CS553 Cloud Computing

Performance

Shruti Gupta (A20381966)

Sagar Mane (A20379756)

System Information:

The project aims to benchmarking different part of computer system, from the CPU, GPU, memory, disk, and network. This benchmarking implemented using KVM virtual machine m1.medium (2 virtual processors with 4GB RAM and 40GB disk) and CentOS 7 Linux for the OS.

```
[cc@pa1-shruti-sagar ~]$ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                2
On-line CPU(s) list:   0,1
Thread(s) per core:    1
Core(s) per socket:    1
Socket(s):             2
NUMA node(s):          1
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 42
Model name:            Intel Xeon E312xx (Sandy Bridge)
Stepping:              1
CPU MHz:               2299.998
BogoMIPS:              4599.99
Hypervisor vendor:     KVM
Virtualization type:   full
L1d cache:             32K
L1i cache:             32K
L2 cache:              4096K
NUMA node0 CPU(s):     0,1
```

Benchmarking result and Analysis:

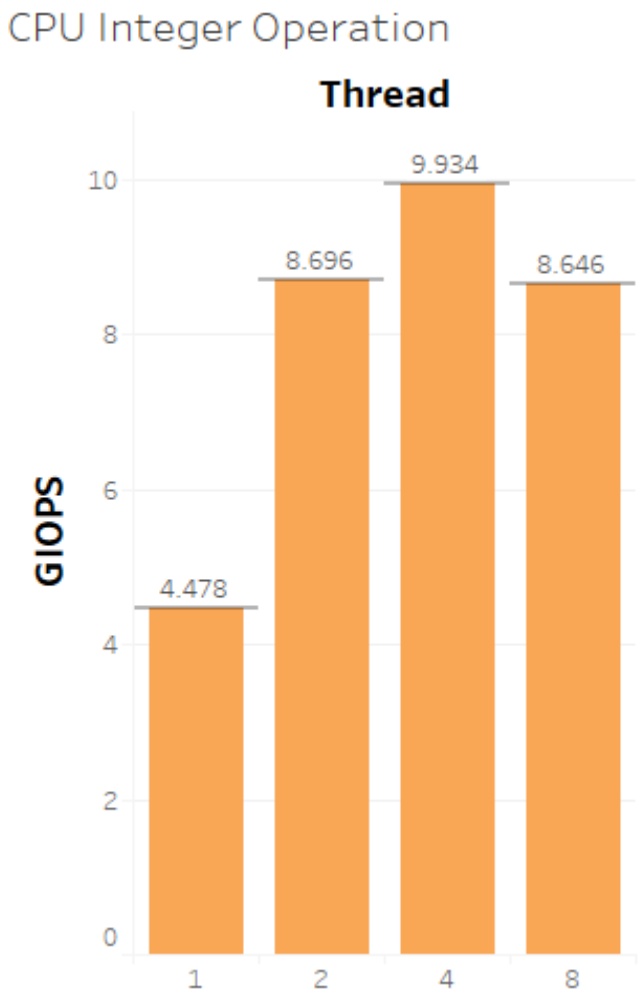
CPU Benchmarking:

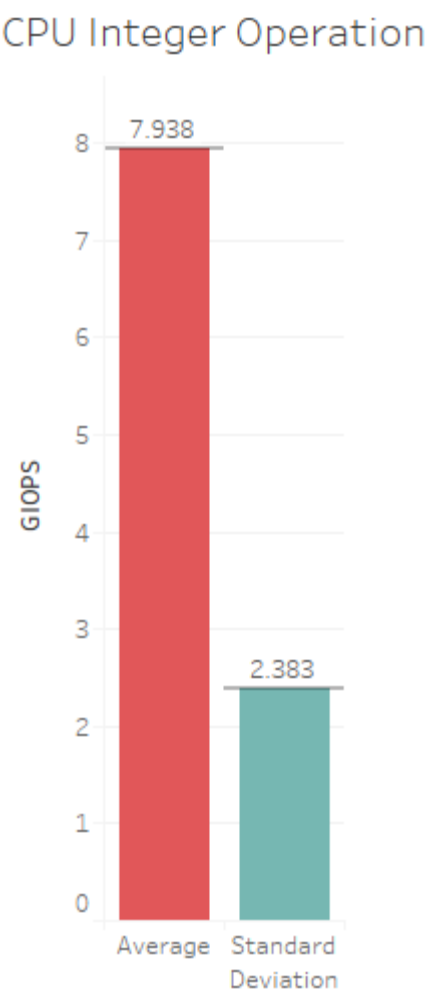
Theoretical Peak Performance:

Number of cores * Average Frequency * cycle per seconds

= 2 * 2.2 * 8

=35.2 GFLOPS

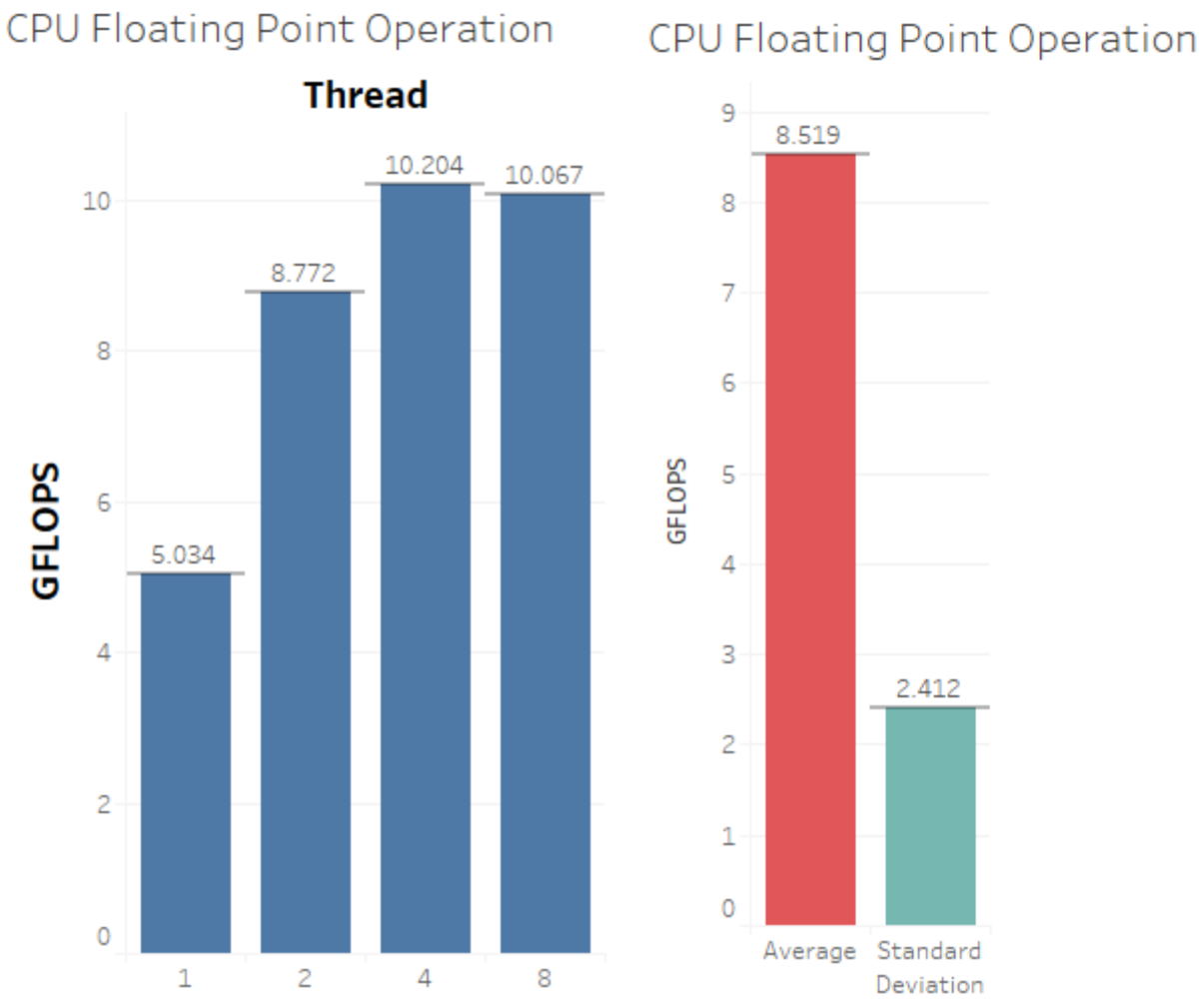




Above plot represents integer operation per seconds (GIOPS) for different threads and average and standard deviation of all threads.

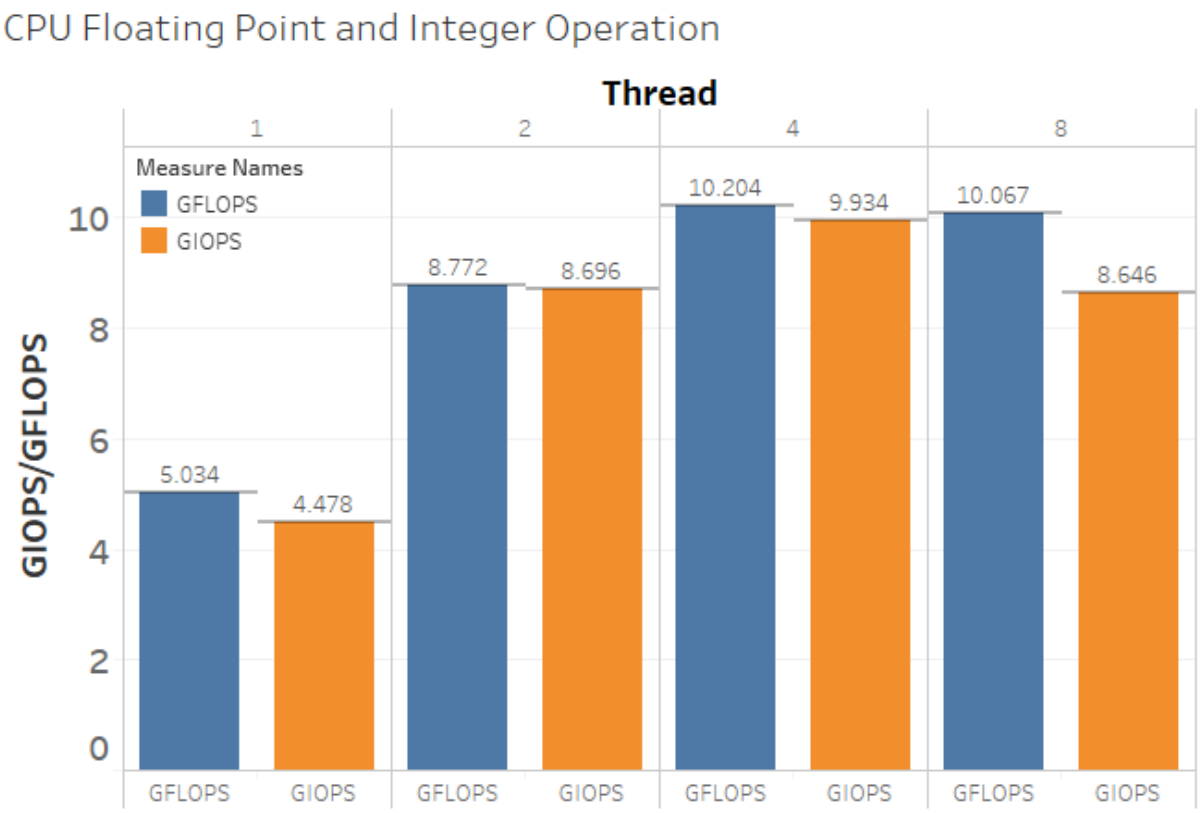
We observed that as number of threads increase, we get more GIOPS. We get average 7.938 GIOPS and standard deviation 2.383 GIOPS.

Optimal number of thread align perfectly with the number of cores (2). Anything less, and the cores aren't being fully utilized. Anything more, and the threads are left waiting in queue to be swapped into context.



Above plot represents floating point operation per seconds (GFLOPS) for different threads and average and standard deviation of all threads.

We observed that as number of threads increase, we get more GFLOPS. We get average 8.519 GFLOPS and standard deviation 2.412 GFLOPS.



Above plot represents comparison between floating point operation and integer operation per seconds (GIOPS) for different threads.

For both we observed that GIOPS/GFLOPS increases with number of thread.

Efficiency with respect to theoretical peak performance:

=10.204/35.2

=29.15%

For AVX instruction we can get more efficiency (76.42%) as compare to CPU operations.

LINPACK output:

```
[cc@pa1-shruti-sagar linpack]$ ./xlinpack_xeon64
Input data or print help ? Type [data]/help :

Number of equations to solve (problem size): 5000
Leading dimension of array: 10000
Number of trials to run: 10
Data alignment value (in Kbytes): 400000
Current date/time: Thu Oct  5 03:30:44 2017

CPU frequency:    3.095 GHz
Number of CPUs: 2
Number of cores: 2
Number of threads: 2

Parameters are set to:

Number of tests                             : 1
Number of equations to solve (problem size) : 5000
Leading dimension of array                  : 10000
Number of trials to run                     : 10
Data alignment value (in Kbytes)            : 400000

Maximum memory requested that can be used = 809800000, at the size = 5000

============= Timing linear equation system solver =================

Size   LDA    Align. Time(s)    GFlops   Residual     Residual(norm)
5000   10000  400000    2.225      37.4831  2.581643e-11 3.599893e-02
5000   10000  400000    2.232      37.3650  2.581643e-11 3.599893e-02
5000   10000  400000    2.244      37.1524  2.581643e-11 3.599893e-02
5000   10000  400000    2.242      37.1988  2.581643e-11 3.599893e-02
5000   10000  400000    2.217      37.6024  2.581643e-11 3.599893e-02
5000   10000  400000    2.156      38.6745  2.581643e-11 3.599893e-02
5000   10000  400000    2.183      38.1978  2.581643e-11 3.599893e-02
5000   10000  400000    2.206      37.7974  2.581643e-11 3.599893e-02
5000   10000  400000    2.203      37.8510  2.581643e-11 3.599893e-02
5000   10000  400000    2.195      37.9946  2.581643e-11 3.599893e-02

Performance Summary (GFlops)

Size   LDA    Align.  Average  Maximal
5000   10000  400000     37.7317  38.6745

End of tests
```

Using LINPACK benchmarking we get 37.7317 GFLOPS.

