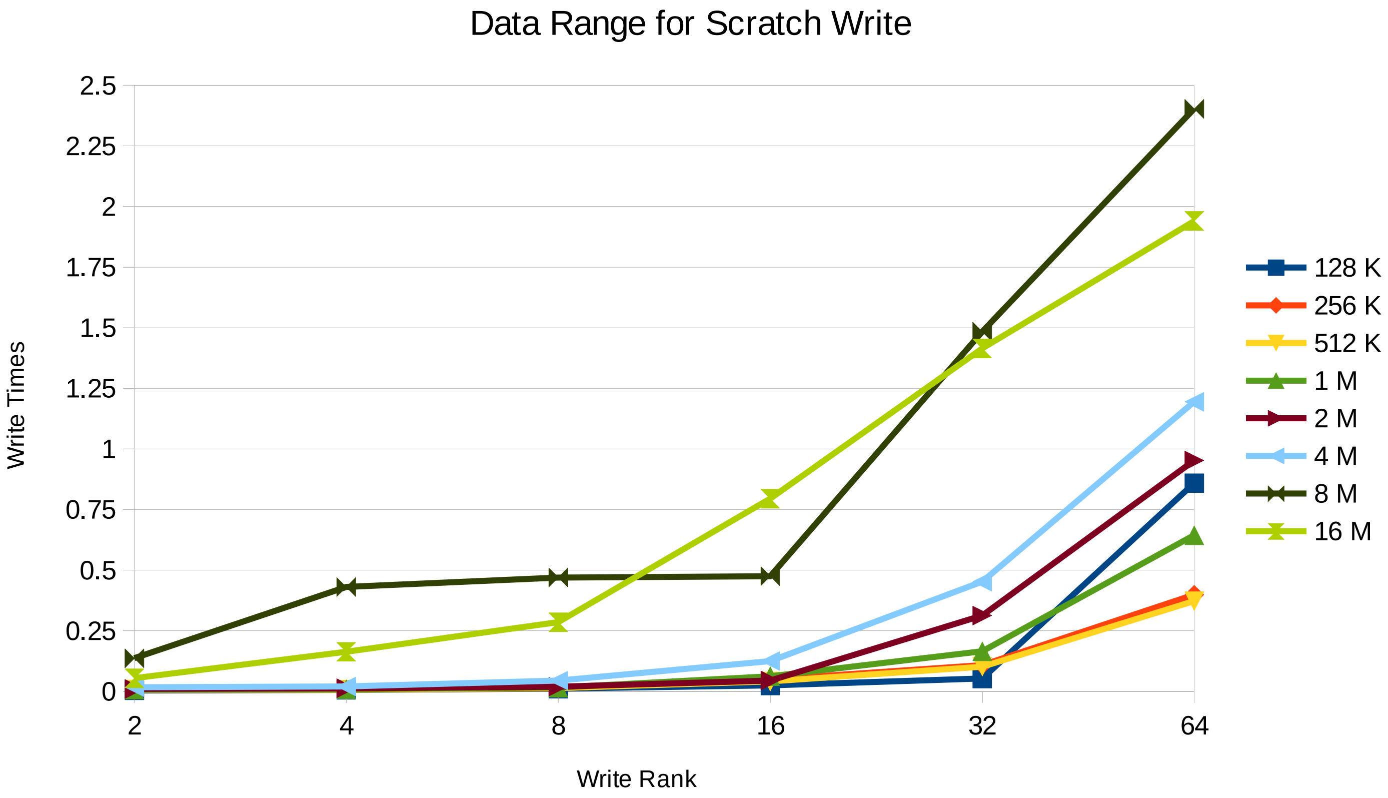
Timing Difference between NVME and Scratch

