Evaluation

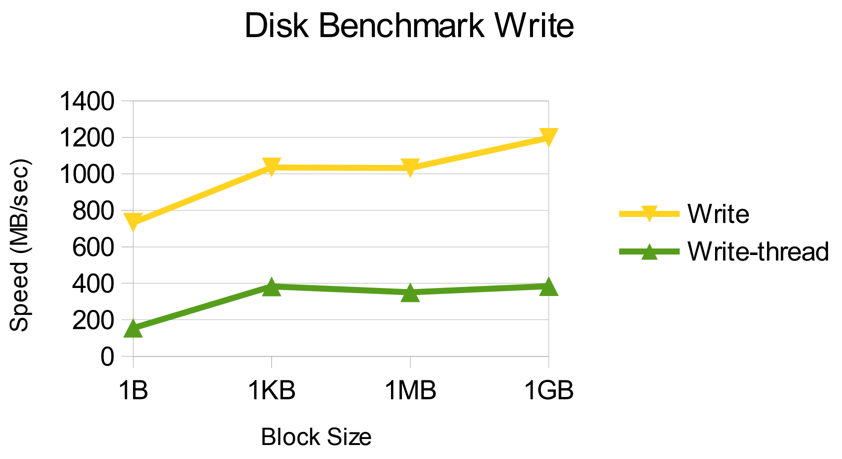
Kevin Brandstatter
Weiwei Wu

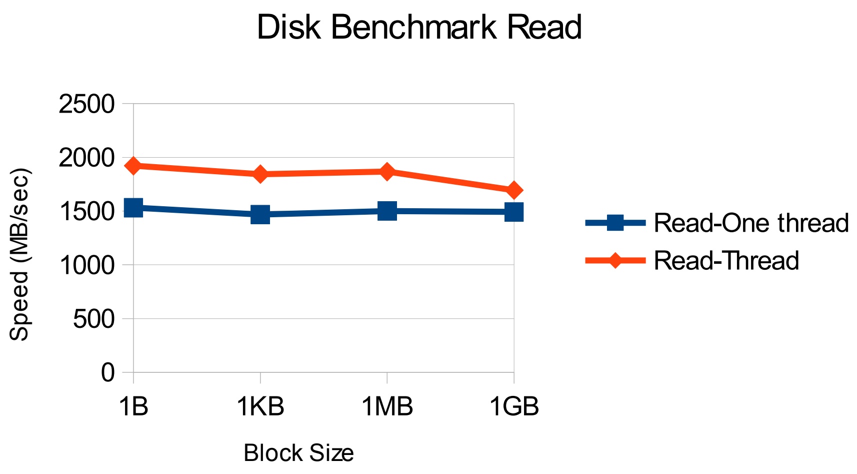
All Benchmarks done on AMD Phenom x6 @ 3.3 GHz (6 cores). Gigabit Ethernet card. Two 1 TB harddrives in Raid 1. 16 GB of RAM. System at mostly Idle for tests. Network tests done over the local loopback interface.

Diskbench:

From the results of the disk benchmark its easy to conclude several things. As expected reads are much faster than writes. We can also conclude that we can get higher throughput using larger blocksizes for the write operations. On the other hand, Using larger blocksizes for the read operations seem to degrade performance.

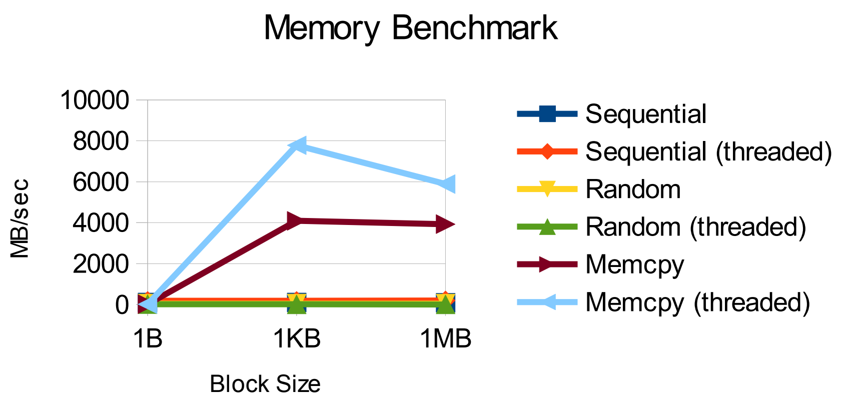
On a more unexpected result, it seems that our threaded operations are much slower than their non threaded counterpart. This is likely due to the fact that the threads are competing for the read and write heads of the disk which do not parallelize very well. This causes the operations to thrash on disk thus lowering performance.





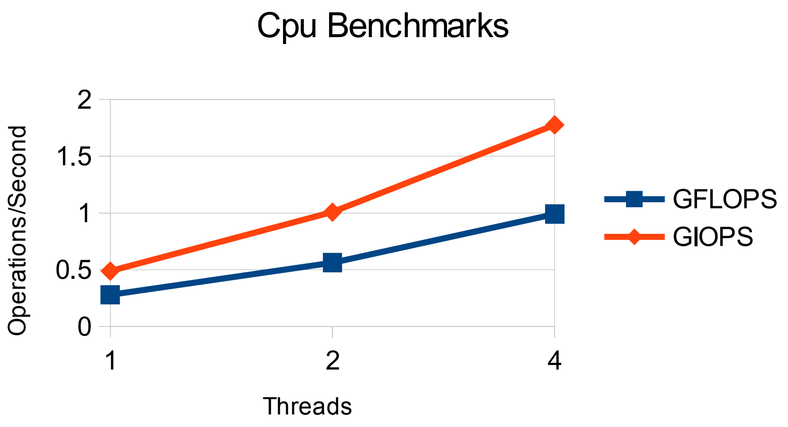
Memory Benchmark

From our memory benchmark we can see that memcpy was the most affected by an increase in block size which idicates that the memcpy operation is very slow for large data sizes. Also, they all saw perfomance degrade as the number of threads increased. This is most likely due to hardware constraints of memory access.