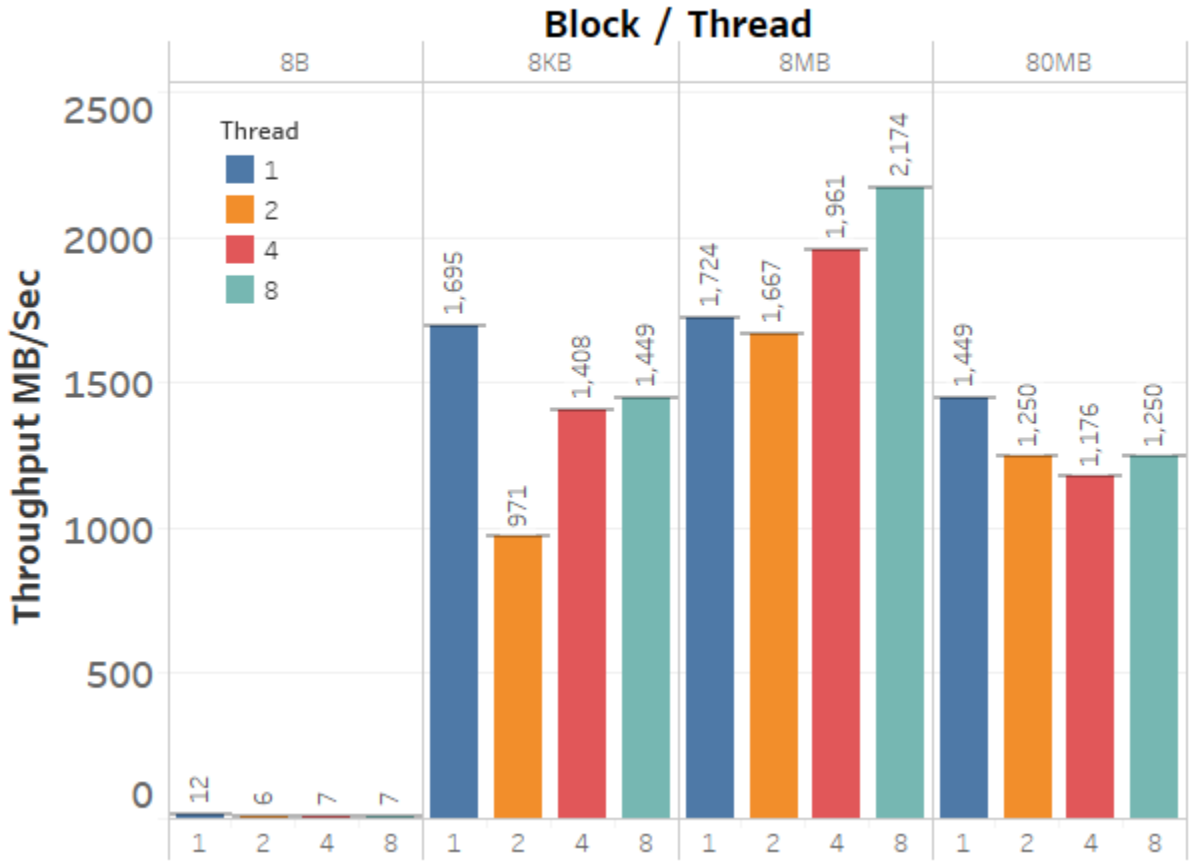
Efficiency with respect to LINPACK benchmark:

=8.772/37.7317 =23.24%

This is not best performance as compare to LINPACK benchmark. We get more better performance if we performing operation using AVX instruction (did VAX implementation for extra credit).

Disk Benchmarking

Write Sequential



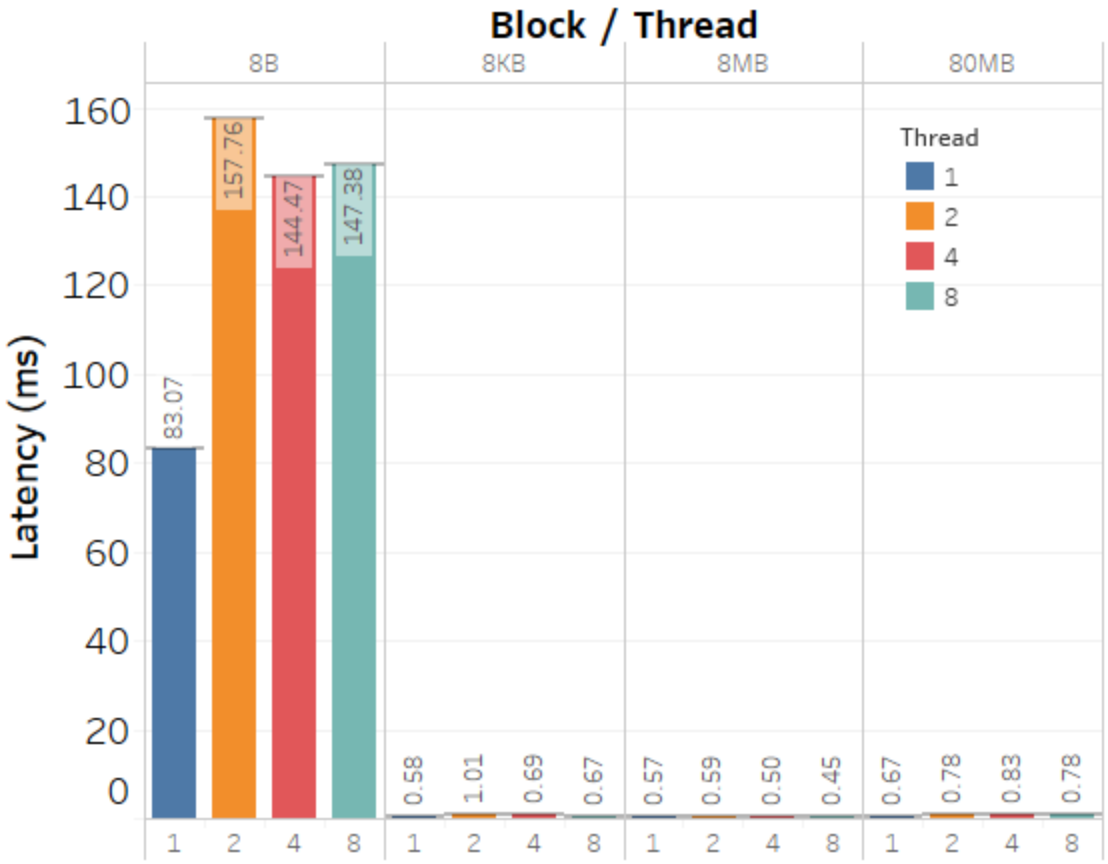
Write Sequential

Thread	Average	Standard Deviation
1	1,220.0	814.9
2	973.4	705.4
4	1,138.1	822.9
8	1,220.0	901.1

Above plot represents the throughput for sequential writing on disk for different block size and number of threads and average and standard deviation of all threads. Throughput is measure in the MegaBytes per seconds.

We observed that we get low throughput for block size 8B as it takes more time to write file compare to other block size.

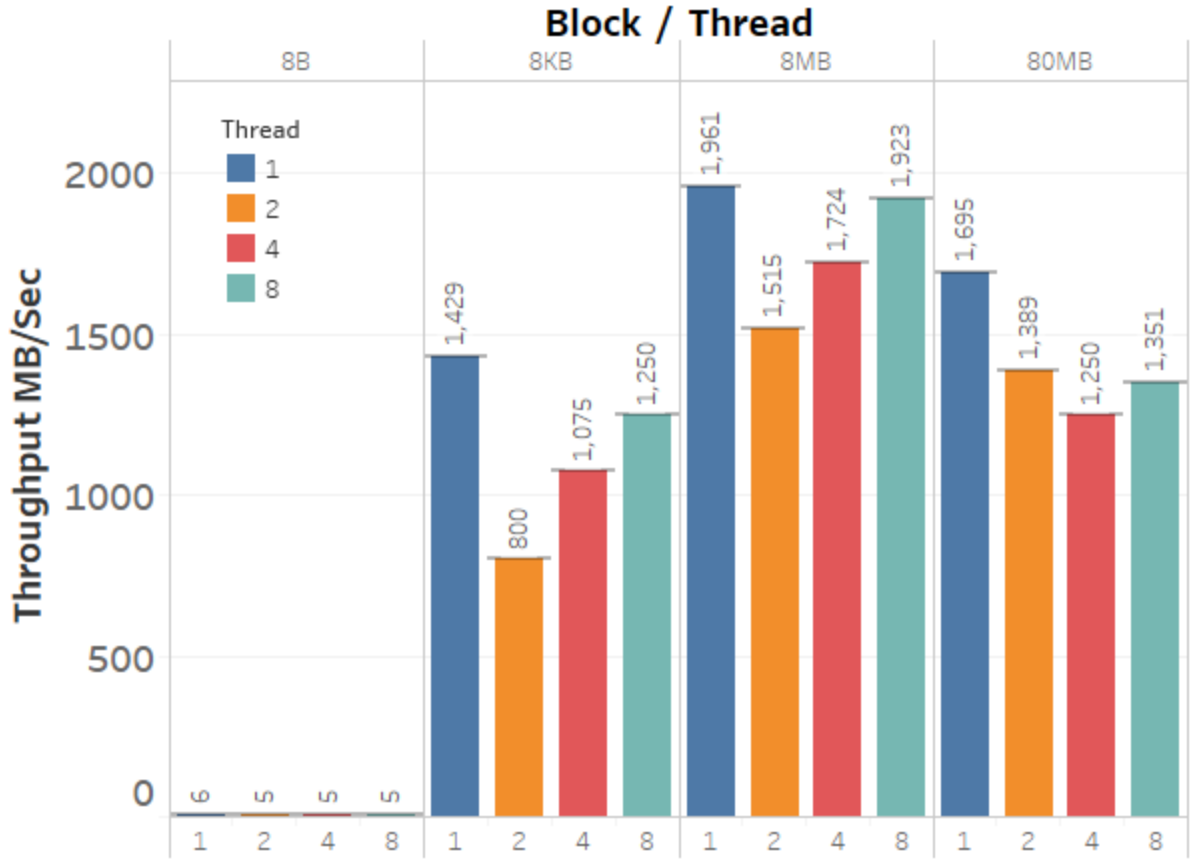
Write Sequential



Above plot represents the latency for sequential writing on disk for different block size and number of threads. latency is measure in milliseconds.

We observed that we get high latency for block size 8B as it takes more time to write file compare to other block size.

Write Random



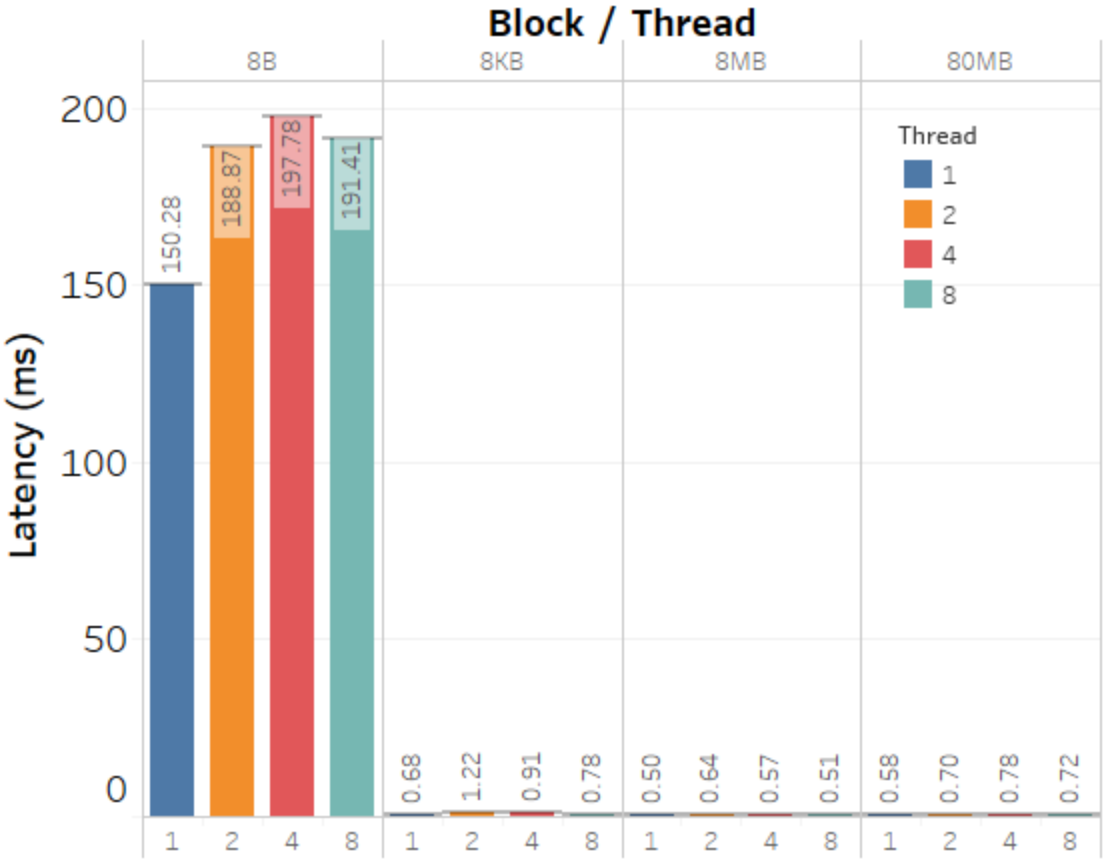
Write Random

Thread	Average	Standard Deviation
1	1,272.7	871.6
2	927.3	689.2
4	1,013.6	726.2
8	1,132.4	807.8

Above plot represents the throughput for random writing on disk for different block size and number of threads and average and standard deviation of all threads. Throughput is measure in the MegaBytes per seconds.

We observed that we get low throughput for block size 8B as it takes more time to write file compare to other block size.

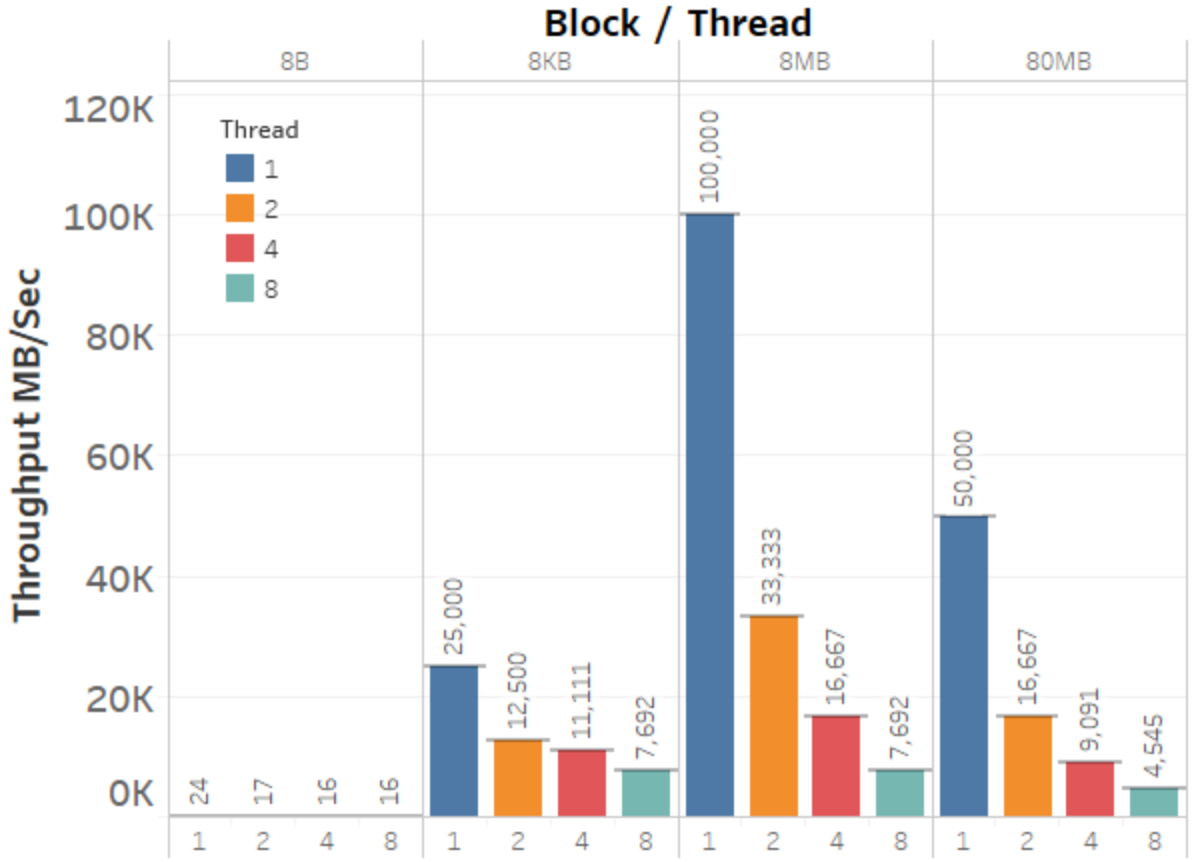
Write Random



Above plot represents the latency for random writing on disk for different block size and number of threads. latency is measure in milliseconds.