By:
Trevor B
Aaron C
Bill H

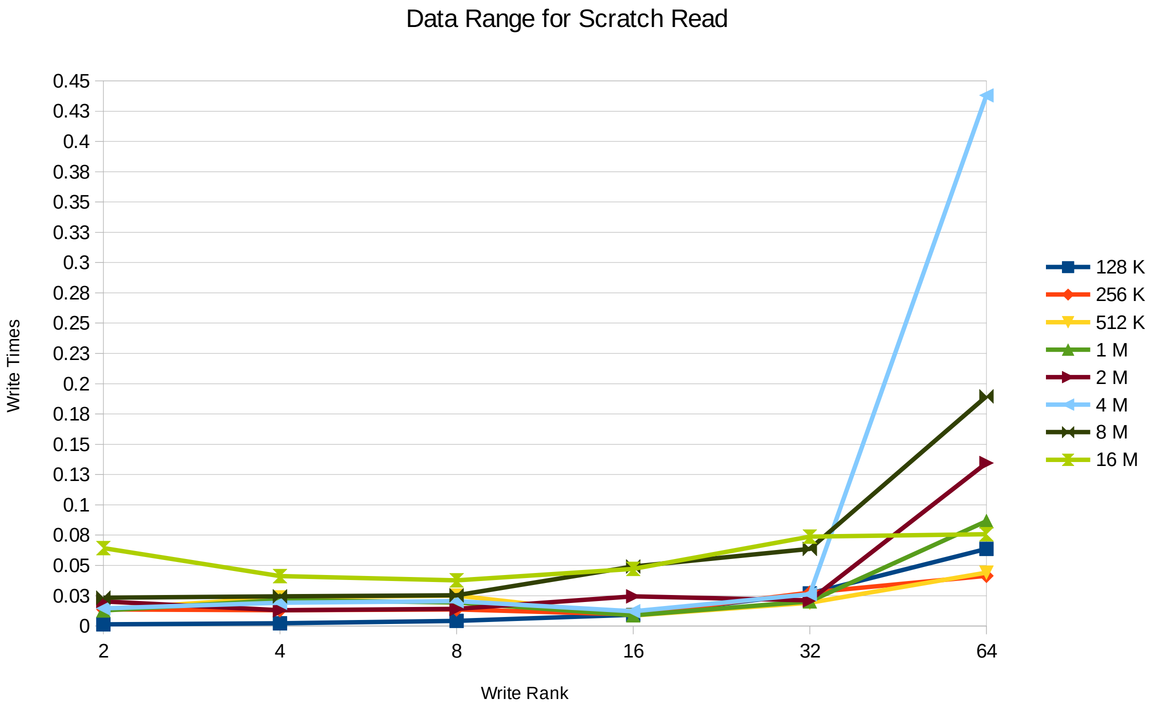
Notes for when running the scripts for the code:

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

Data Range for the Scratch Write:



Data Range for the Scratch Read:



Some things that we noticed with the data was that for both the read and write, up to rank 16 the data was fairly consistent with the other ranks. The trends and values were fairly consistent with each other. You start to see the data separate from the other points at write rank 32 and read rank 64. With this the amount of time between these can vary quite differently. Our times for these jobs are below:

