Timing Difference between NVME and Scratch

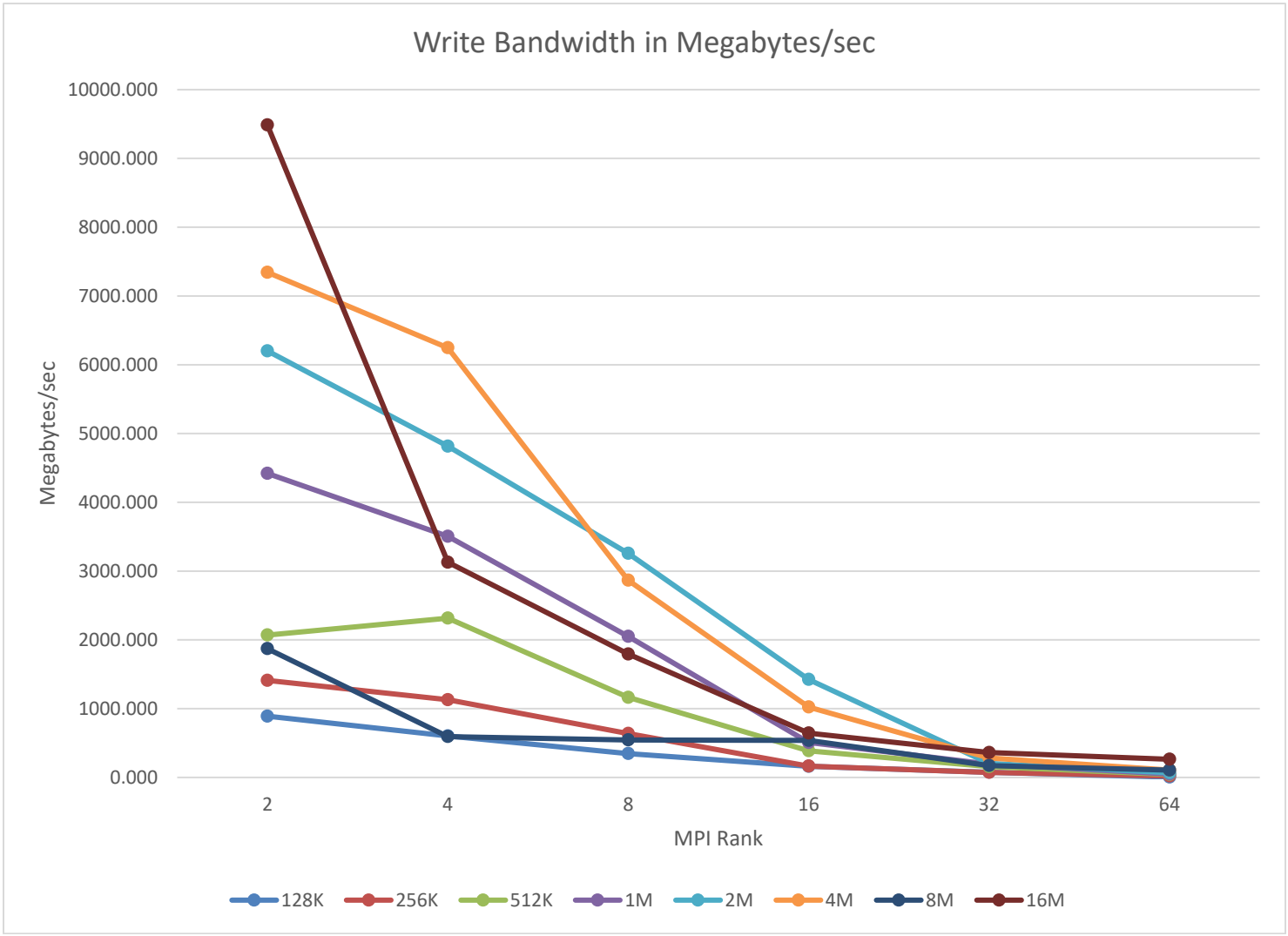
By:
Trevor B
Aaron C
Bill H

Notes for when running the scripts for the code:

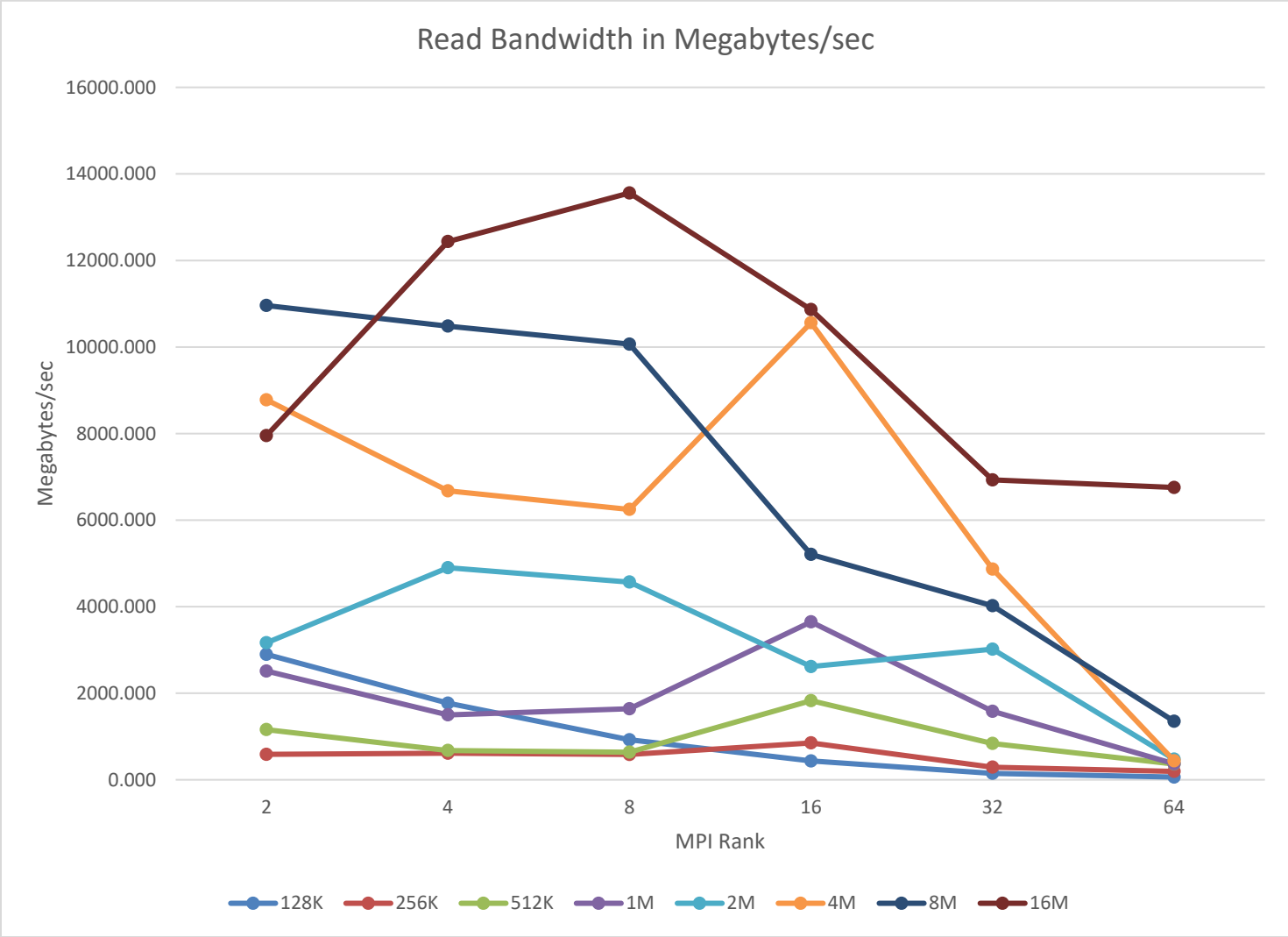
There are two different .sh files for running our code.
The slurm_nvme.sh file runs all block sizes for the nvme tests
The slurmSpecturm.sh runs for a one block size for the scratch tests