We observed that we get high latency for block size 8B as it takes more time to write file compare to other block size.

Read Sequential



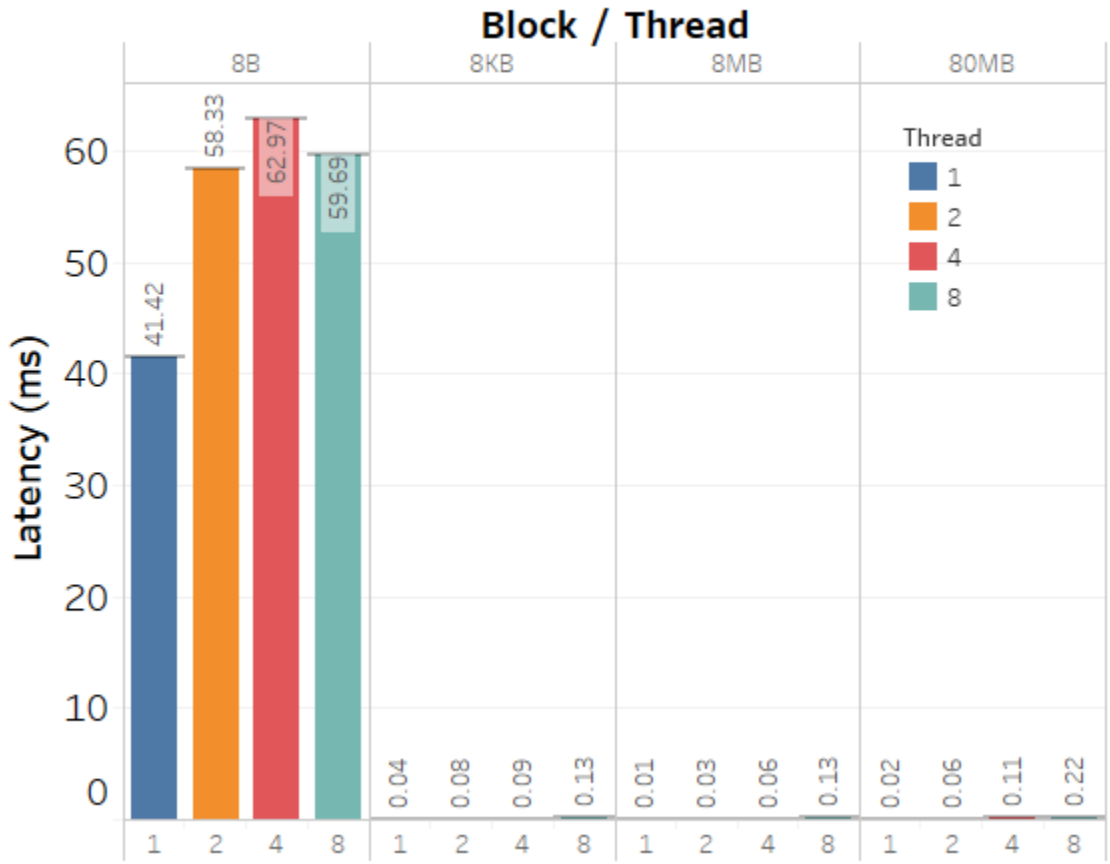
Read Sequential

Thread	Average	Standard Deviation
1	43,756	42,688
2	15,629	13,760
4	9,221	6,923
8	4,987	3,630

Above plot represents the throughput for sequential reading on disk for different block size and number of threads and average and standard deviation of all threads. Throughput is measure in the MegaBytes per seconds.

We observed that we get low throughput for block size 8B as it takes more time to read file compare to other block size.

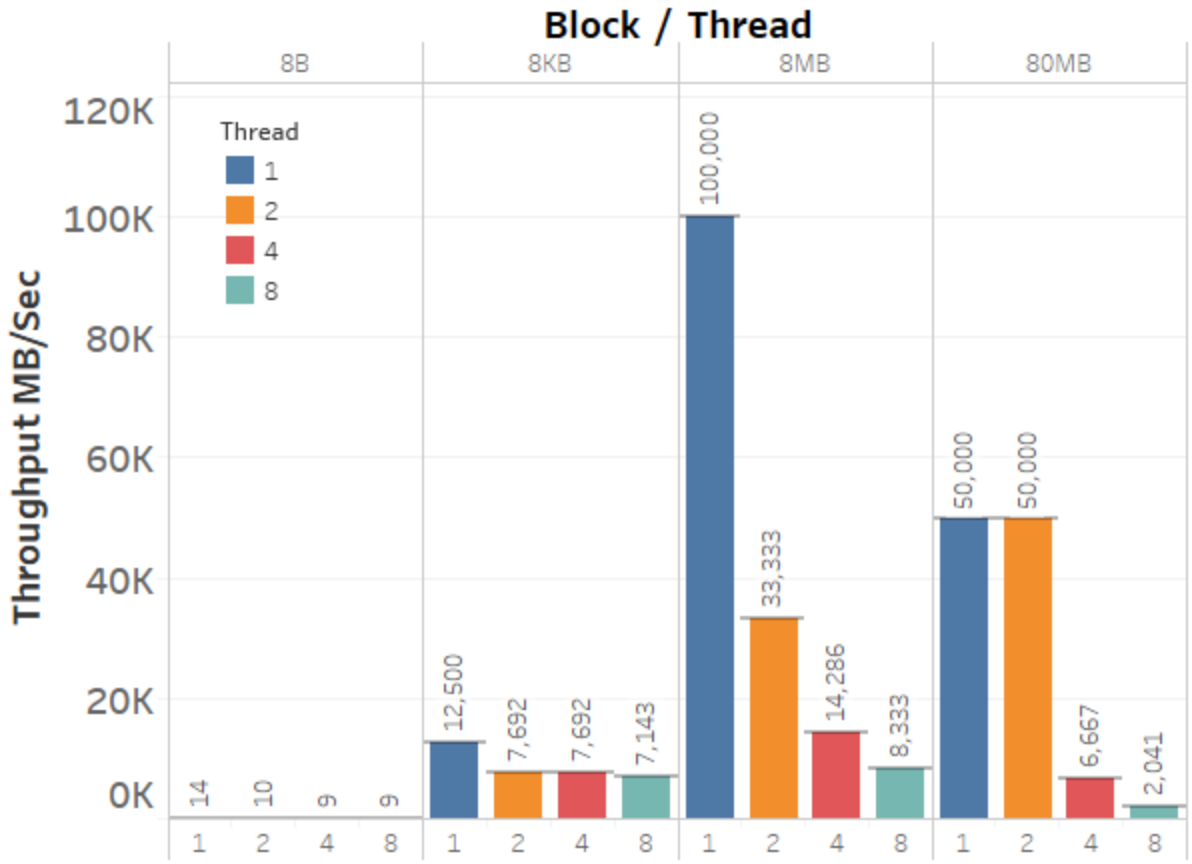
Read Sequential



Above plot represents the latency for sequential reading on disk for different block size and number of threads. latency is measure in milliseconds.

We observed that we get high latency for block size 8B as it takes more time to read file compare to other block size.

Read Random



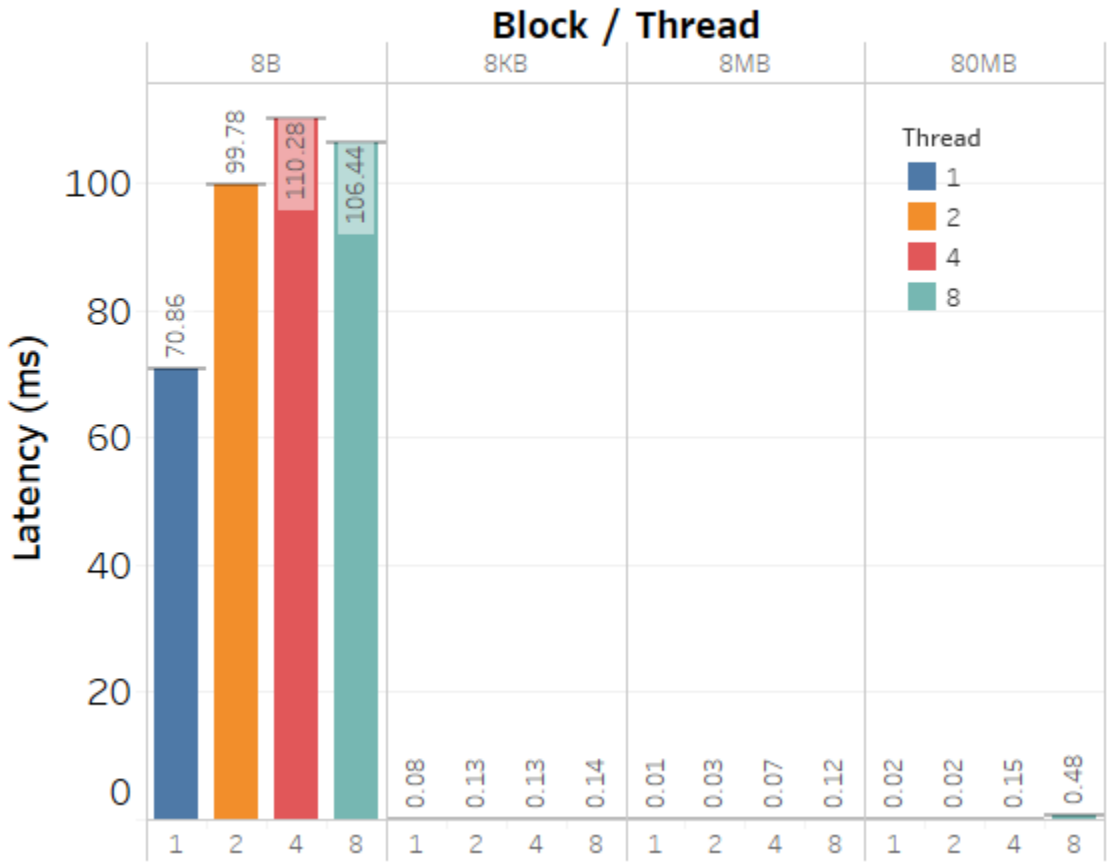
Read Random

Thread	Average	Standard Deviation
1	40,628	44,921
2	22,759	23,083
4	7,163	5,844
8	4,382	3,993

Above plot represents the throughput for random reading on disk for different block size and number of threads and average and standard deviation of all threads. Throughput is measure in the MegaBytes per seconds.

We observed that we get low throughput for block size 8B as it takes more time to read file compare to other block size.

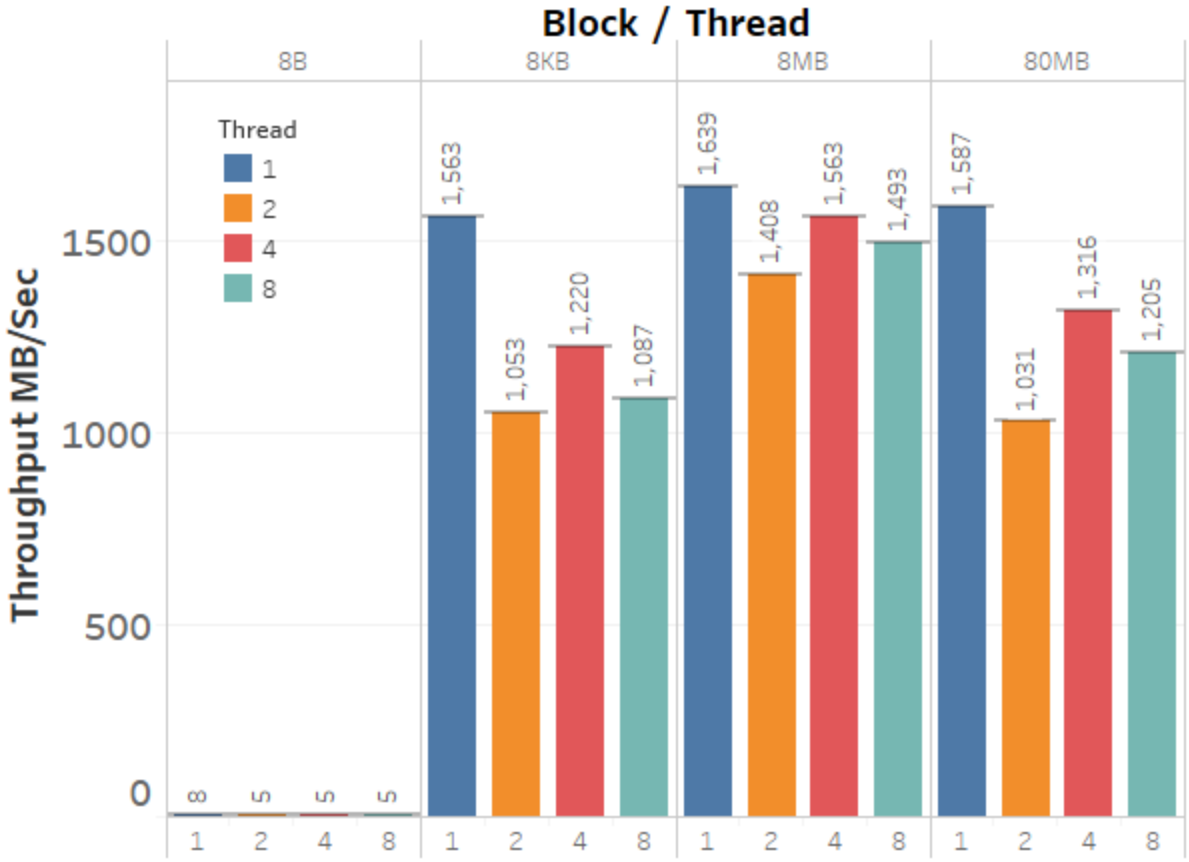
Read Random



Above plot represents the latency for random reading on disk for different block size and number of threads. latency is measure in milliseconds.

We observed that we get high latency for block size 8B as it takes more time to read file compare to other block size.

