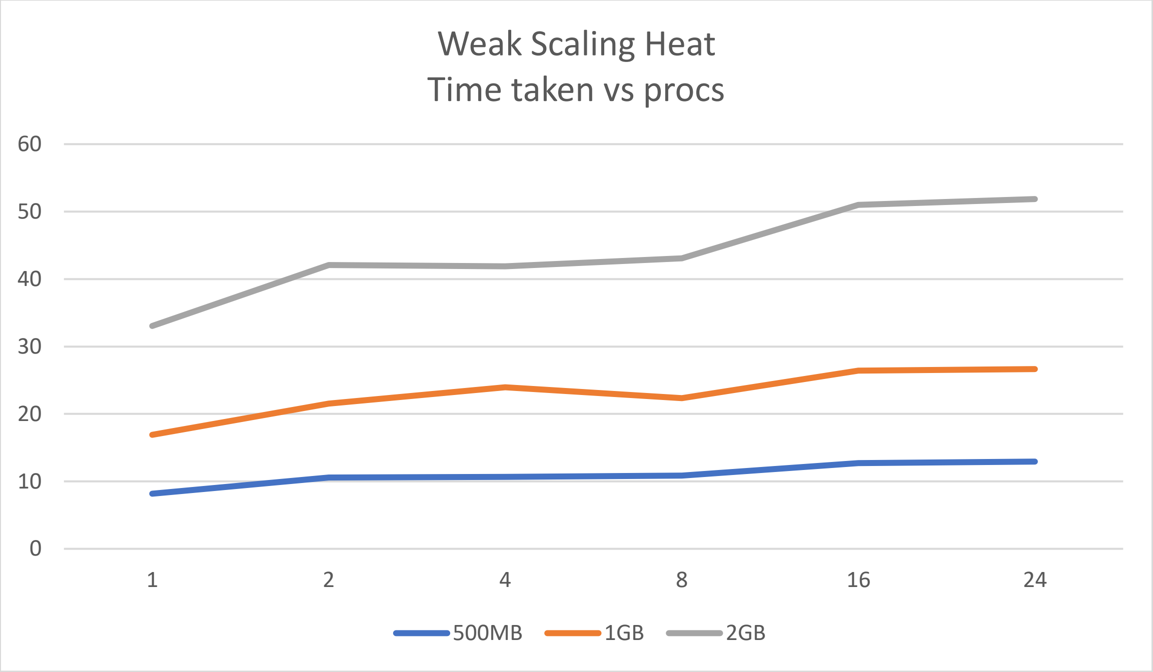
Strong Scaling Experiment	Procs	Time in Seconds
1GB	1	17.86
	2	10.92
	4	5.5
	8	2.76
	16	1.69
	24	1.10
4Gb	1	65
	2	41.28
	4	20.73
	8	10.81
	16	6.24
	24	6.48
10Gb	1	-
	2	107.35
	4	53.9
	8	27.7
	16	16.6
	24	10.89
80	1	-
	2	-
	4	-
	8	137.99
	16	82.5
	24	54

Weak Scaling Experiment	Procs	Time in seconds
500	1	8.17
	2	10.58
	4	10.65
	8	10.85
	16	12.7
	24	12.93
1gb	1	16.9
	2	21.55
	4	23.99
	8	22.33
	16	26.44
	24	26.65
2gb	1	33.04
	2	42.07
	4	41.91
	8	43.07

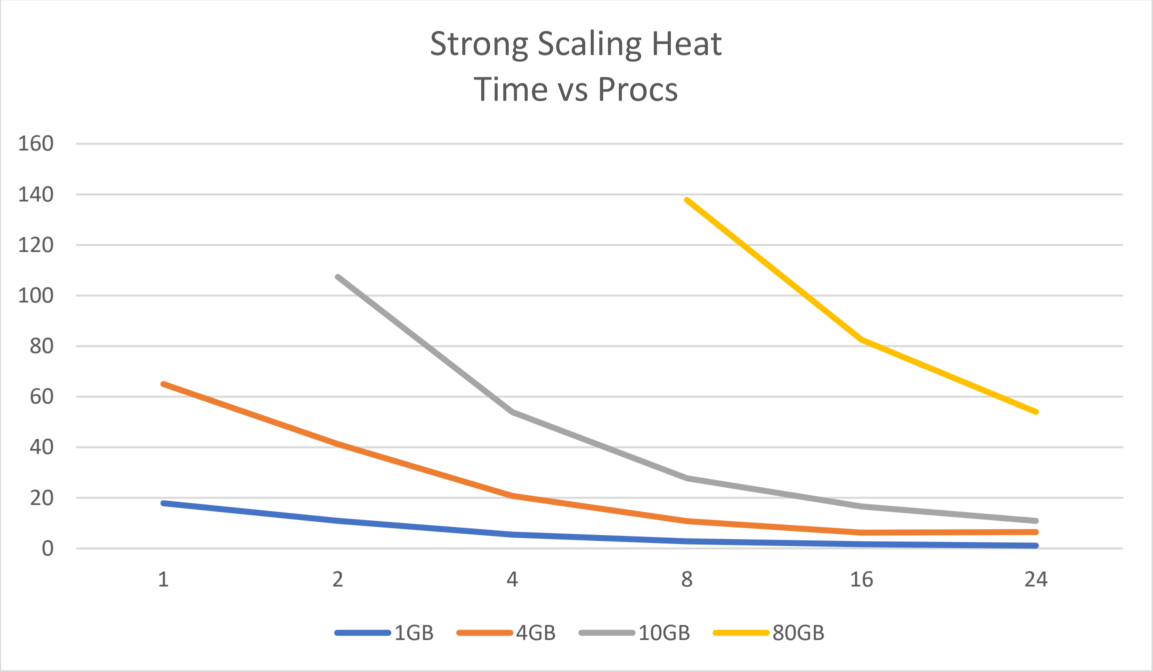
Venkata Shiva Krishna Reddy Avula
Email- vavula1@uncc.edu
Id- 800963515

