

Program Structures and Algorithms
Spring 2023(SEC –3)

NAME: Kaushik Gnanasekar
NUID: 002766012

Task: Parallel Sorting

Relationship Conclusion:

- From the tests run we find that, as the size of the input array increases, the performance of the parallel merge sort improves.
- We find parallel sort works best when
length of array / $2^9 + 1 \leq \text{cutoff value} < 2^4 + 1$.
- Since the host machine has 8 cores, it seems to have worked consistently well from 4 – 8 threads. There are instances where 16 and 32 occasionally do well, but the overhead increases as the threads may spend more time waiting for each other. Managing the threads is an additional overhead and reduces performance.
- The performance of the parallel merge is highly dependent on the size of the input array, the cut off value for switching to sequential sort and the number of thread used for parallel sorting.

Evidence to support that conclusion:

Size	Cut-off value	Threads				
		2	4	8	16	32
32768	33	47.77844	45.99284	57.41401	36.68063	54.1852
	65	17.17646	9.40345	16.39672	18.83177	20.53028
	129	8.377642	7.948158	9.07055	8.470338	9.604033
	257	5.295792	4.807708	5.068971	5.079525	5.365413
	513	3.726817	3.614029	3.776037	3.873975	3.902888
	1025	2.787254	3.216925	3.452087	3.222154	2.928921
	2049	2.565354	2.9672	4.638583	3.7192	2.534217
	4097	2.822912	2.789146	2.863667	4.171304	2.497071
	8193	3.517025	2.350129	3.670279	3.524588	2.963113
	16385	2.797488	1.80015	3.162846	2.042146	2.832417
	32769	2.836121	2.837842	2.829962	1.906612	3.054358

Size	Cut-off value	Threads				
		2	4	8	16	32

1048576	1025	78.29353	80.09089	78.90559	77.57944	95.91934
	2049	47.56792	45.05024	54.04005	45.37641	56.91568
	4097	41.26229	41.7207	48.64243	47.40741	38.50228
	8193	36.39687	37.72699	43.0254	45.67245	36.55703
	16385	37.54335	38.62409	36.61676	36.51503	36.95573
	32769	39.02963	39.3303	38.15051	42.30719	32.14715
	65537	43.69852	43.37497	42.15135	35.11809	32.28892
	131073	54.83293	45.81129	40.09797	33.75142	33.13891
	262145	59.14599	48.1234	37.42054	35.22664	36.18713
	524289	43.15455	43.19981	44.12469	45.24966	47.57439
	1048577	66.78743	66.9415	67.34577	69.35719	67.18683

Size	Cut-off value	Threads				
		2	4	8	16	32
2097152	2049	131.6477	119.3487	126.2775	129.8828	151.5295
	4097	92.55391	113.4295	93.24558	82.56407	81.4156
	8193	71.16703	85.64523	76.02942	77.99525	81.84653
	16385	72.85899	71.631	75.41515	74.91332	74.43244
	32769	73.02237	80.17005	75.92763	95.99372	80.20764
	65537	80.70801	82.01894	78.44914	88.43422	71.19905
	131073	98.6689	95.2543	97.91977	77.25585	68.42556
	262145	102.5109	117.298	88.724	67.21209	68.86018
	524289	121.8072	98.33406	75.60064	79.4403	77.0712
	1048577	90.78186	93.77628	98.3787	98.67425	99.16264
	2097153	140.9911	140.0429	144.0419	144.824	147.3691

Size	Cut-off value	Threads				
		2	4	8	16	32
4194304	4097	222.1426	229.4767	214.7608	256.977	219.3078
	8193	153.8627	152.8425	171.0824	193.5773	162.6699
	16385	145.7601	154.9321	154.6511	156.0789	160.2744
	32769	146.7211	156.8542	158.3745	158.3908	153.5819
	65537	154.6767	166.6599	181.1013	153.4659	162.5771

	131073	178.9217	171.7509	174.7077	172.7314	137.0311
	262145	198.4115	218.3092	187.9169	158.5506	137.5918
	524289	209.2992	232.7925	193.4576	133.5776	146.7699
	1048577	259.5999	210.5898	158.9649	160.7802	159.1808
	2097153	195.6371	199.6034	202.7555	207.4354	201.5283
	4194305	294.9495	300.335	300.331	301.7693	298.4891