Read + Write



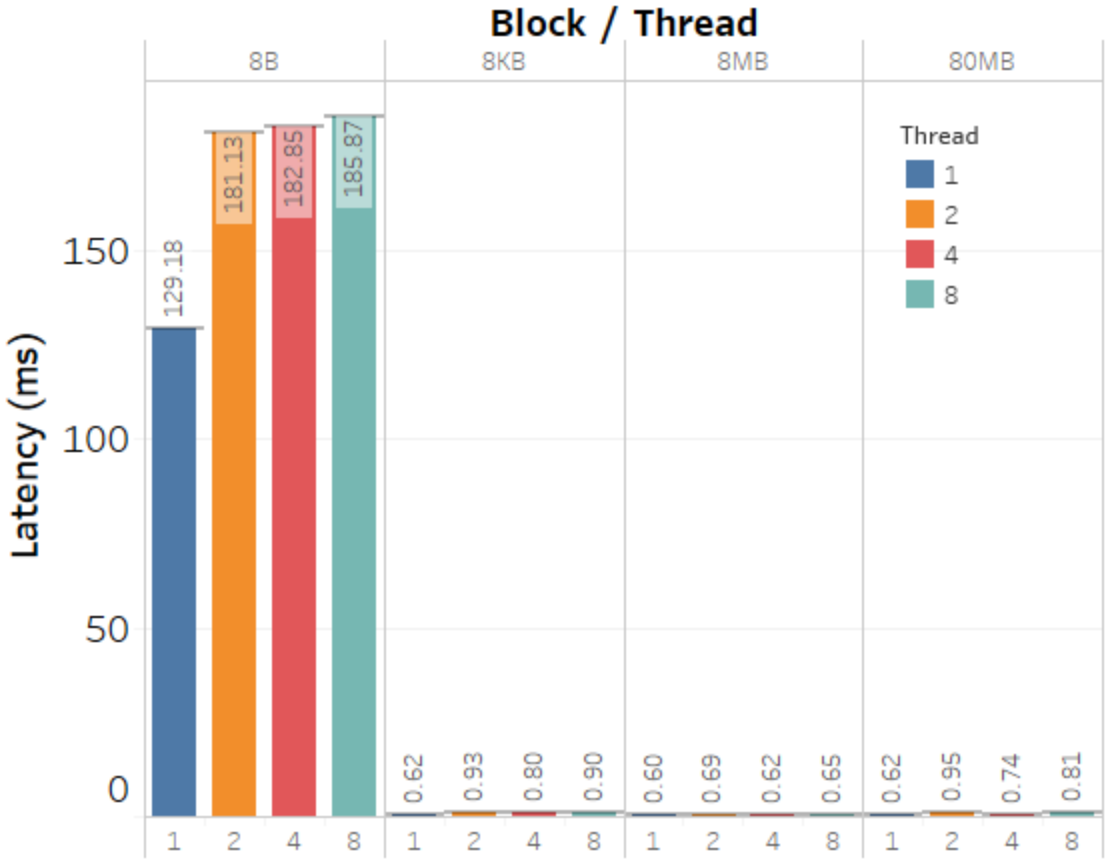
Read + Write

Thread	Average	Standard Deviation
1	1,199.2	795.1
2	874.4	604.6
4	1,025.8	695.5
8	947.4	650.8

Above plot represents the throughput for reading and writing file on disk for different block size and number of threads and average and standard deviation of all threads. Throughput is measure in the MegaBytes per seconds.

We observed that we get low throughput for block size 8B as it takes more time to read and write file compare to other block size. For more concurrency 8 we get better throughput.

Read + Write



Above plot represents the latency for reading and writing on disk for different block size and number of threads. latency is measure in milliseconds.

We observed that we get high latency for block size 8B as it takes more time to read and write file compare to other block size.

We observed that for concurrency 8 we get optimal number of performance as compare to another concurrency.

IOZone Benchmark output:

```
[cc@pa1-shruti-sagar current]$ ./iozone -a
        Iozone: Performance Test of File I/O
                Version $Revision: 3.471 $
                Compiled for 64 bit mode.
                Build: linux

        Contributors:William Norcott, Don Capps, Isom Crawford, Kirby Collins
                     Al Slater, Scott Rhine, Mike Wisner, Ken Goss
                     Steve Landherr, Brad Smith, Mark Kelly, Dr. Alain CYR,
                     Randy Dunlap, Mark Montague, Dan Million, Gavin Brebner,
                     Jean-Marc Zucconi, Jeff Blomberg, Benny Halevy, Dave Boone,
                     Erik Habbinga, Kris Strecker, Walter Wong, Joshua Root,
                     Fabrice Bacchella, Zhenghua Xue, Qin Li, Darren Sawyer,
                     Vangel Bojaxhi, Ben England, Vikentsi Lapa,
                     Alexey Skidanov.

        Run began: Thu Oct  5 03:53:09 2017

        Auto Mode
        Command line used: ./iozone -a
        Output is in kBytes/sec
        Time Resolution = 0.000001 seconds.
        Processor cache size set to 1024 kBytes.
        Processor cache line size set to 32 bytes.
        File stride size set to 17 * record size.
```

kB	reclen	write	rewrite	read	reread	random read	random write	bkwd read	record rewrite	stride read	fwrite	frewrite	fread	freread
64	4	695778	1421755	5860307	4564786	4274062	1991276	2772930	1991276	3057153	1183548	1690338	2772930	4274062
64	8	1143223	2203800	5860307	6271021	5860307	1780008	2561267	2379626	3791156	2133730	1484662	3541098	2801873
64	16	1049372	2067979	7100397	6421025	6421025	2467108	2923952	2467108	2561267	2006158	2772930	4274062	7100397
64	32	1066042	2561267	7100397	9006179	7100397	2662899	3791156	1828508	3738358	2298136	2892445	4643754	7100397
64	64	1017549	2467108	9006179	9318832	7100397	1734015	2923952	2561267	3791156	2467108	2923952	4274062	5735102
128	4	743788	1732927	3560017	5784891	4267461	2035099	2843510	2660335	3445772	1967960	1684006	3657016	4407601
128	8	1142750	2409592	6727225	7176872	5784891	2511022	4012317	3199360	5603747	2367096	1939522	4889281	6114306
128	16	1152564	2453642	6727225	7082197	6114306	2843510	4135958	3106789	4717434	2464907	2166499	4934216	6114306
128	32	1281909	2608629	6406138	7582312	6727225	3036502	4012317	3445772	4934216	2511022	3124872	5325799	6406138
128	64	1294270	2714132	7582312	7917784	7082197	3199360	4135958	3468030	4934216	2326073	4135958	4889281	6727225
128	128	1391557	2453642	3895854	8548124	6727225	3380677	4267461	3380677	4889281	3277486	3359523	5603747	7082197
256	4	1080434	1911233	5117791	5022044	3197509	1907837	3316006	2692393	3197509	2097948	2131261	4422226	3950402
256	8	1332507	2187712	6936061	6554972	5569035	2726577	4332998	3879045	5117791	2533571	2205688	5687020	5320671
256	16	1256123	2097948	5810112	5841722	5217259	2148318	4332998	3326279	5320671	2148318	2533571	5117791	5428265
256	32	1241598	2563818	6213578	6398720	5810112	2975955	3511189	3499745	5428265	2588541	2639446	5117791	6073004
256	64	1274008	2872459	7120034	7364196	6719046	3368013	5217259	3935921	6107548	2880164	3368013	5569035	6891544
256	128	1471270	3009318	6891544	6107548	5687020	3454704	5242734	3285566	5117791	2463808	3756894	5687020	6595234
256	256	1235882	2783115	6398720	6554972	6398720	2880164	4907284	2692393	5569035	3285566	3368013	5428265	6249745
512	4	1002183	1619757	4228947	4340054	3905895	1619757	2764505	2391924	3659616	1895721	1910902	4038098	3970896
512	8	1125040	2265742	6736026	5822804	5624545	2668325	5019764	3304808	4457157	2600470	2560168	4973263	6472112
512	16	1319250	2753870	6571132	6736026	4701087	3046915	5227492	3822466	5453155	2497638	2509312	5227492	6414119
512	32	1004057	2845081	6571132	6736026	6319739	3086326	5019764	3504346	6319739	2497638	2521095	5278892	6551087
512	64	1551867	2584820	6821616	6931711	7022379	3029721	5886650	3220553	6736026	2524058	3158966	4973263	6821616
512	128	1546280	3012719	6736026	6821616	6472112	3653390	4882800	4271001	5227492	3325278	3510074	5566231	6174377
512	256	1365383	2246778	5495016	5566231	5384786	2625909	5164632	3163620	5509113	2124526	4131319	5008057	5886650
512	512	1433753	2722449	6018636	6018636	5822804	2922519	5214798	3029721	5439343	3201349	2722449	4650188	5951911

1024	4	982873	1615374	4146499	4270175	3425544	1777861	3379719	2301937	3229770	1868347	1723636	3866533	3567824
1024	8	1230718	2169388	6128613	6393168	5336651	2572135	4995276	3951918	5120336	2270300	2188177	4943530	4720752
1024	16	1253344	2410457	5845038	6479979	4826859	2926114	4594502	3544270	4972145	2565989	2480051	5500686	6244448
1024	32	1212991	2409105	5200941	5529011	5226256	3020783	4949226	3955557	5096034	2730767	2730767	4926518	6244448
1024	64	1395010	2565989	6559147	6650556	5853003	3379719	5144870	3908759	6569179	2835325	2753527	5507740	5950309
1024	128	1406431	2798378	5690162	4995276	6128613	3251778	4943530	3966517	5950309	2746483	2993413	5821271	5623115
1024	256	1298041	2290886	4720752	5144870	5917516	3037876	5096034	3518141	5251818	3210456	3239514	5169641	5789881
1024	512	1389594	2455943	5564829	5630486	6025439	3083680	5500686	3433760	5782087	2645000	3891053	5065980	6025439
1024	1024	1416169	2723840	6208343	5813392	4317393	3314514	4995276	3094790	5728107	2852271	2900425	5251818	6016997
2048	4	958291	1959469	4741090	4686767	3618634	1972971	2964289	2412357	3319389	1882186	1785550	4257067	4172290
2048	8	1226291	2356760	6185123	5447681	4945823	2537792	4656281	3579429	3946119	2364545	2343259	5868220	5389574
2048	16	1257897	2547576	5719737	5900467	5349298	2901215	5674397	3611028	5430461	2429413	2458618	5785224	5290001
2048	32	1344739	2909075	6711905	5549749	5689430	3074635	5800851	3644736	5029807	2500122	2642401	6080053	5319484
2048	64	1454736	2916978	5970183	6167360	5280246	3121562	5521212	4385291	6167360	2805512	2656292	6075753	6092991
2048	128	1423398	2723672	5640860	5332694	5754221	3448657	5578583	3929871	6583305	2691246	2771120	10339282	11917129
2048	256	2528082	6041567	9567698	12561952	8608824	7293167	5119743	3915540	5785224	2809182	2855880	5489457	5754221
2048	512	1450315	2756890	5674397	6024617	5884299	3225896	5820505	3443127	10780463	5416763	6280083	6114678	9708272
2048	1024	1729816	5461535	11502234	11770166	12118884	6397004	9441504	6629029	11015480	4697018	6780783	11579763	12561952
2048	2048	1867050	6523312	12979541	10490809	11564174	6623917	4697018	3341339	5986827	3340040	3210223	5820505	5800851
4096	4	1019951	2032926	4946971	5050220	4100397	2160499	3949571	2834585	3957760	2183565	2140045	4773769	4633432
4096	8	1405213	2588965	6041154	5468151	4729091	2353791	4681411	3540193	4964124	2482709	2422154	5454263	5841891
4096	16	1329011	2608620	6291135	5497899	5424983	2923334	5589120	3660115	5580043	2587405	2556602	6041154	6225028
4096	32	1259539	2551665	5971855	6178018	5291314	3000952	4887857	3761077	5404504	2612587	2622558	6060333	5657217
4096	64	1396532	3119752	6522830	6545196	6693092	3747131	6142675	4638436	6311938	2996241	3002525	6480999	6471234
4096	128	1538614	3214912	6389405	6142675	6420447	3775126	6015769	4718699	5794602	2917377	2925823	5681538	6522830
4096	256	1520906	2946902	6401308	6468797	6449370	3673419	6060333	4043458	6628532	2927818	2826656	5736555	6439700
4096	512	1444681	3040789	6263611	6216019	6234064	3625358	6024207	4132949	6272759	3045101	3065750	5810280	6160296
4096	1024	1474944	3080040	6815229	6815229	6769576	3853020	6439700	4432592	6439700	3424463	3706707	5759634	6782940
4096	2048	1599346	3311586	6727163	7038605	6772244	3882626	5961494	4266376	6716643	2561939	4192455	5596402	6225028
4096	4096	1447358	6015769	11409087	11907264	10612681	6615769	11252164	7038605	12606249	6847827	7158858	10860922	12450933

8192	4	2071885	1881405	4259926	4459520	3449493	1877190	3455043	2443827	3526677	1960746	2035311	4327528	4605365
8192	8	1329463	2555994	6330318	6282859	5542961	2685864	5375591	3959496	5392464	2676241	2610564	6425016	6325656
8192	16	1474173	3246751	7468200	7779318	7093541	3591558	5406890	4061528	5760412	2784463	2746185	6547448	6244039
8192	32	1554120	3497599	7419818	7487730	7179509	4031505	7049878	4929091	7148143	3346356	3236658	6965556	7522154
8192	64	1645596	3614985	7591960	7728574	7678487	4273702	7407022	5301769	7543623	3335636	3394960	7707769	7765253
8192	128	1708776	3541946	7036884	7407022	7281448	4318282	7320230	5475825	7111158	3273040	2526115	6086965	5979968
8192	256	1371930	2965931	5863631	6186701	6244039	3683582	6192276	4220680	5834754	2880400	2817109	5992483	6307078
8192	512	1524943	2916338	5701152	5872650	5915112	3416226	5531360	3980135	5940679	2768534	2713438	5905961	5928379
8192	1024	1446554	3112349	6108608	6415419	6163396	3666679	6091282	4368796	6263389	3045318	3144824	6254268	6360787
8192	2048	1563596	3304836	6866721	6965556	6906749	3951754	6692839	4540851	5454094	2818958	2986555	5825851	6022946
8192	4096	1316828	2886449	6001904	6045199	6036702	3436040	5793435	3715045	5773964	2280038	3934558	5838720	5630154
8192	8192	1344708	3090236	6496691	6495463	6445505	3682003	6350207	3701837	6336155	3695070	3637565	6331484	6385612
16384	4	1102549	1714865	4620478	4973975	4062441	2151530	3211820	2528421	3754381	2197700	3793343	4111293	4224781
16384	8	1229949	2508210	6318564	6527426	5552278	2732079	5709112	4294482	5923170	2862857	2723417	5736755	5396617
16384	16	1275824	2662740	7117644	7173368	6298872	3426925	6983824	4617994	6445388	3176338	2338605	6372469	6532390
16384	32	1236522	2493376	5145674	5416182	5835154	3246874	5613049	4074485	5824273	2669255	2579772	5889161	5971034
16384	64	1370693	2805929	6019150	6142904	6243929	3489041	6335457	4783544	6657054	3025582	3147100	6948516	7083163
16384	128	1311670	2708070	5467027	5655548	6106332	3598480	5692559	4166886	5083250	2485081	2479432	5531274	5637917
16384	256	1402730	2583457	5663472	5775325	5847069	3685124	5942635	4491232	6289072	2791680	2347633	5661139	5841105
16384	512	1329099	2582195	6166607	6682300	6761860	3901898	5331717	4131065	5847567	2798844	3023985	6861103	6945707
16384	1024	1821261	2559212	10182464	10722665	11881359	3433260	5352482	4107362	5975188	2763391	2920404	6030242	6365386
16384	2048	1551341	2562074	5261503	5586129	6065906	3454315	6215691	4295824	6362439	2850862	2768958	5929815	6643539
16384	4096	1562914	3267096	6872081	7249803	7336490	4117698	5273212	3975715	5979867	2997995	2863334	5783101	6551072
16384	8192	1563341	2890675	4801929	5616719	5634219	3124350	4716244	3474926	5520609	2617206	3754381	5268360	6028127
16384	16384	1421974	2187208	4942494	5291891	5398313	3064713	4159571	2805471	4191283	2662327	2466351	4492406	5104015