Both scripts need to be called with sbatch to determine the number of ranks

Scratch Write Data:



Scratch Read Data:



For the Write Bandwidth data, overall as the number of MPI Ranks increased, bandwidth fell at a fairly consistent rate across all block sizes. Between block sizes, as they increased, typically the amount of bandwidth increased as well but still decreased to a relatively small value as the MPI Rank approached 64. The rate at which bandwidth decreased also differed between block sizes with a more linear nature until roughly block size 4M where is became more exponential. The 8M block size seems to be an outlier in our data with a very low Write Bandwidth that we expected to be somewhere between the 4M and 16M data lines especially for MPI Rank 2. This is an interesting trend that we couldn't figure out as it consistently ran with much lower bandwidth compared to the 4M and 16M block sizes and could definitely be an area of further testing. The 16M block size also had a very drastic drop from MPI Rank2 to MPI Rank 4 which we believe is due to the 16M size overloading the I/O.

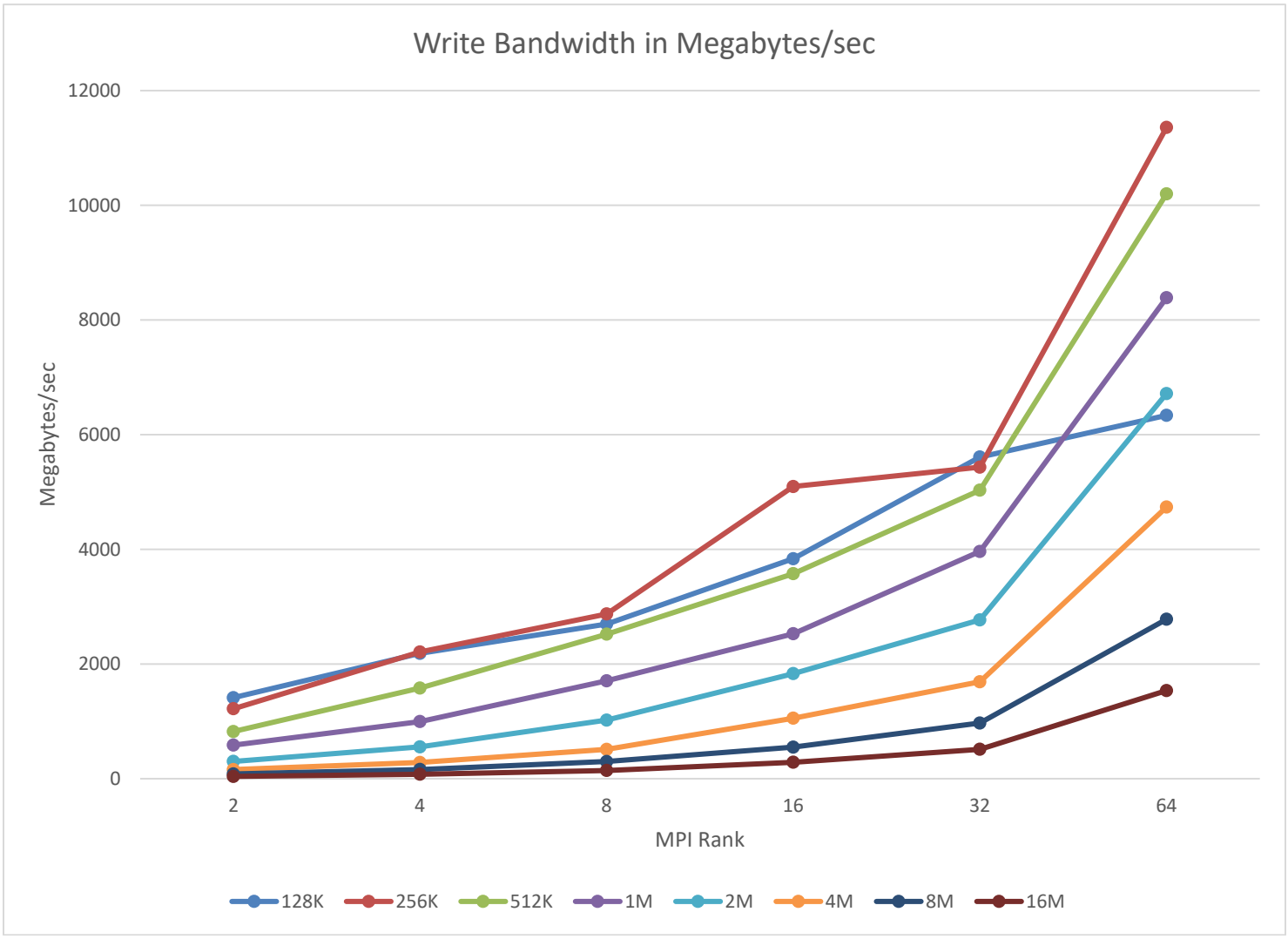
Read Bandwidth data differed mostly from the Write Bandwidth data except that the overall trend was a decreasing bandwidth as MPI Rank increased. Read bandwidth also generally increased as the size of the block increased, similar to the Write Bandwidth data except that it the gap between block size bandwidths stayed relatively constant from MPI Rank 2 to MPI Rank 16 before converging to a much lower value at MPI Rank 64. The Read Bandwidth was also much more inconsistent, often spiking to a larger Bandwidth at a seemingly random MPI Rank before ultimately decreasing lower than the MPI Rank 2 bandwidth as it hit MPI Rank 64. Read Bandwidths were generally lower than Write Bandwidths at lower MPI Ranks but became larger usually past MPI Rank 8 or 16 before a fairly drastic drop in performance at MPI Rank 64, this trend is probably due to the fact that writing must set the bytes while read just needs to check them allowing bandwidth to remain more constant across a block size.

