



Thread Count	Total Program Duration	Total Read Duration	Total Write Duration	Total Overhead Duration
512m				
1	332	55	152	125
25	2846	347	940	1559
50	3813	464	1258	2090
75	3721	470	1255	1996
100	4023	476	1371	2176
1g				
1	639	104	293	241
25	5352	639	1745	2968
50	6983	845	2323	3815
75	7142	869	2381	3892
100	8969	2643	2991	4903
2g				
1	1345	212	640	492
25	11494	1363	3780	6351
50	12110	1449	4094	6566
75	14493	1723	4867	7906
100	16476	1648	4654	7794
4g				

1	2727	400	1351	976
25	23817	2730	7736	13351
50	24240	2804	8216	13248
75	24685	2768	7879	13145
100	28082	3161	8955	13041

1. Why did you choose to measure these parts – provide reasons for each.

Total Program Duration: to calculate how long it took to finish running the program.

Total Read Duration: to calculate how long it took to append a line to the writer queue.

Total Write Duration: to calculate how long it took to write the line to the output file.

Total Overhead Duration: the difference in program duration minus read and write duration.

2. Where was the most time spent? How does that relate to the operations being done and to the overall time?

The writing takes the longest amount of time.

3. Did increasing and/or decreasing the number of threads change the performance of the program? How do you account for this?

Yes, it decreased the performance. Reading, writing, and the overhead take longer because there are more threads waiting. With only a single thread, copier was able to finish about 10x faster than mtcopier. At 512m, copier took 332 seconds while mtcopier at 20, 50, 75, and 100 threads took 2846, 3813, 3721, 4023 seconds respectively. You will find similar results for each size groups.

Even just looking at mtcopier, the increase of threads is correlated with an increased duration for the total program to complete, to read a line, to write a line, and the overhead.

4. Was it worthwhile to turn this program into a multithreaded program? Explain your reasoning.

No because waiting for each thread to finish their operation takes too long. It would be faster to just have one thread do all the work and not waste time waiting.