32768	64	1515634	2298356	4949757	5413581	5985025	2986257	5017337	4448034	5556724	2373983	2203048	6081691	6377439
32768	128	1337803	2428129	4743558	5219305	5498040	3074847	4979166	4554909	5435849	2536625	2191107	5271554	5586993
32768	256	1569425	2316563	4431683	4967827	5578376	3237910	5300423	4808280	5885077	2777397	2123533	4512882	4875484
32768	512	1493190	2250669	4525364	4995090	5270544	3035191	4741431	4413184	5670203	2687251	2154222	4584231	4898770
32768	1024	1429231	2301781	4617033	5064669	5641806	3204690	4784345	4078572	5639722	2544279	2146048	4560047	4938020
32768	2048	1483183	2256915	4625579	4976822	5244599	2855351	4714918	4075186	6354146	2442282	2101840	4796199	5254625
32768	4096	1547477	2263270	4706683	5115186	5552908	2965190	5140055	4660400	6853851	2608503	2131107	4507997	4943881
32768	8192	1425067	2273491	4892841	5360374	5382416	3002633	4681832	3985377	6465644	2871338	2471532	4542264	4837218
32768	16384	1395520	3276427	6422439	6553510	6534193	3688863	6057032	4583467	5586085	3783819	3790812	4593270	6289304
65536	64	2136122	2315194	5306543	5609546	5913157	2850874	4959206	4518208	5514121	2525388	2224510	5129677	5413987
65536	128	1630289	2361234	4938623	5249187	5392532	2983917	4903472	5202492	5894644	2709897	2198604	5108038	5564802
65536	256	1744560	2326047	4642987	4798694	5130060	2940285	4615545	4752728	5432390	2586993	2157990	4546381	4743461
65536	512	1627085	2340147	4657463	4981855	5279635	3035888	4600251	4511312	5303574	2543615	2157990	4644321	5028425
65536	1024	1700684	2374841	4698221	4995890	5223253	2982557	4674809	4770379	5682958	2644943	2217565	4747885	4975003
65536	2048	1700684	2457123	4720975	4985560	4792754	2611497	4747229	4409348	4733495	2201862	2108690	4801880	5114787
65536	4096	1708443	3718797	4662519	4960279	5126042	2892148	4852658	4954022	7856167	2676674	2098099	4766988	5011556
65536	8192	1647338	2548544	5068573	5581866	5789486	3138527	5001981	5159913	6975037	2550436	2269021	4830571	5010004
65536	16384	1550601	2323648	4461231	4828111	4442485	2608820	4220786	3285662	4436820	2736252	2479799	4316427	4768972

```
 65536   16384  1956061  2525046  4461251  4828111  4442483  2066820  4220780  5285062  4450820  2750252  2479799  4310427  4768972
131072      64  1769998  2491474  5159322  5853210  6188796  3063785  5071650  5992946  6448460  3058007  2376173  5231793  5891410
131072     128  1958107  3638169  7100696  7401446  7416523  3773157  4970496  6151884  6227001  2959609  2454488  5003520  5517653
131072     256  1871389  2509122  5040819  5694366  5905523  3371973  4912534  5821538  5967057  2942186  2377334  5125985  5457456
131072     512  1957584  2435006  4359895  4817441  4984105  3040620  4413801  5646234  5298146  2803206  2303435  4417525  4730636
131072    1024  1919626  2415311  4419301  4900797  5195501  3200854  5070855  5539112  6178294  2939889  2431215  5050729  5492680
131072    2048  2076549  2523519  5059142  5614519  6007550  3284265  5083891  5887687  6368235  2819060  2283847  5306328  5865826
131072    4096  2119256  2548400  5219920  4826280  4795379  2972089  4862092  5591847  6303396  2709213  2164582  4914202  5359298
131072    8192  1989225  2554463  5008899  5553660  6068962  3488280  5245370  5963821  8813465  2820637  2277611  5152601  5613888
131072   16384  1977099  2374541  4699096  5217196  5387026  3526182  4491256  3898955  5681243  2898977  2312620  4643490  5075958
262144      64  2148424  2827599  6223610  6912350  7509771  3326824  5580036  8638743  8029863  3492976  2779451  5773599  6651017
262144     128  2419732  2860772  5671070  6523092  7121890  3238870  6169551  9974269  7779207  3434076  2706391  5974249  6602134
262144     256  2362083  2641115  6126716  7540880  7513209  4020860  5281547  7966447  7041795  3312951  2723884  5280786  6028361
262144     512  2264234  2891383  5383513  6237946  7246558  3906363  5429424  7343795  7559804  3392310  2600893  7656936  7733519
262144    1024  2372107  2758947  6068453  5209948  6177246  3700716  5689263  8029629  7400988  3223374  2761857  5700028  6375576
262144    2048  2394402  2950411  7763278  7710578  7392479  3730726  5658083  8288829  7438188  2967940  2844534  6155769  7468249
262144    4096  1972222  2937319  5640291  6467418  7119399  3720035  5805548  7897326  7413163  2833436  2410079  5455663  6236177
262144    8192  2211456  3108073  5449417  6553418  5376537  3395872  5660384  7590229  7682992  2909154  2391257  5666014  6366900
262144   16384  2160285  3127123  5706862  5714426  4513264  2952867  5131123  4475967  4595798  2577366  2365259  4795208  5394686
524288      64  2101033  3322907  8042221  7832540  7666701  3310755  5750286 10330803  7841114  3359098  3055680  6258661  7546775
524288     128  1824511  2614564  6445009  6190755  7460086  3686798  5759126 10188644  7590015  3249763  3076319  6348864  7869370
524288     256  1984202  2691764  5178256  6627580  5184042  3209813  5333114  8185707  7127933  3155056  2951923  5920953  6011647
524288     512  2020853  2828606  5632847  6295716  4809985  2976980  5568644  8713737  6467890  3183775  2715001  5376429  6216885
524288    1024  2276353  2982451  5279681  6254638  6376183  3110152  5605323  8211505  7130938  3112295  2921995  5434721  6504488
524288    2048  1943888  2793670  7106006  7214647  7155581  3377403  5866020  7913548  7311632  2872592  2572321  6208040  7355630
524288    4096  1941403  3008915  6181758  7100705  7103802  3657578  5949757  7985854  6093185  2459898  2611319  6091834  7431447
524288    8192  2093050  2819814  6229584  7431748  7649952  3666066  5928232  8121585  5526450  2654219  2611620  6228401  6655260
524288   16384  2195427  2759483  5395306  5737323  5540290  3101672  5065685  4806494  5437261  2582674  2469864  5306922  5851939

iozone test complete.
```

IOZone benchmark for 1MB write sequential is 2071885Kb/s and we get 1569 Mb/sec

Efficiency is

= 1667/2071.885

=80.45%

IOZone benchmark for 1MB write random is 4031515 KB/s and we get 1616 Mb/sec

Efficiency is

= 1667/2071.885

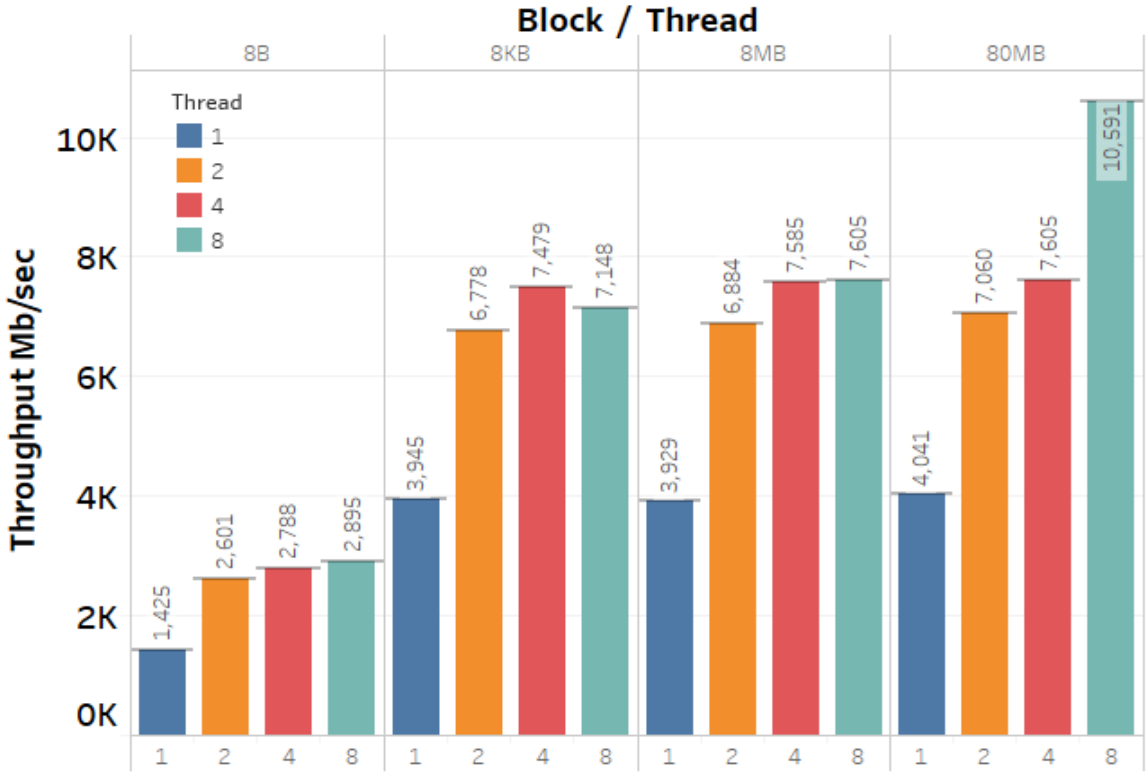
=40.08%

We observed that for sequential write we get better performance as compare to random write.

According to result disk we are evaluating is spinning hard drive (HDD).

Memory Benchmarking:

Write Sequential



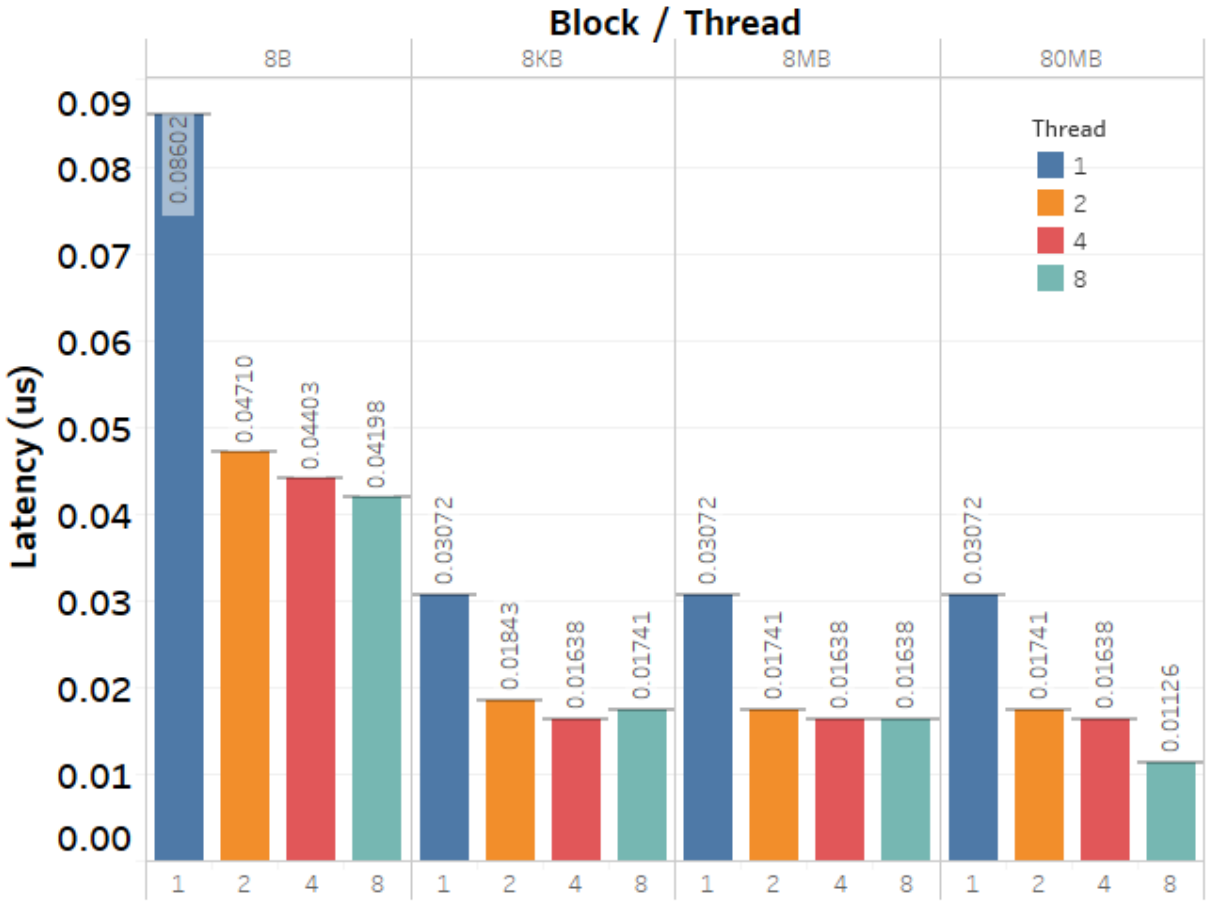
Write Sequential

Thread	Average	Standard Deviation
1	3,335	1,274
2	5,831	2,157
4	6,364	2,385
8	7,059	3,169

Above plot represents the throughput for sequential writing on memory for different block size and number of threads threads and average and standard deviation of all threads. Throughput is measure in the MegaBytes per seconds.

We observed that we get low throughput for block size 8B as it takes more time to write memory as compare to other block size.

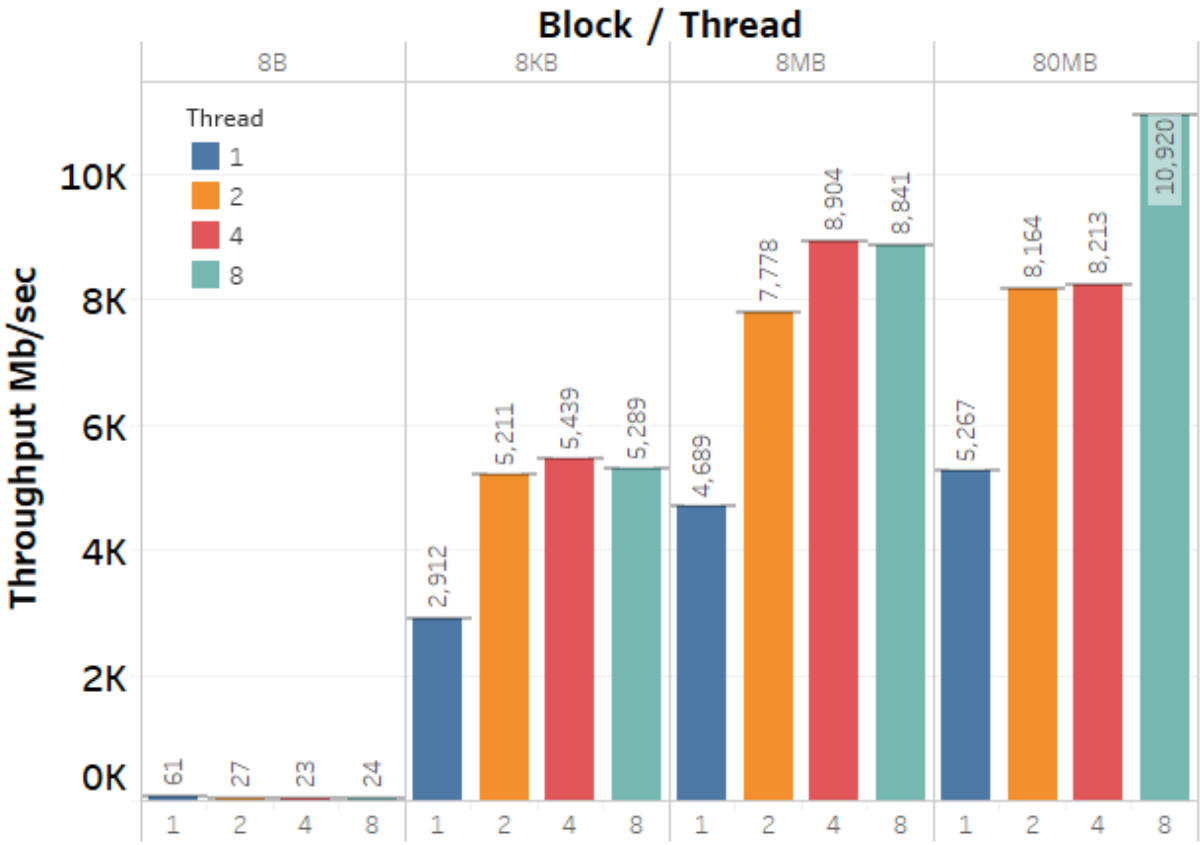
Write Sequential



Above plot represents the latency for sequential writing on memory for different block size and number of threads. latency is measure in microseconds (us).