Raw Bandwidth Data:

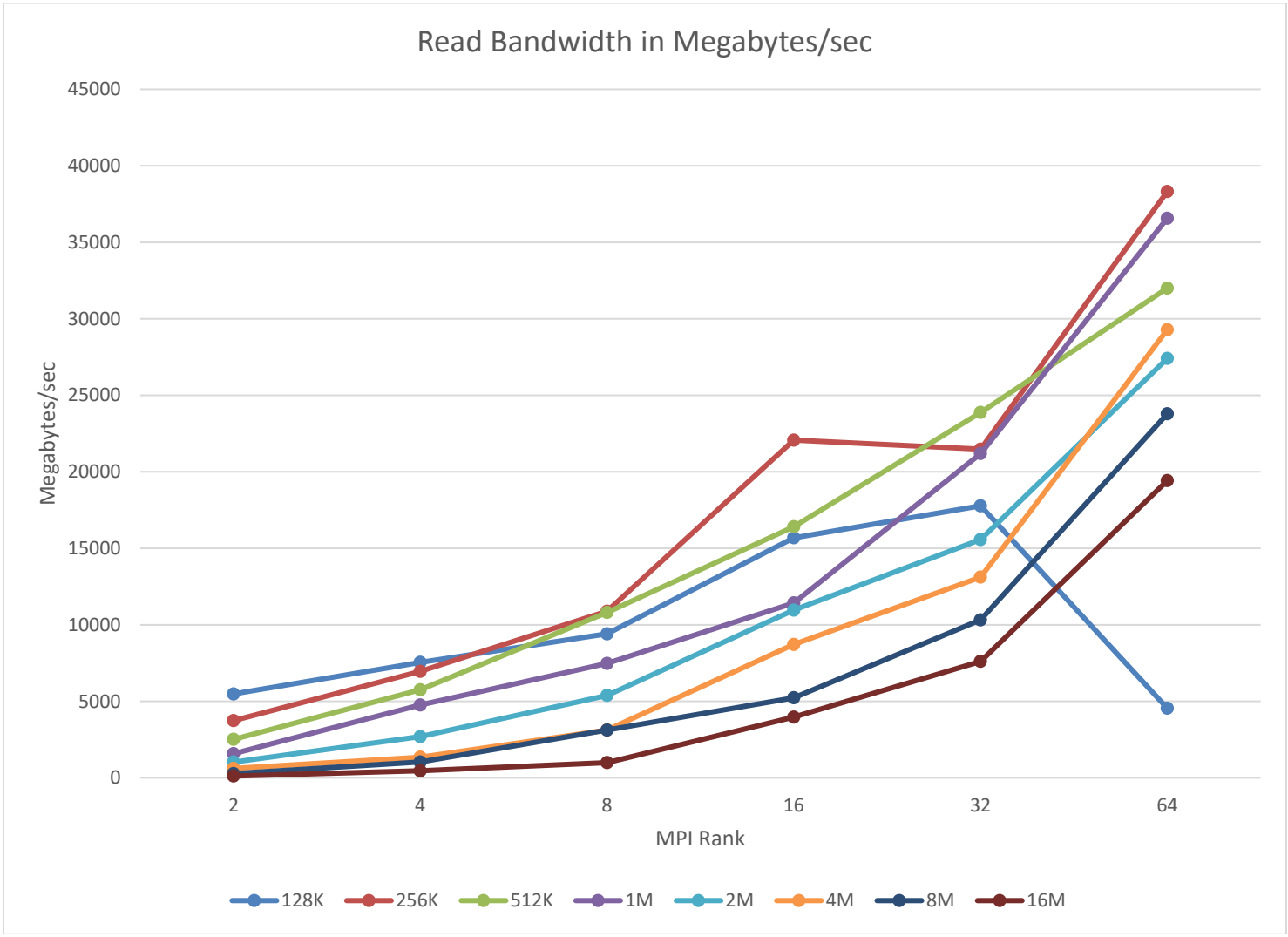
Write Bandwidth								
	128K	256K	512K	1M	2M	4M	8M	16M
2	890.869	1410.935	2069.858	4419.890	6201.550	7343.660	1871.756	9484.994
4	603.318	1129.944	2315.485	3504.929	4815.651	6246.950	594.050	3128.628
8	346.921	638.468	1163.636	2049.968	3256.997	2866.099	544.890	1794.288
16	161.225	163.968	387.597	508.098	1423.804	1022.936	538.460	643.896
32	75.415	73.415	156.832	193.435	204.538	282.710	172.477	361.883
64	4.656	20.052	42.978	49.669	67.184	107.158	106.526	263.875

Read Bandwidth								
	128K	256K	512K	1M	2M	4M	8M	16M
2	2898.551	586.510	1158.581	2511.774	3165.183	8779.150	10958.904	7951.545
4	1769.912	613.027	676.533	1498.829	4900.459	6677.100	10483.210	12436.240
8	923.788	583.516	636.943	1640.185	4568.166	6246.950	10066.850	13559.322
16	434.311	851.064	1828.571	3648.803	2616.517	10561.056	5207.486	10865.874
32	148.588	290.592	837.696	1582.591	3017.445	4868.771	4020.732	6929.219
64	62.794	193.470	363.306	369.430	475.801	438.988	1350.567	6754.617

NVME Write Data:



NVME Read Data:



For Write Bandwidth, there was an overall exponential-like increase for each block size as the number of MPI Ranks increased. Write Bandwidth decreased as the block size increased, with the best bandwidth achieved at a block size 256K and decreasing at each block size after until 16M which had the lowest bandwidth. The exception to this is the 128K block size which stayed almost even with the 256K block size until the 64 rank run where it flattens out and stays relatively close to the bandwidth performance at 32 ranks. This is probably because 64 ranks creates too much overhead for the smaller block size and actually hinders the performance. Knowing this, if rank continued to increase, we expect to see a similar trend for the other block sizes, but further testing would obviously be needed to show this behavior. NVME performed best overall at 64 MPI ranks for each block size.

