EECS87 Parallel computing HW2

Usage

run all tests on greatlake.

```
bash run.sh
```

Example, `mpirun -np 36 --bind-to core:overload-allowed ./main 2000 500 0 1 > parallel-small-36.txt` run a (2000,500) with 36 core on without interdemiate detail with parallel mode and output to parallel-small-36.txt.

```
mpirun -np [number of core] --bind-to core:overload-allowed ./main [m] [n]
[verbose(1) or not(0)] [parallel(1) or serial(0)] > [output path]
```

Methos

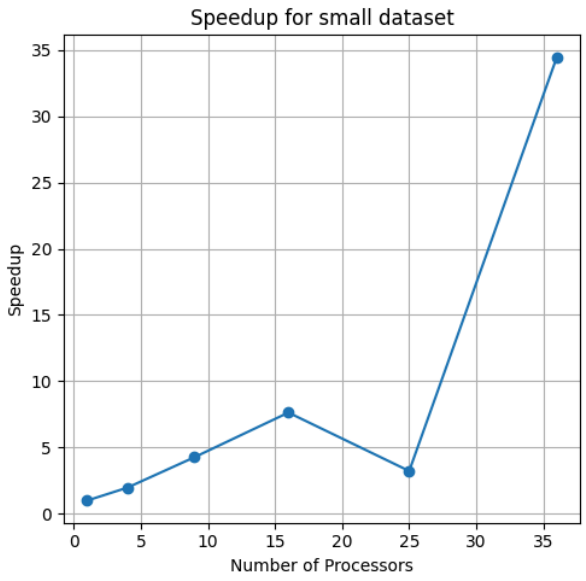
I crop the matrix into p blocks and each part has size $(\frac{m}{sqrt(p)}, \frac{n}{sqrt(p)})$. Except for the blocks on the boundary, the inner blocks need to send its left col, right col, bottom row, up row and 4 corners to the neibours.

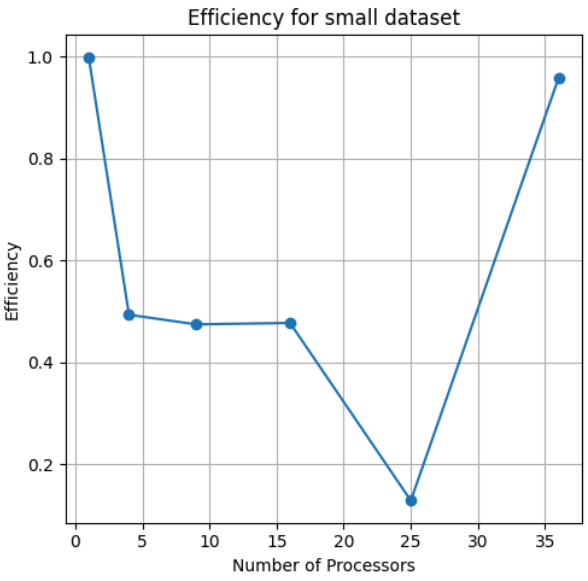
Result

I have test the model with more different numbers of processor(more than the problem mentiond.)

Small (2000,500)

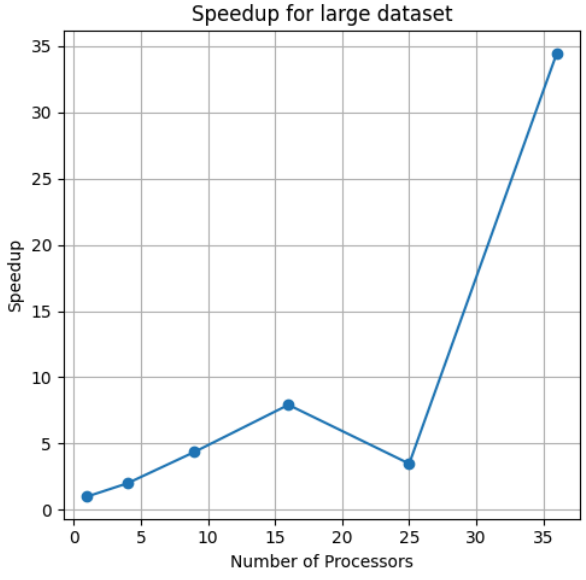
	Processors	Time Used	Sum	Sum of Square	Speedup	Efficiency
0	serial	24.3069	3.01716e+07	6.862e+09	1	1
1	1	24.3469	3.01716e+07	6.862e+09	0.998357	0.998357
2	4	12.3164	3.01711e+07	6.86199e+09	1.97354	0.493385
3	9	5.69145	3.01659e+07	6.86193e+09	4.27077	0.474531
4	16	3.18266	3.01702e+07	6.86197e+09	7.63729	0.477331
5	25	7.52991	3.01715e+07	6.86201e+09	3.22805	0.129122
6	36	0.705498	3.01365e+07	6.86163e+09	34.4535	0.957043

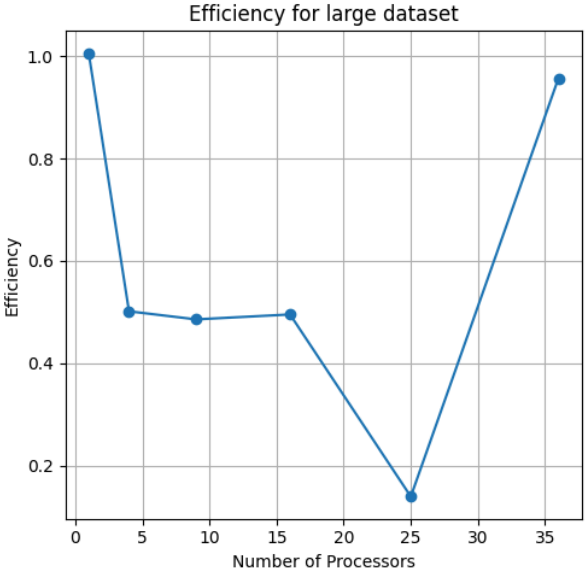




Large (1000,4000)

	Processors	Time Used	Sum	Sum of Square	Speedup	Efficiency
0	1	97.2765	1.13255e+08	1.40519e+10	1	1
1	1	96.6835	1.13255e+08	1.40519e+10	1.00613	1.00613
2	4	48.4839	1.13359e+08	1.40541e+10	2.00637	0.501592
3	9	22.2486	1.12758e+08	1.40455e+10	4.37225	0.485806
4	16	12.2718	1.12278e+08	1.40465e+10	7.92683	0.495427
5	25	27.8985	1.12026e+08	1.40411e+10	3.4868	0.139472
6	36	2.82327	1.11682e+08	1.40382e+10	34.4553	0.957091





Scaling, Speedup and efficiency

From Strong Scaling perspective, we can see that the program is linear speedup but not perfect speedup since the efficiency is less than 1 as shown in the images above. I guess this maybe caused by the communication cost when the processor number goes up. So it's not perfect strong scaling.

From weak scaling perspective, we have scaling in terms of the size of the matrix, as well as in the number of processors. When we expand the matrix size 4 time, we compare the processor number parts like (1,4) (4,16) (9,36), we can see the running time is similar, so it's perfect weak scaling.