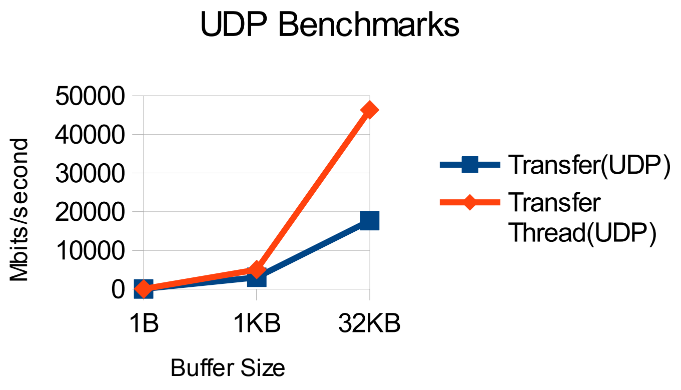


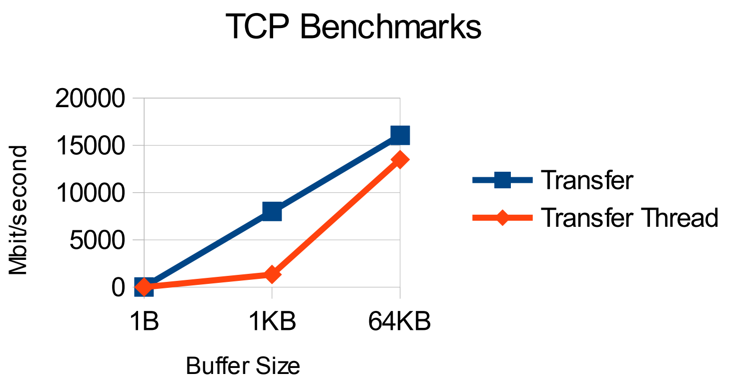
blocksize	Sequential	Sequential (threaded)	Random	Random (threa	Memcpy	Memcpy (threa
1B	105.085998	184.537755	43.241198	17.85525	16.273393	21.971986
1KB	94.731311	182.74619	45.775756	12.108502	4089.456869	7780.415994
1MB	85.508912	197.477533	41.957996	3.406231	3928.477298	5883.149813

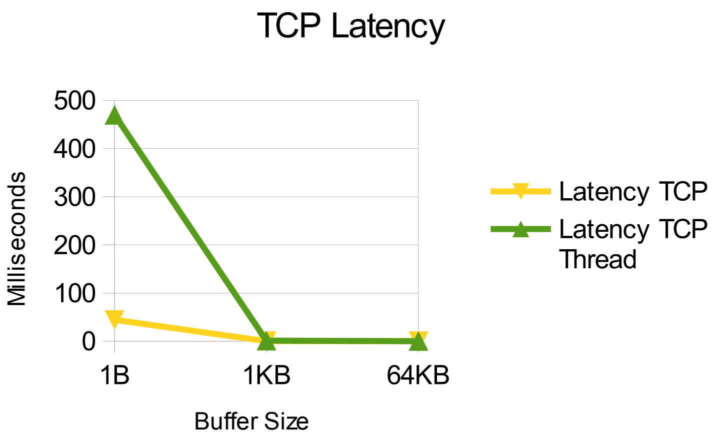
Cpubench:
From the results its clear to see that as we add parrallelism to the process, we increase cpu performance. While this performance is not the ideal 2x it approaches 2x which is very promising given that some overhead will be used by the system for scheduling and such. Also, as expected GFLOPS are lower than GIOPS as they are more cpu intensive.

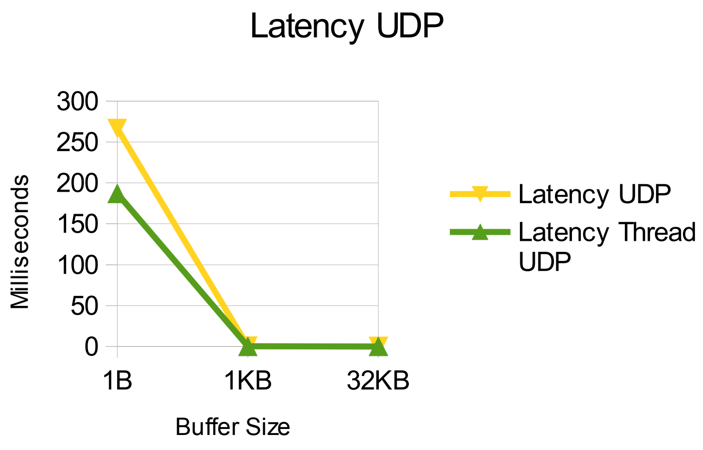


NetBench









From the network benchmarks we can see that there is a serious benifet to threading the operations as UDP and TCP both had higher throughput and lower latency with multiple threads. Also, The tests show that there is a benefit to using a larger buffer size. This is because with a larger buffer size, the overhead of the data transmitted is amortized into the time of sending the data itself and does not have as big of an impact as it does with smaller buffer sizes.