We observed that we get high latency for block size 8B as it takes more time to write memory compare to other block size.

Write Random



Write Random

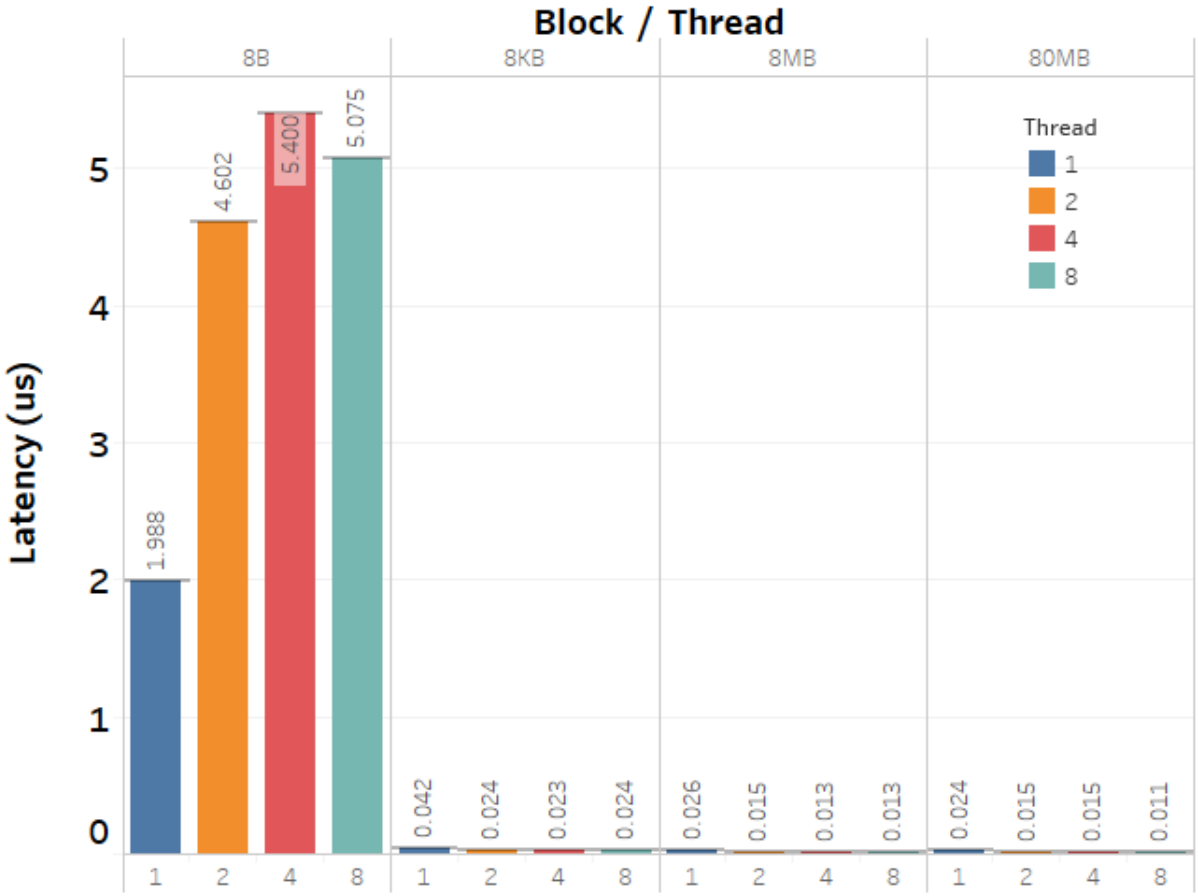
Thread	Average	Standard Deviation
1	3,232	2,339
2	5,295	3,749
4	5,645	4,036
8	6,269	4,768

Above plot represents the throughput for random writing on memory for different block size and number of threads threads and average and standard deviation of all threads. Throughput is measure in the MegaBytes per seconds.

We observed that we get low throughput for block size 8B as it takes more time to write memory as compare to other block size.

We also observed that random throughput of random write with 8B is low as compare to sequential write.

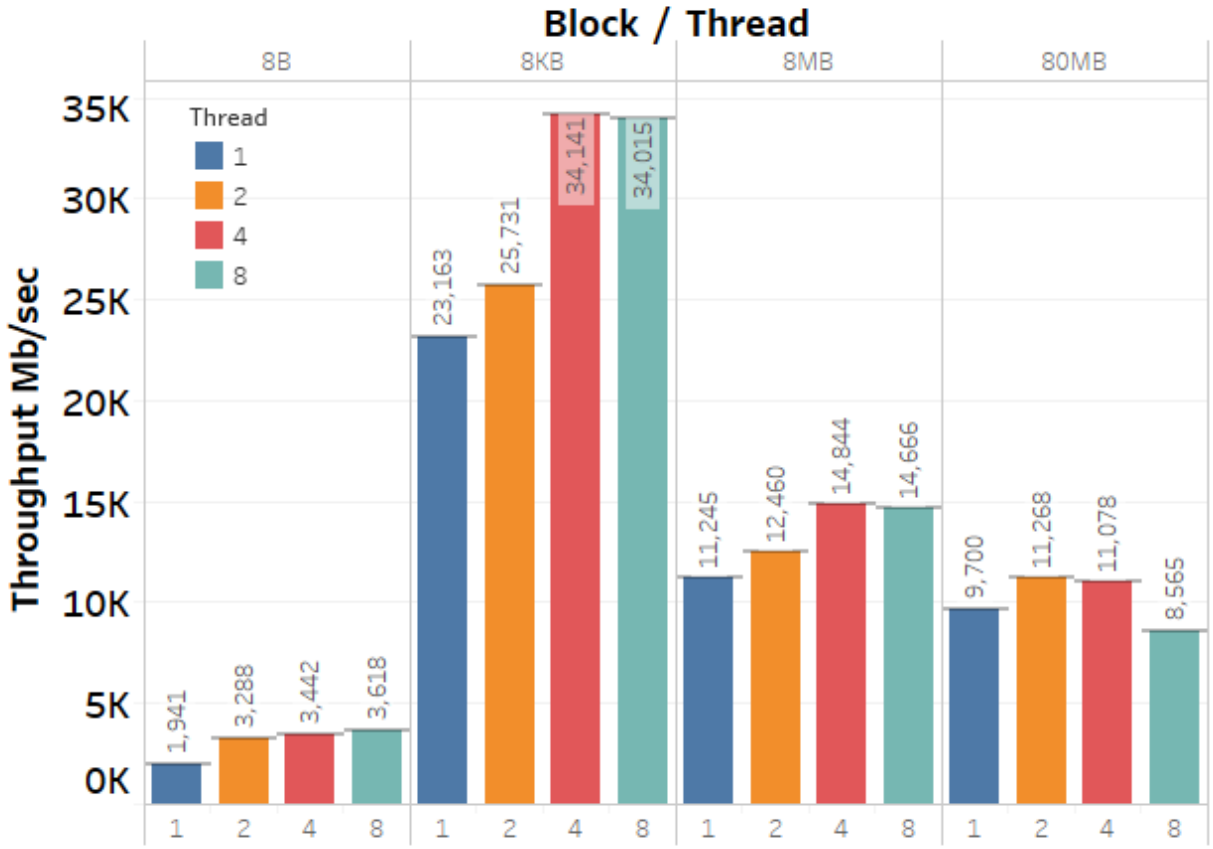
Write Random



Above plot represents the latency for random writing on memory for different block size and number of threads. latency is measure in microseconds (us).

We observed that we get high latency for block size 8B as it takes more time to write memory compare to other block size.

Read Sequential



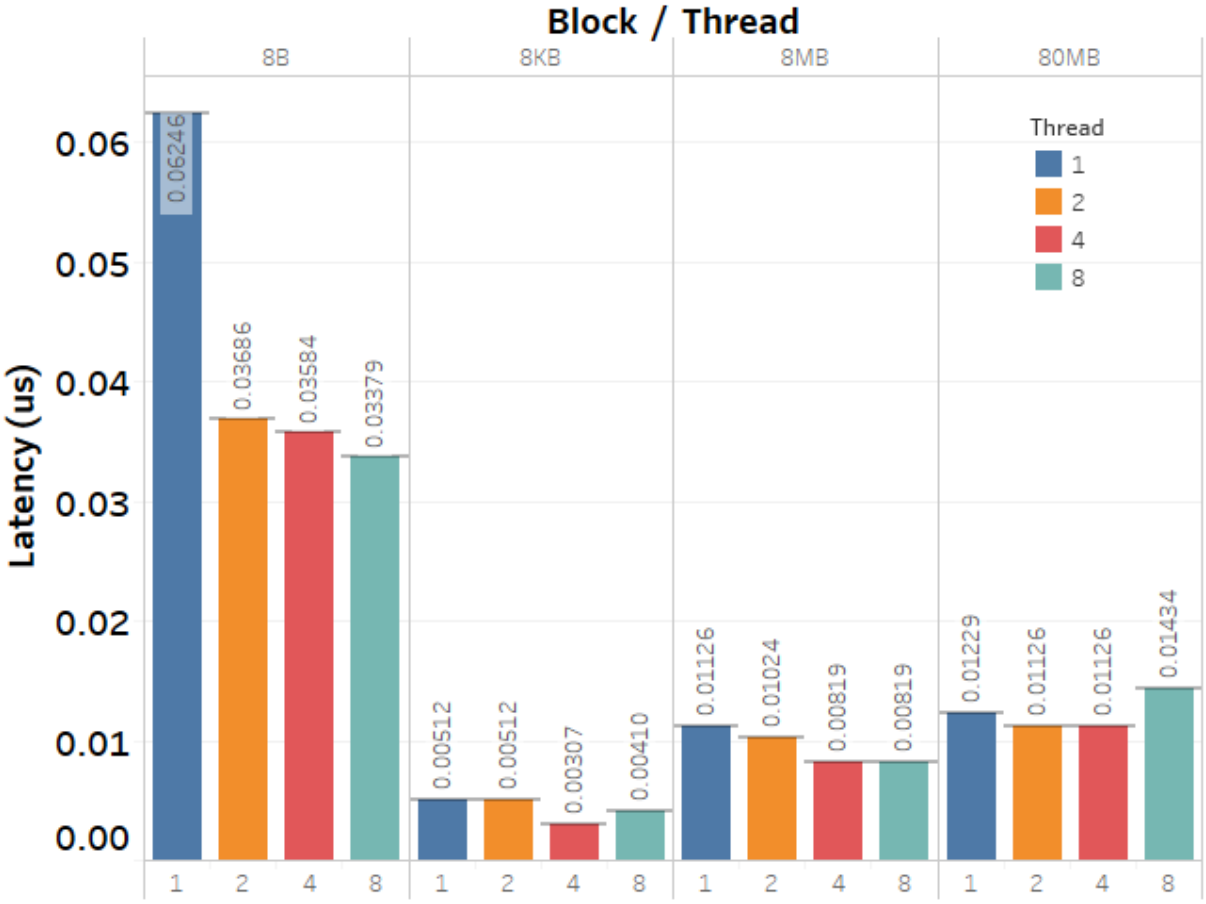
Read Sequential

Thread	Average	Standard Deviation
1	11,512	8,769
2	13,187	9,302
4	15,876	13,068
8	15,216	13,322

Above plot represents the throughput for sequential reading on memory for different block size and number of threads threads and average and standard deviation of all threads. Throughput is measure in the MegaBytes per seconds.

We observed that we get low throughput for block size 8B as it takes more time to read memory as compare to other block size.

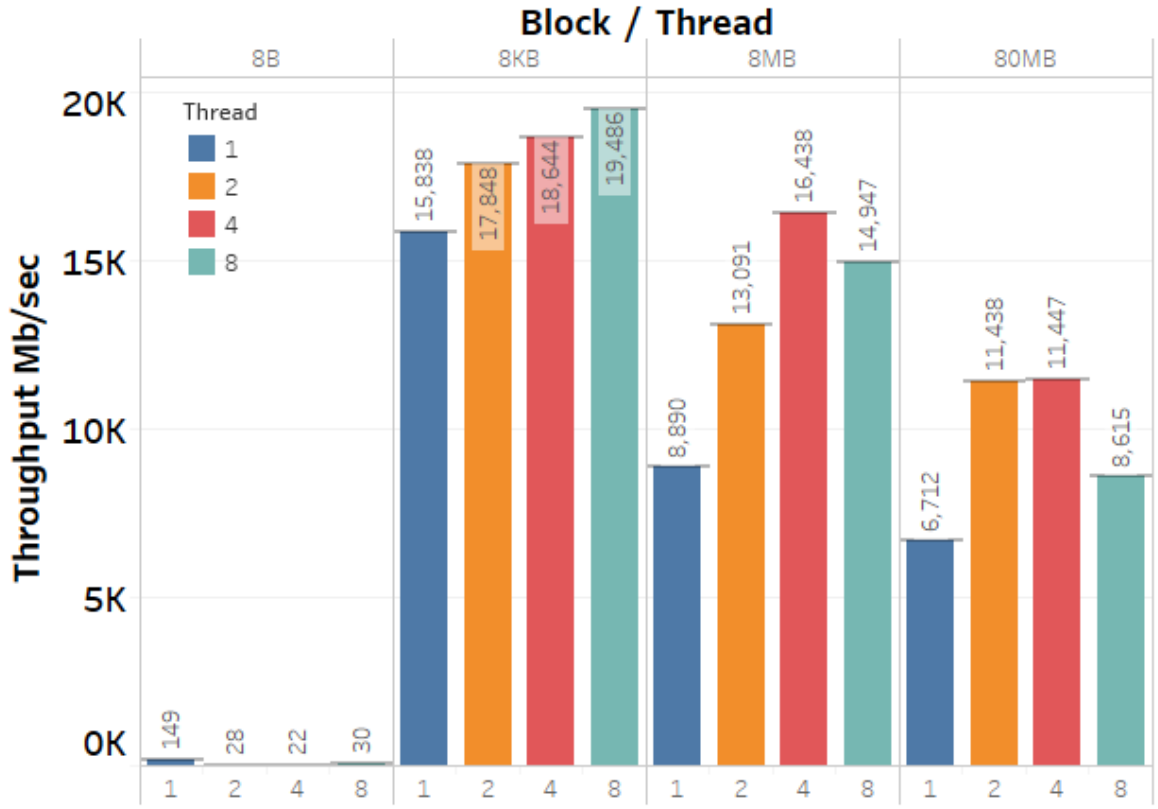
Read Sequential



Above plot represents the latency for sequential reading on memory for different block size and number of threads. latency is measure in microseconds (us).

We observed that we get high latency for block size 8B as it takes more time to read memory compare to other block size.