write

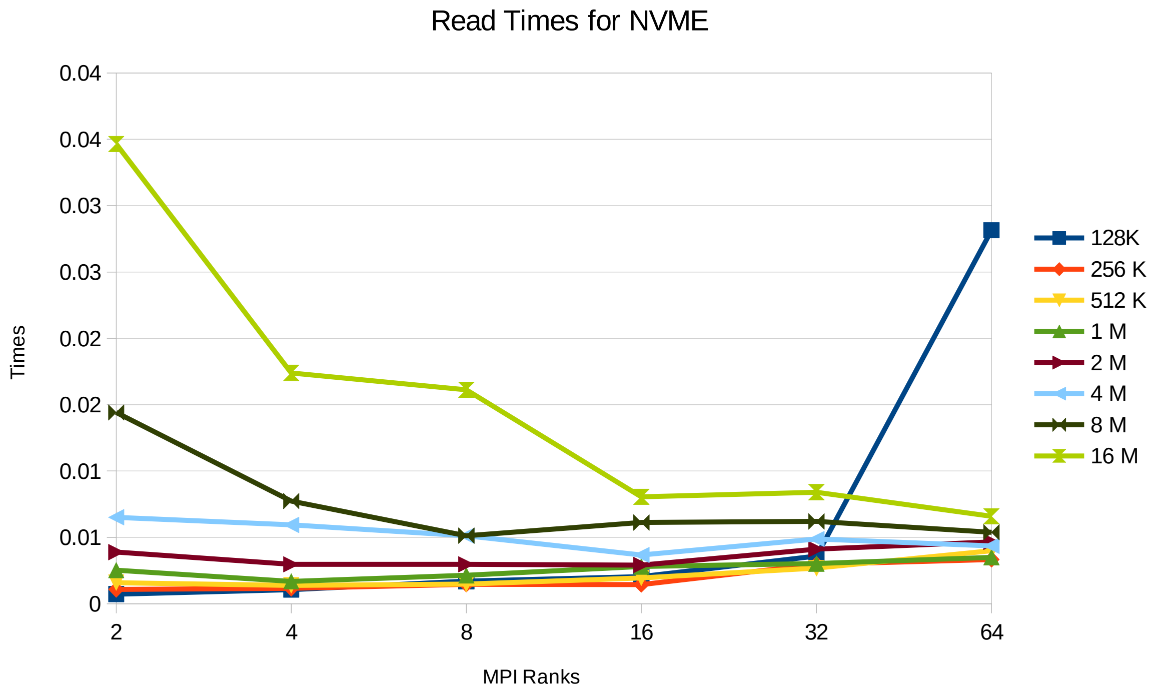
	128K	256K	512K	1M	2M	4M	8M	16M
2	0.00449	0.00567	0.00773	0.00724	0.01032	0.01743	0.13677	0.05398
4	0.00663	0.00708	0.00691	0.00913	0.01329	0.02049	0.43094	0.16365
8	0.01153	0.01253	0.01375	0.01561	0.01965	0.04466	0.46982	0.28535
16	0.02481	0.04879	0.04128	0.06298	0.04495	0.12513	0.47543	0.79516
32	0.05304	0.10897	0.10202	0.16543	0.3129	0.45276	1.48426	1.41482
64	0.85915	0.39896	0.37228	0.64426	0.95261	1.1945	2.40316	1.94031

read

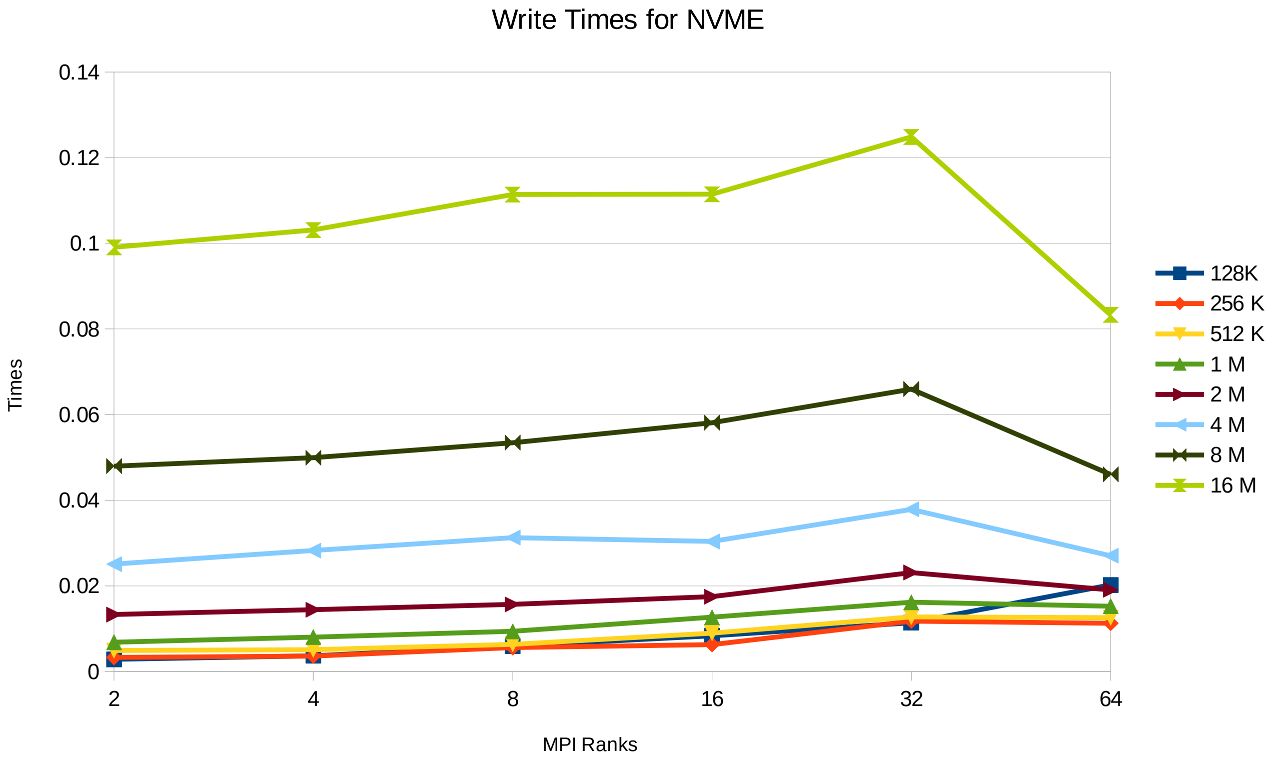
	128K	256K	512K	1M	2M	4M	8M	16M
2	0.00138	0.01364	0.01381	0.01274	0.02022	0.01458	0.02336	0.06439
4	0.00226	0.01305	0.02365	0.02135	0.01306	0.01917	0.02442	0.04117
8	0.00433	0.01371	0.02512	0.01951	0.01401	0.02049	0.02543	0.03776
16	0.00921	0.0094	0.00875	0.00877	0.02446	0.01212	0.04916	0.04712
32	0.02692	0.02753	0.0191	0.02022	0.02121	0.02629	0.06367	0.07389
64	0.0637	0.04135	0.04404	0.08662	0.13451	0.43796	0.18955	0.0758