Read Bandwidth had very similar results to the Write Bandwidth with an overall exponential-like increase for each block size as the number of MPI Ranks increased. The best bandwidth was also achieved at a block size of 256K and the overall bandwidth decreased as block size increased. There was also a drop in performance at the 128K block size with 64 ranks, but instead of just flattening out like for the Write, the Read actually had a large drop in performance to below the performance at 2 ranks. We again believe that this is due to the amount of overhead produced by the 64 MPI ranks on the relatively small block size. Similarly, we would also expect some drop off point for each of the other block sizes as the number of MPI Ranks increased from 64, but more testing would be needed to show this. Read Bandwidth was a bit more inconsistent when compared to Write Bandwidth data where the Write data had very clear separation between block size performances (discussed above), Read data had a bit more variability with larger block sizes sometimes outperforming a smaller block size for a specific MPI Rank. This is most likely due to differences in the system between runs, since the read only needs to check the bits instead of setting them, small differences between runs could have a larger effect on the Read performances. NVME performed best overall at 64 MPI ranks for each block size except for the 128K block size which performed best at 32 MPI ranks. Read performances overall had much higher bandwidths compared to Write Bandwidths, ~3x for the best performance at 256K block size and ~10x for the best performance at 16M block size.

Raw Write Bandwidth Data for NVME:

	128K	256K	512K	1M	2M	4M	8M	6M
2	1413.428	1219.512	823.0453	585.6515	300.9782	159.5533	83.38545	40.37549
4	2185.792	2209.945	1581.028	996.264	554.785	283.2861	160.2243	77.59457
8	2698.145	2872.531	2519.685	1707.577	1022.364	511.8362	299.5133	143.6395
16	3836.93	5095.541	3575.419	2527.646	1832.761	1054.713	550.585	287.0728
32	5609.115	5432.937	5031.447	3962.848	2769.364	1689.992	970.285	512.5741
64	6336.634	11357.59	10199.2	8387.942	6715.635	4737.232	2780.795	1536.984

Raw Read Bandwidth Data for NVME:

	128K	256K	512K	1M	2M	4M	8M	16M
2	5479.452	3738.318	2515.723	1581.028	1028.278	613.4969	277.3925	115.4401
4	7547.17	6956.522	5755.396	4761.905	2693.603	1346.801	1033.592	459.7701
8	9411.765	10884.35	10810.81	7476.636	5387.205	3125	3118.908	991.9405
16	15686.27	22068.97	16410.26	11428.57	10958.9	8719.346	5228.758	3970.223
32	17777.78	21476.51	23880.6	21192.05	15571.78	13114.75	10322.58	7609.988
64	4547.069	38323.35	32000	36571.43	27408.99	29290.62	23791.82	19423.37

Final Conclusions – Comparing Scratch and NVME:
For writing data, NVME for block sizes up to 2M at 64 MPI ranks performed better (~10x at best to ~1x at worst) than Scratch for block sizes up to 2M at 2 MPI ranks, so NVME would be the obvious choice if performing writes of this size. After 2M block size, Scratch had better performances at 2 MPI ranks to 8 MPI ranks, this might make Scratch better for larger block sizes (~5x at best to ~1x at worst), but we did not see any drop in performance yet for larger block sizes in NVME. This could mean that larger MPI ranks might continue to improve performance to better than Scratch, but more testing would be needed.

For Read Bandwidth, NVME outperformed Scratch at every MPI Rank for block sizes up to 512 K. Even the drop at the 128K with 64 ranks had a better performance than all of Scratch's 128K performances. After a block size of 512 K, NVME eventually outperforms Scratch as the number of ranks increases, but needs more MPI Ranks to increase performance passed Scratch as the block size increases. NVME had a ~3x increased performance between its best performance and Scratch's best performance. Also, the general trend shows increasing NVME performance and decreasing Scratch performance for reads, so it would be best to use NVME for reads especially if MPI ranks can be used. Obviously it would be better to perform more tests to see where the performance drop off exists to determine an optimal number of ranks for each block size before doing any important research/work since there is a drastic drop at 64 ranks for block size 128K.