Read Random



Read Random

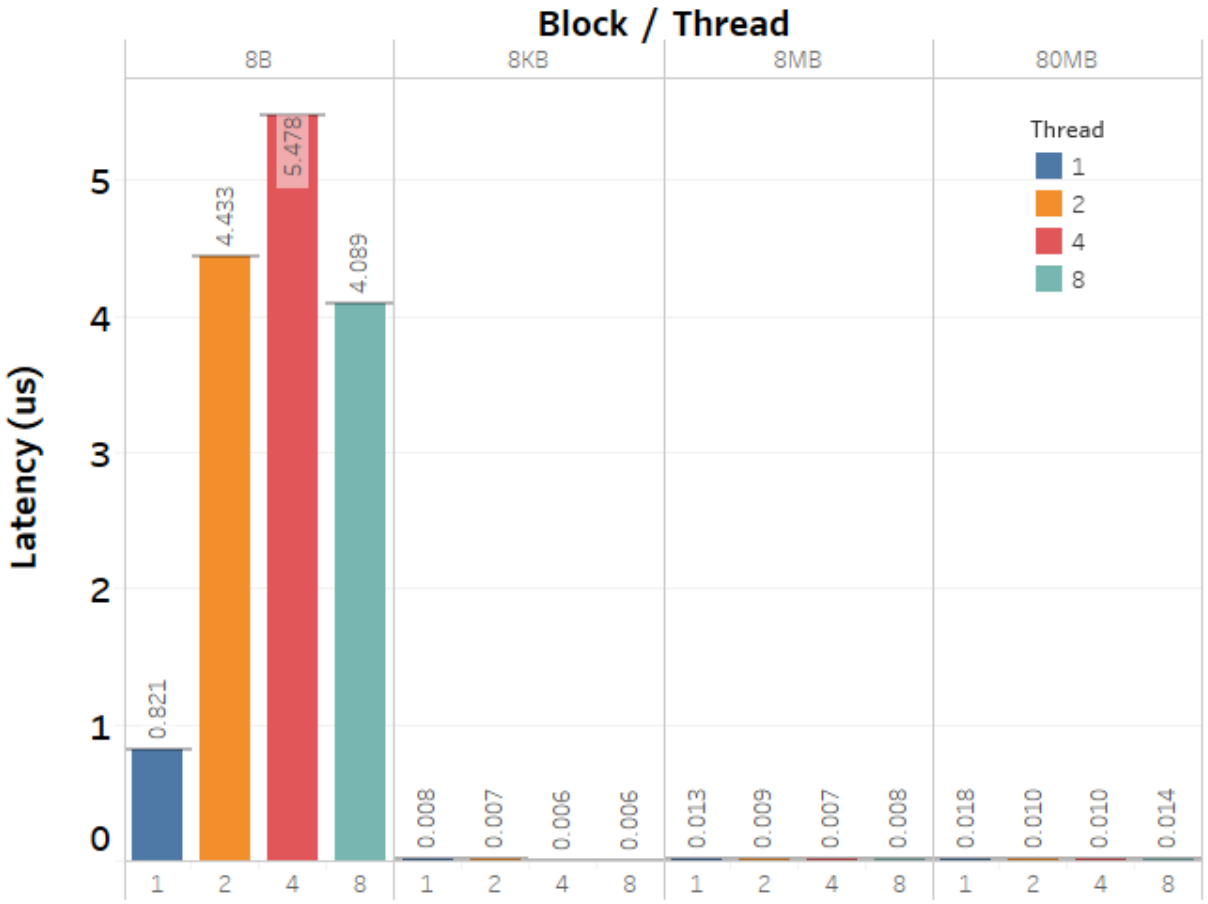
Thread	Average	Standard Deviation
1	7,897	6,467
2	10,601	7,555
4	11,638	8,308
8	10,770	8,434

Above plot represents the throughput for random reading on memory for different block size and number of threads threads and average and standard deviation of all threads. Throughput is measure in the MegaBytes per seconds.

We observed that we get low throughput for block size 8B as it takes more time to read memory as compare to other block size.

We also observed that random throughput of random read with 8B is low as compare to sequential read.

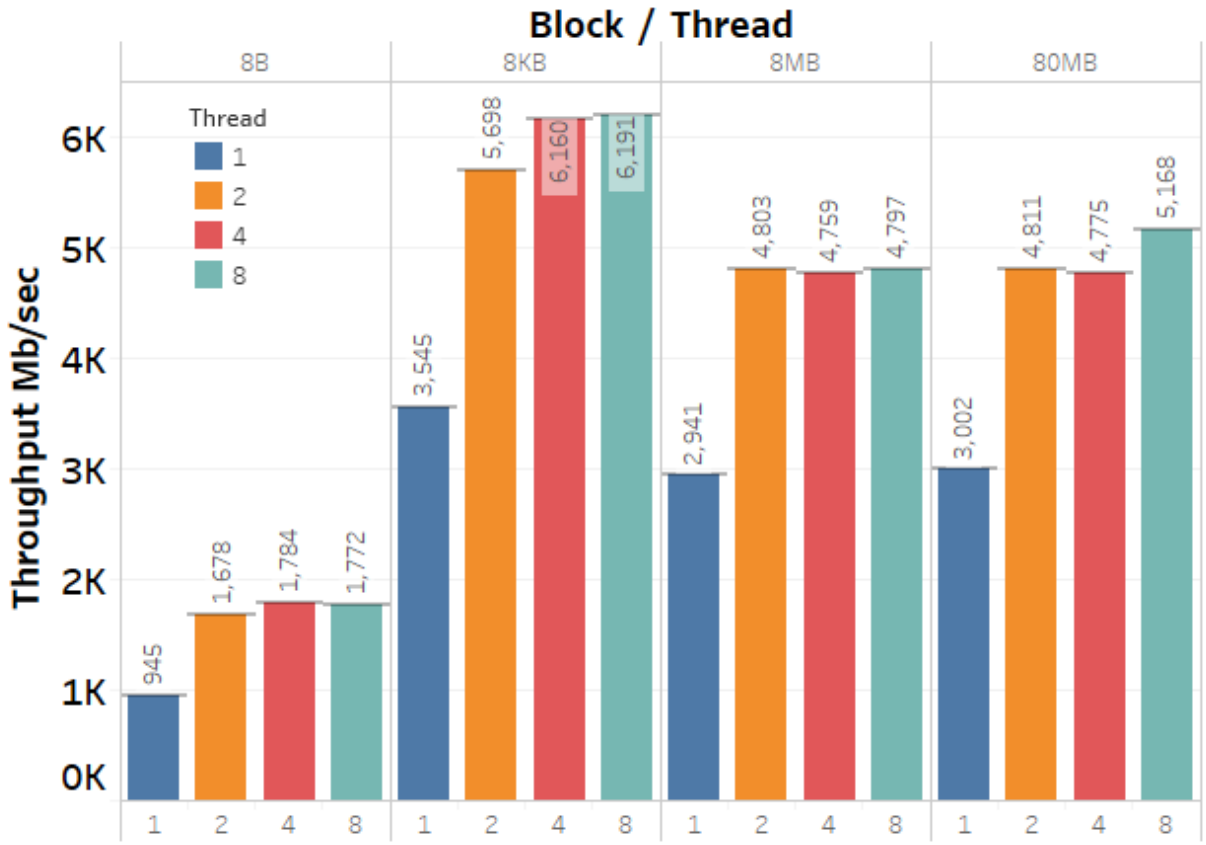
Read Random



Above plot represents the latency for random reading on memory for different block size and number of threads. latency is measure in microseconds (us).

We observed that we get high latency for block size 8B as it takes more time to read memory compare to other block size.

Read + Write



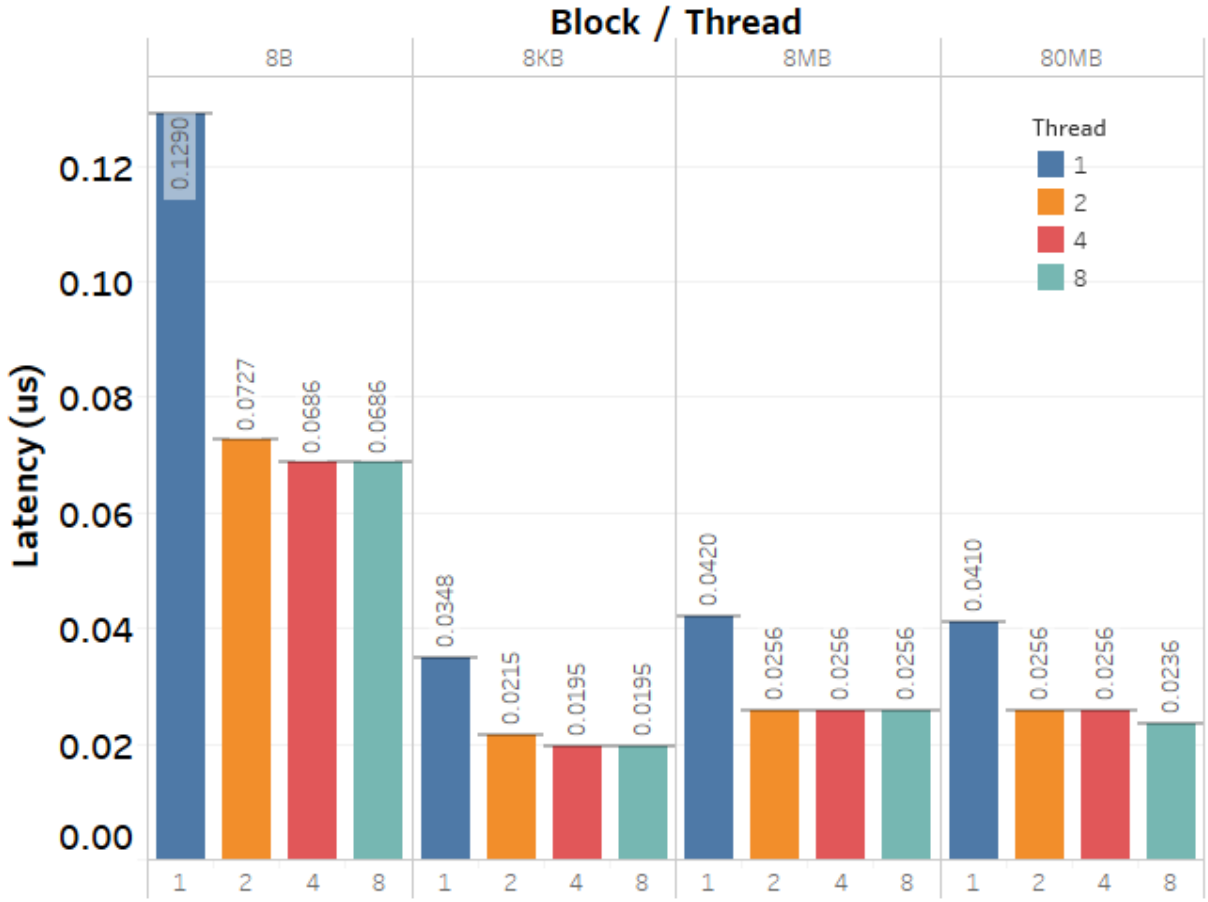
Read + Write

Thread	Average	Standard Deviation
1	2,608	1,142
2	4,248	1,764
4	4,370	1,844
8	4,482	1,900

Above plot represents the throughput for reading and writing on memory for different block size and number of threads threads and average and standard deviation of all threads. Throughput is measure in the MegaBytes per seconds.

We observed that we get low throughput for block size 8B as it takes more time to read and write memory as compare to other block size.

Read + Write



Above plot represents the latency for reading and writing on memory for different block size and number of threads. latency is measure in microseconds (us).

We observed that we get high latency for block size 8B as it takes more time to read and write memory compare to other block size.

We observed that for concurrency 8 we get optimal number of performance as compare to another concurrency.

Stream Benchmark Output:

```
-------------------------------------------------------------
STREAM version $Revision: 5.10 $
-------------------------------------------------------------
This system uses 8 bytes per array element.
-------------------------------------------------------------
Array size = 10000000 (elements), Offset = 0 (elements)
Memory per array = 76.3 MiB (= 0.1 GiB).
Total memory required = 228.9 MiB (= 0.2 GiB).
Each kernel will be executed 10 times.
 The *best* time for each kernel (excluding the first iteration)
 will be used to compute the reported bandwidth.
-------------------------------------------------------------
Your clock granularity/precision appears to be 1 microseconds.
Each test below will take on the order of 28862 microseconds.
   (= 28862 clock ticks)
Increase the size of the arrays if this shows that
you are not getting at least 20 clock ticks per test.
-------------------------------------------------------------
WARNING -- The above is only a rough guideline.
For best results, please be sure you know the
precision of your system timer.
-------------------------------------------------------------
Function    Best Rate MB/s  Avg time     Min time     Max time
Copy:            4949.4     0.034250     0.032327     0.037157
Scale:           4827.4     0.035421     0.033144     0.039901
Add:             7147.6     0.036772     0.033578     0.042070
Triad:           6652.5     0.040410     0.036077     0.044236
-------------------------------------------------------------
Solution Validates: avg error less than 1.000000e-13 on all three arrays
-------------------------------------------------------------
```

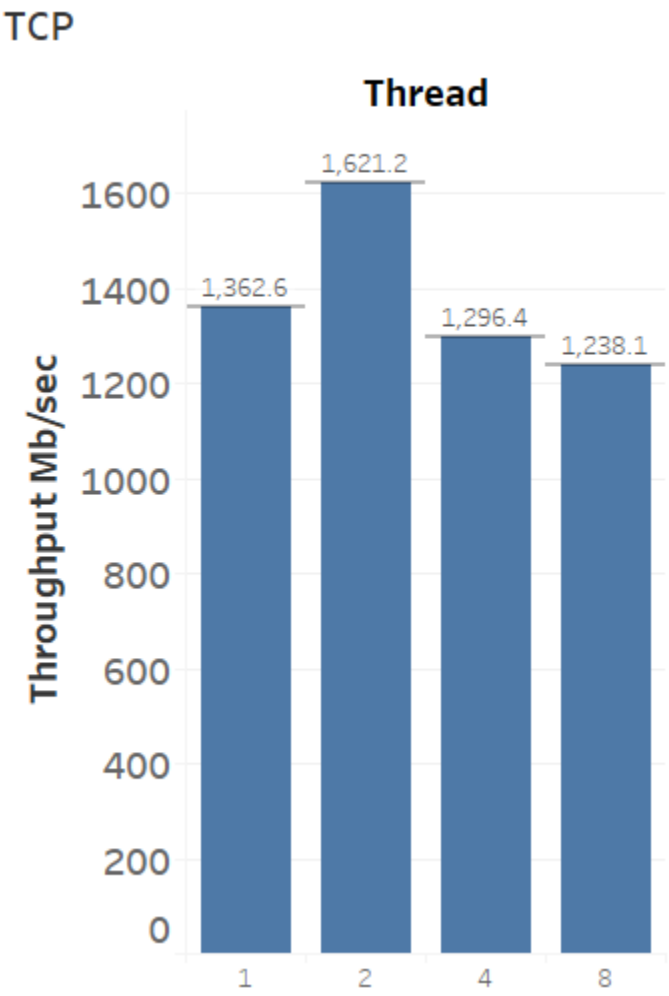
Stream Benchmark for memory is 4949.4 MB/s and we get average 3335 Mb/s for sequential write.