These graphs seemed to follow an exponential-ish kind of graph, with increasing rank size meaning longer read and write times. For some reason the 4M, 64 rank time jumped way more than we expected, but we believe this is a bug or some error when we ran the code on the computer.

NVME Read Data:



NVME Write Data:



So the data for this graph looks fairly similar to the ones above, however there are some key differences. With this graph the first noticeable point is the high initial time the 64 rank has. We are unsure to why this is, but are assuming it is because of the high overhead for such little rank usage. The other noticeable point is that, with rank increase, the 16M read times decreased which was super interesting to see. In general however, the trend for the read was that once a certain rank was reached, the time would level out. For the write data, its a easier more difficult to make a generalization. For all file sizes, the times would increase then decrease, with the 32nd rank taking the most time for all file sizes. We are not sure why this rank in particular took the longest, but assumed it must be something with MPI.

Write Times for NVME:

MPI Ranks Block size KB	2	4	8	16	32	64
128	0.00283	0.00366	0.00593	0.00834	0.01141	0.0202
256	0.00328	0.00362	0.00557	0.00628	0.01178	0.01127
512	0.00486	0.00506	0.00635	0.00895	0.01272	0.01255
1024	0.00683	0.00803	0.00937	0.01266	0.01615	0.01526
2048	0.01329	0.01442	0.01565	0.01746	0.02311	0.01906
4096	0.02507	0.02824	0.03126	0.03034	0.03787	0.02702
8192	0.04797	0.04993	0.05342	0.05812	0.06596	0.04603
16384	0.09907	0.1031	0.11139	0.11147	0.12486	0.08328

Read Times for NVME:

MPI Ranks Block size KB	2	4	8	16	32	64
128	0.00073	0.00106	0.0017	0.00204	0.0036	0.02815
256	0.00107	0.00115	0.00147	0.00145	0.00298	0.00334
512	0.00159	0.00139	0.00148	0.00195	0.00268	0.004
1024	0.00253	0.00168	0.00214	0.0028	0.00302	0.0035
2048	0.00389	0.00297	0.00297	0.00292	0.00411	0.00467
4096	0.00652	0.00594	0.00512	0.00367	0.00488	0.00437
8192	0.01442	0.00774	0.00513	0.00612	0.0062	0.00538
16384	0.03465	0.0174	0.01613	0.00806	0.00841	0.00659

Final Conclusions:

We noticed that the read times for both scratch and nvme were fairly similar and following the same general trends. The biggest factor however was the write times for NVME, which all across the board where much faster than scratch. For large file sizes, where timing is a factor, NVME would be the route to take for the file system, for at least the writing of these files and probably should be for the read of these files as well.