	16	51
	24	51.86

Graphs for weak Scaling



Graphs for Strong Scaling

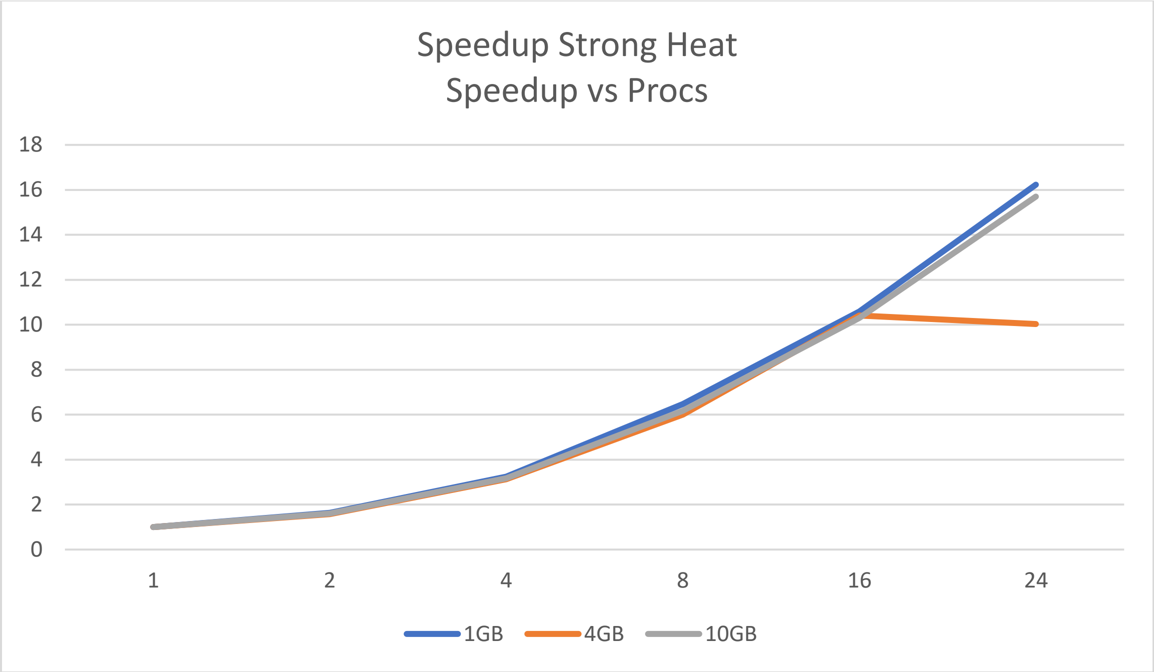


D.

2. A. Please find the matmul.cpp code file

D.

Venkata Shiva Krishna Reddy Avula
Email- vavula1@uncc.edu
Id- 800963515



D. The computation and communication can be over lapped by first calculating the elements of row succeeding the 1st data row in each node till the second last row. In the meanwhile, the nodes can send and receive data between the adjacent nodes through MPI_ISEND and MPI_IRECV. Before calculating the first and last data row each node can wait to check whether all the communication has happened and if it finishes they can calculate the first and last rows in each node in each iteration.

2. A. Please consider the mat_mul.cpp code file.

```
Initializing the X array on top row nodes>>>

Initializing the A array in each node.>>
Iterations =0
While(iterations <20){
Sending the X array to the nodes in same column>>
Nodes that are not in the first receive the X array>>
Calculating the partial sum in the X array>>
Row wise reduction of elements on the nodes in first column>>
Sending the reduced sum X array to the nodes in top row.>>
The top row nodes except node0 receive the X array
Iterations++}
```

D. The communication and computation overlap can be done for the large matrices on each ode by further dividing the Xarray. So that when a part of Xarray of is received each node start calculation for the corresponding indices and store at the first index. Meanwhile, second part of X array can be received and so on. And for the reduction we could first reduce on each node and the reduction can be done on all nodes before sending it to the top row nodes for each iteration.