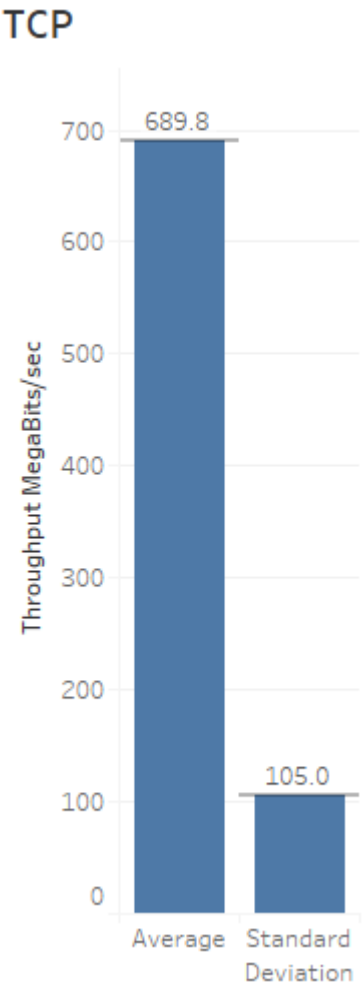
Efficiency is

=3335/4949.4

=67.38%

Network Benchmarking:



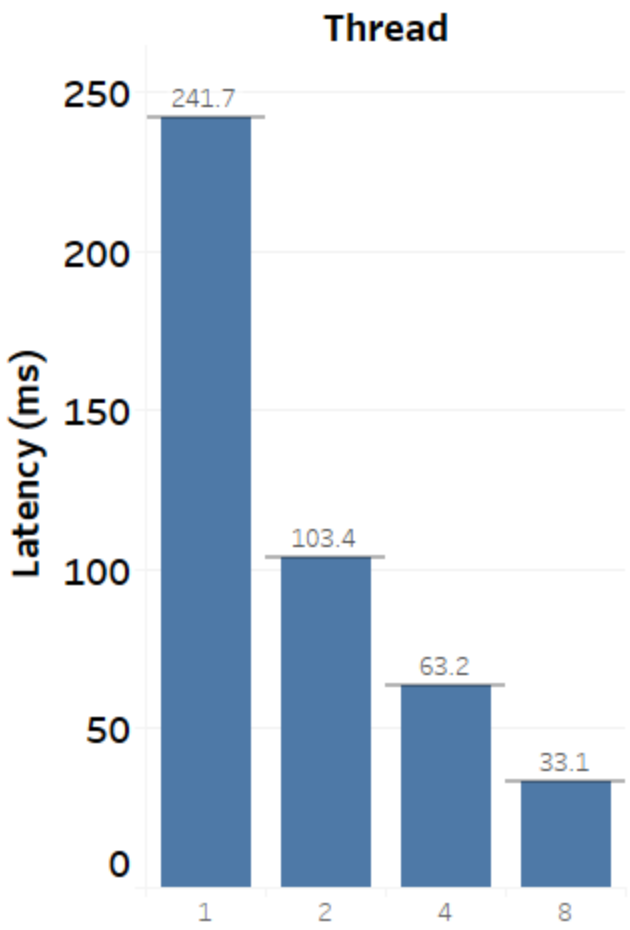


Above represents throughput of TCP communication over loopback interface card and different number of thread and average and standard deviation of all threads. For this benchmark we fix packet size to transfer(64kb). Throughput is measure in megabits per seconds

We observed that for thread 1 we get low throughput as compare to other and its take more time for transfer the packet from client to server.

Throughput is increases as we increase the concurrency for TCP communication.

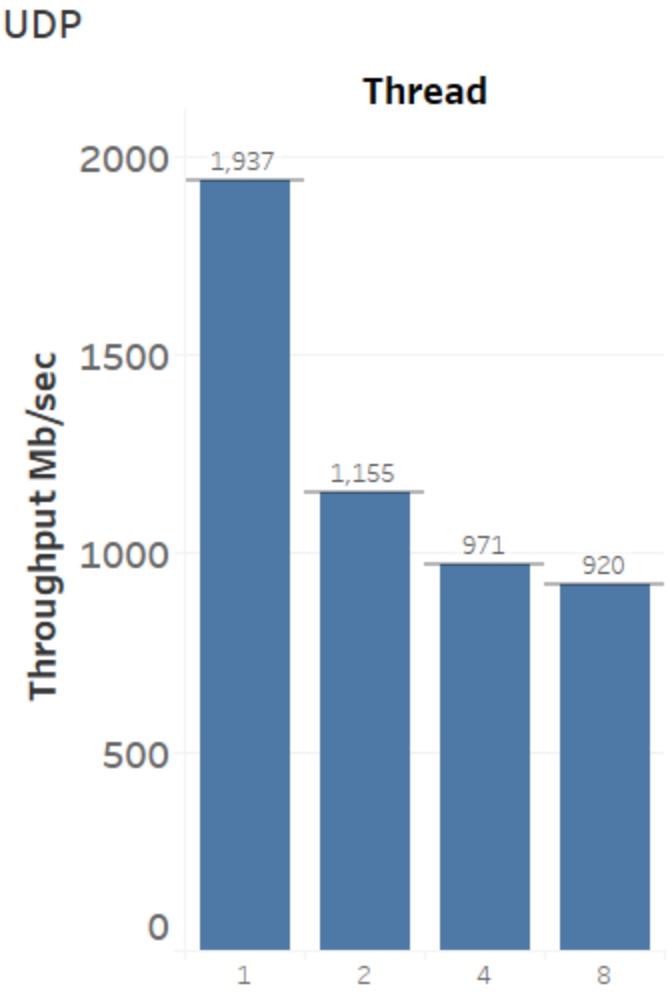
TCP

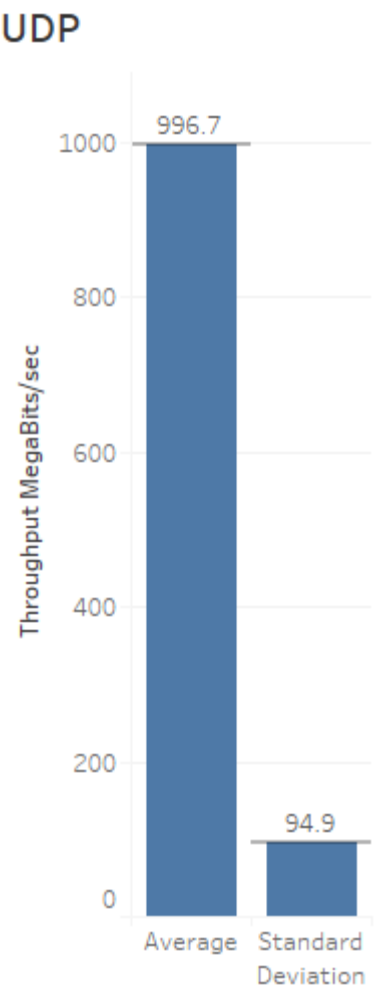


Above represents latency of TCP communication over loopback interface card and different number of thread. For this benchmark we fix packet size to transfer(64kb).

We observed that for thread 1 we get low latency as compare to other and its take more time for transfer the packet from client to server.

We observed that latency decrease as we increase the concurrency for TCP communication.



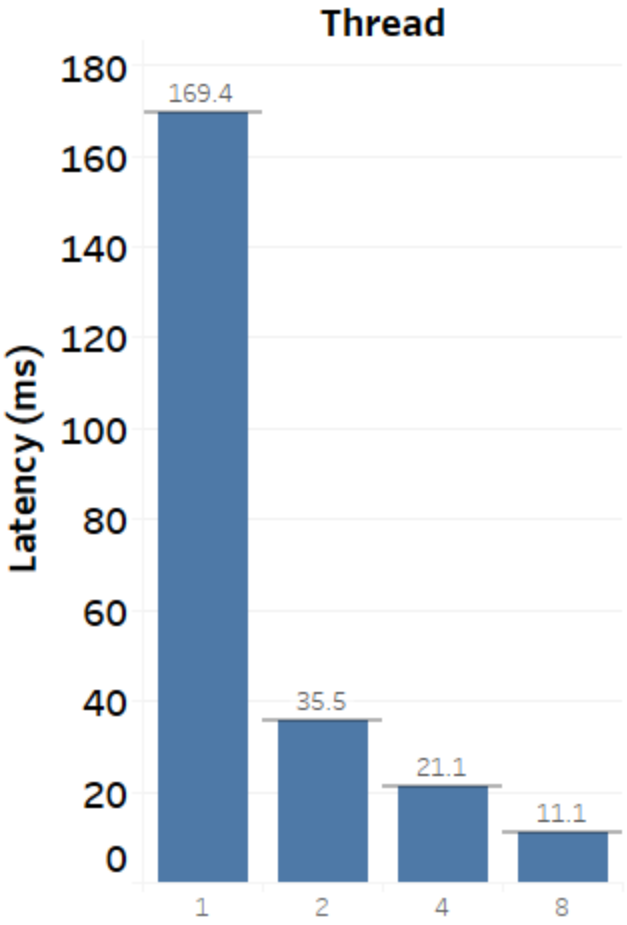


Above represents throughput of UDP communication over loopback interface card and different number of thread and average and standard deviation of all threads. For this benchmark we fix packet size to transfer(64kb). Throughput is measure in megabits per seconds.

We observed that for thread 1 we get low throughput as compare to other and its take more time for transfer the packet from client to server.

We observed that performance for UDP is better than performance of TCP.

UDP



Above represents throughput of UDP communication over loopback interface card and different number of thread. For this benchmark we fix packet size to transfer(64kb).

We observed that for thread 1 we get low throughput as compare to other and its take more time for transfer the packet from client to server.

We observed that latency decrease as we increase the concurrency for TCP communication.

IPRef Output:

TCP

```
Server listening on 5201
-----------------------------------------------------------
Accepted connection from ::1, port 60514
[  5] local ::1 port 5201 connected to ::1 port 60516
[ ID] Interval           Transfer     Bandwidth
[  5]   0.00-1.00   sec  2.36 GBytes  20.3 Gbits/sec
[  5]   1.00-2.00   sec  2.43 GBytes  20.9 Gbits/sec
[  5]   2.00-3.00   sec  2.34 GBytes  20.1 Gbits/sec
[  5]   3.00-4.00   sec  2.22 GBytes  19.0 Gbits/sec
[  5]   4.00-5.00   sec  2.47 GBytes  21.3 Gbits/sec
[  5]   5.00-6.00   sec  2.26 GBytes  19.4 Gbits/sec
[  5]   6.00-7.00   sec  2.27 GBytes  19.5 Gbits/sec
[  5]   7.00-8.00   sec  2.08 GBytes  17.8 Gbits/sec
[  5]   8.00-9.00   sec  2.07 GBytes  17.8 Gbits/sec
[  5]   9.00-10.00  sec  2.89 GBytes  24.8 Gbits/sec
[  5]  10.00-10.04  sec   107 MBytes  25.5 Gbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth
[  5]   0.00-10.04  sec  0.00 Bytes  0.00 bits/sec                  sender
[  5]   0.00-10.04  sec  23.5 GBytes  20.1 Gbits/sec                  receiver
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

```
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
Accepted connection from ::1, port 60514
[  5] local ::1 port 5201 connected to ::1 port 60516
[ ID] Interval           Transfer     Bandwidth
[  5]   0.00-1.00   sec  2.36 GBytes  20.3 Gbits/sec
[  5]   1.00-2.00   sec  2.43 GBytes  20.9 Gbits/sec
[  5]   2.00-3.00   sec  2.34 GBytes  20.1 Gbits/sec
[  5]   3.00-4.00   sec  2.22 GBytes  19.0 Gbits/sec
[  5]   4.00-5.00   sec  2.47 GBytes  21.3 Gbits/sec
[  5]   5.00-6.00   sec  2.26 GBytes  19.4 Gbits/sec
[  5]   6.00-7.00   sec  2.27 GBytes  19.5 Gbits/sec
[  5]   7.00-8.00   sec  2.08 GBytes  17.8 Gbits/sec
[  5]   8.00-9.00   sec  2.07 GBytes  17.8 Gbits/sec
[  5]   9.00-10.00  sec  2.89 GBytes  24.8 Gbits/sec
[  5]  10.00-10.04  sec   107 MBytes  25.5 Gbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth
[  5]   0.00-10.04  sec  0.00 Bytes  0.00 bits/sec                  sender
[  5]   0.00-10.04  sec  23.5 GBytes  20.1 Gbits/sec                  receiver
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

UDP

```
Server listening on 5201
-----------------------------------------------------------
Accepted connection from ::1, port 60538
[  5] local ::1 port 5201 connected to ::1 port 33568
[ ID] Interval           Transfer     Bandwidth       Jitter    Lost/Total Datagrams
[  5]   0.00-1.00   sec   128 KBytes  1.05 Mbits/sec  0.002 ms  0/6 (0%)
[  5]   1.00-2.00   sec   128 KBytes  1.05 Mbits/sec  0.011 ms  0/6 (0%)
[  5]   2.00-3.00   sec   128 KBytes  1.05 Mbits/sec  0.016 ms  0/6 (0%)
[  5]   3.00-4.00   sec   128 KBytes  1.05 Mbits/sec  0.018 ms  0/6 (0%)
[  5]   4.00-5.00   sec   128 KBytes  1.05 Mbits/sec  0.023 ms  0/6 (0%)
[  5]   5.00-6.00   sec   128 KBytes  1.05 Mbits/sec  0.022 ms  0/6 (0%)
[  5]   6.00-7.00   sec   128 KBytes  1.05 Mbits/sec  0.016 ms  0/6 (0%)
[  5]   7.00-8.00   sec   128 KBytes  1.05 Mbits/sec  0.018 ms  0/6 (0%)
[  5]   8.00-9.00   sec   128 KBytes  1.05 Mbits/sec  0.020 ms  0/6 (0%)
[  5]   9.00-10.00  sec   128 KBytes  1.05 Mbits/sec  0.021 ms  0/6 (0%)
[  5]  10.00-10.04  sec  0.00 Bytes  0.00 bits/sec  0.021 ms  0/0 (0%)
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Jitter    Lost/Total Datagrams
[  5]   0.00-10.04  sec  0.00 Bytes  0.00 bits/sec  0.021 ms  0/60 (0%)
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

```
Connecting to host localhost, port 5201
[  4] local ::1 port 60516 connected to ::1 port 5201
[ ID] Interval           Transfer     Bandwidth       Retr  Cwnd
[  4]   0.00-1.00   sec  2.46 GBytes  21128 Mbits/sec    0    320 KBytes
[  4]   1.00-2.00   sec  2.43 GBytes  20850 Mbits/sec    0    320 KBytes
[  4]   2.00-3.00   sec  2.34 GBytes  20076 Mbits/sec    0    320 KBytes
[  4]   3.00-4.00   sec  2.23 GBytes  19124 Mbits/sec    0    320 KBytes
[  4]   4.00-5.00   sec  2.47 GBytes  21253 Mbits/sec    0    320 KBytes
[  4]   5.00-6.00   sec  2.24 GBytes  19257 Mbits/sec    0    320 KBytes
[  4]   6.00-7.00   sec  2.29 GBytes  19656 Mbits/sec    0    320 KBytes
[  4]   7.00-8.00   sec  2.08 GBytes  17840 Mbits/sec    0    320 KBytes
[  4]   8.00-9.00   sec  2.08 GBytes  17855 Mbits/sec    0    320 KBytes
[  4]   9.00-10.00  sec  2.89 GBytes  24829 Mbits/sec    0    320 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Retr
[  4]   0.00-10.00  sec  23.5 GBytes  20187 Mbits/sec    0             sender
[  4]   0.00-10.00  sec  23.5 GBytes  20187 Mbits/sec                  receiver

iperf Done.
```

Bandwidth for IPRef bechnmark for TCP is 20.1 Gbits/sec. We got 1621 Mbits/s.

Efficency for TCP is

= 1621/(20.1*1000)

=8.06%

Bandwidth for IPRef benchmark for UDP is 23.5 Gbits/sec. We got 1927 Mbits/s.

Efficiency for UDP is

=1927/(23.5*1000)

=8.2%