Nicholas Allaire CSE 160 January 29th, 2017

Heat2drow 500 x 500

nproc	M	N	Tl	Tr	Tt	Tb	eps	iterations	time(ms)
1	500	500	100	0	0	0	0.001	17535	18330000
2	500	500	100	0	0	0	0.001	16978	7150000
4	500	500	100	0	0	0	0.001	16978	3740000
8	500	500	100	0	0	0	0.001	16978	2130000

Heat2drow 1000 x 1000

nproc	M	N	Tl	Tr	Tt	Tb	eps	iterations	time(ms)
1	1000	1000	100	0	0	0	0.001	18154	70340000
2	1000	1000	100	0	0	0	0.001	18154	27700000
4	1000	1000	100	0	0	0	0.001	18154	14210000
8	1000	1000	100	0	0	0	0.001	18154	7480000

Heat2drow 2000 x 2000

nproc	M	N	Tl	Tr	Tt	Tb	eps	iterations	time(ms)
1	2000	2000	100	0	0	0	0.001	18151	355400000
2	2000	2000	100	0	0	0	0.001	18151	155680000
4	2000	2000	100	0	0	0	0.001	18151	109870000
8	2000	2000	100	0	0	0	0.001	18151	77790000

Average time for
500 x 500: 7,837,500
1000 x 1000: 29,932,500
2000 x 2000: 349,685000

With an overall average time being 129,151,667

Nicholas Allaire CSE 160 January 29th, 2017

Heat2dcol 500 x 500

nproc	M	N	Tl	Tr	Tt	Tb	eps	iterations	time(ms)
1	500	500	100	0	0	0	0.001	16979	17430000
2	500	500	100	0	0	0	0.001	16978	7080000
4	500	500	100	0	0	0	0.001	16957	4050000
8	500	500	100	0	0	0	0.001	16998	2570000

Heat2col 1000 x 1000

nproc	M	N	Tl	Tr	Tt	Tb	eps	iterations	time(ms)
1	1000	1000	100	0	0	0	0.001	18154	80310000
2	1000	1000	100	0	0	0	0.001	18154	31020000
4	1000	1000	100	0	0	0	0.001	18140	15880000
8	1000	1000	100	0	0	0	0.001	206827	113710000

Heat2col 2000 x 2000

nproc	M	N	Tl	Tr	Tt	Tb	eps	iterations	time(ms)
1	2000	2000	100	0	0	0	0.001	18151	357180000
2	2000	2000	100	0	0	0	0.001	18151	189550000
4	2000	2000	100	0	0	0	0.001	18151	79310000
8	2000	2000	100	0	0	0	0.001	18167	96440000

Average time for
- 500 x 500: 7,782,500
- 1000 x 1000: 60,230,000
- 2000 x 2000: 180,620,000

With an overall average time being 82,877,500

Nicholas Allaire CSE 160 January 29th, 2017

I expected column distribution to take longer because it took some extra steps to complete compared to row distribution. Some of these steps include copying the neighboring columns to row vectors and then sending/receiving and then copying back from row vector to the column locations. Plus when storing the data in a file, you can't just copy each processes data to the file, you must first store each process into a full M x N array and then save it to a file to ensure correct ordering. My choice of border values could have influenced the time spent for each run.