CS – 553 Cloud Computing

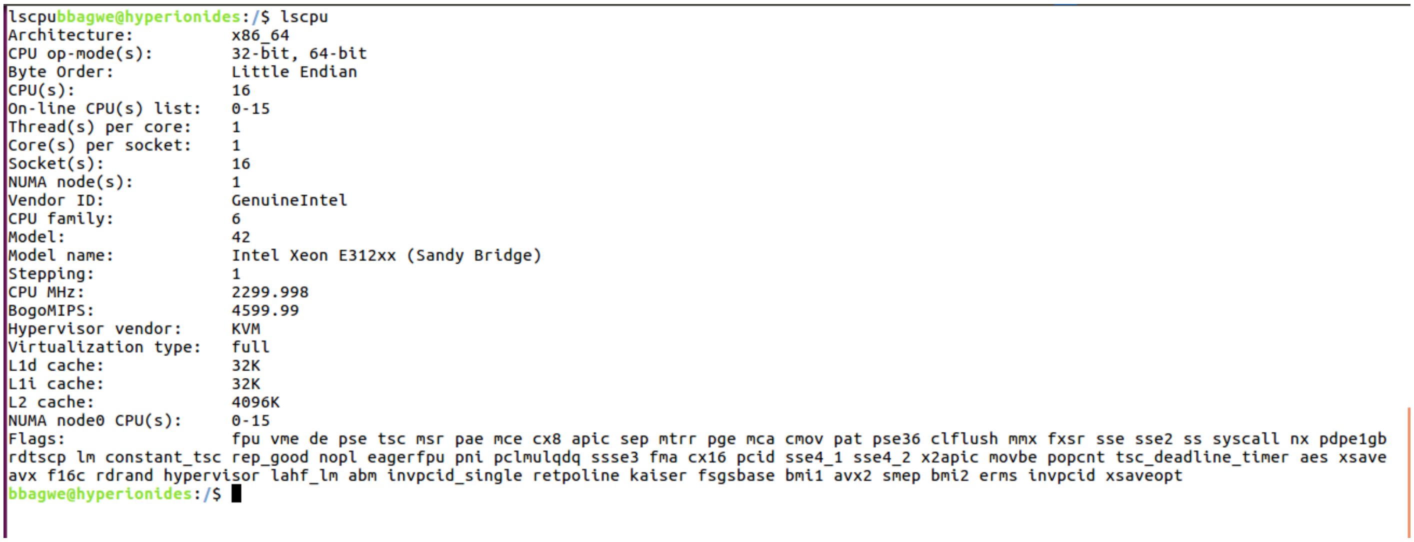
Programming Assignment -1

Benchmarking

Bhagyashree Bagwe
(A20399761)

Note: All the benchmarking programs including standard benchmark are executed on hyperion.
All the source codes can be found at the same location as this document.

A Hyperion instance has following configuration:



```
lscpubbagwe@hyperionides:/$ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                16
On-line CPU(s) list:   0-15
Thread(s) per core:    1
Core(s) per socket:    1
Socket(s):             16
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):     0-15
Flags:                 fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ss syscall nx pdpe1gb
rdtscp lm constant_tsc rep_good nopl eagerfpu pni pclmulqdq ssse3 fma cx16 pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave
avx f16c rdrand hypervisor lahf_lm abm invpcid_single retpoline kaiser fsgsbase bmi1 avx2 smep bmi2 erms invpcid xsaveopt
bbagwe@hyperionides:/$ 
```

1. Processor

a. Source code file : MyCPUBench.c

f. Theoretical peak performance:

$$\text{FLOPS} = \text{sockets} \times \frac{\text{cores}}{\text{socket}} \times \frac{\text{cycles}}{\text{second}} \times \frac{\text{FLOPs}}{\text{cycle}}$$

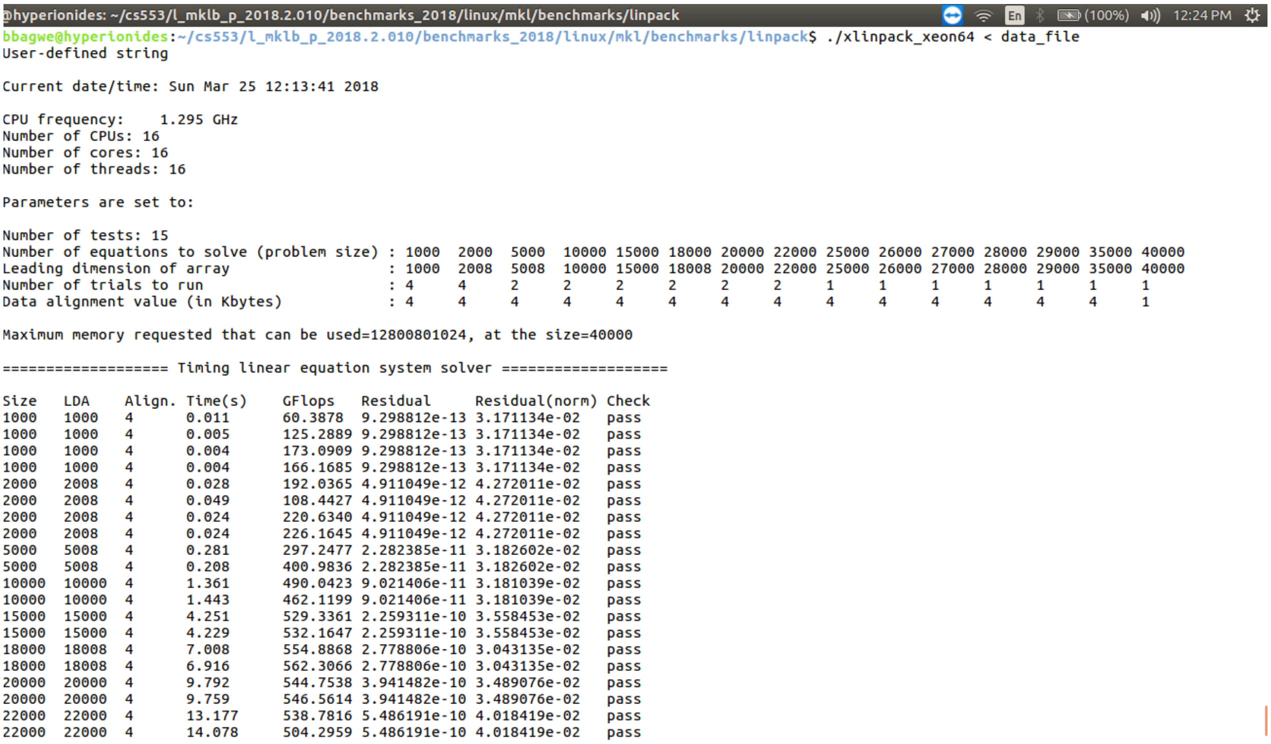
[Ref: https://en.wikipedia.org/wiki/FLOPS]

Using configuration of cluster as shown in above screen-shot, the theoretical performance can be calculated as:

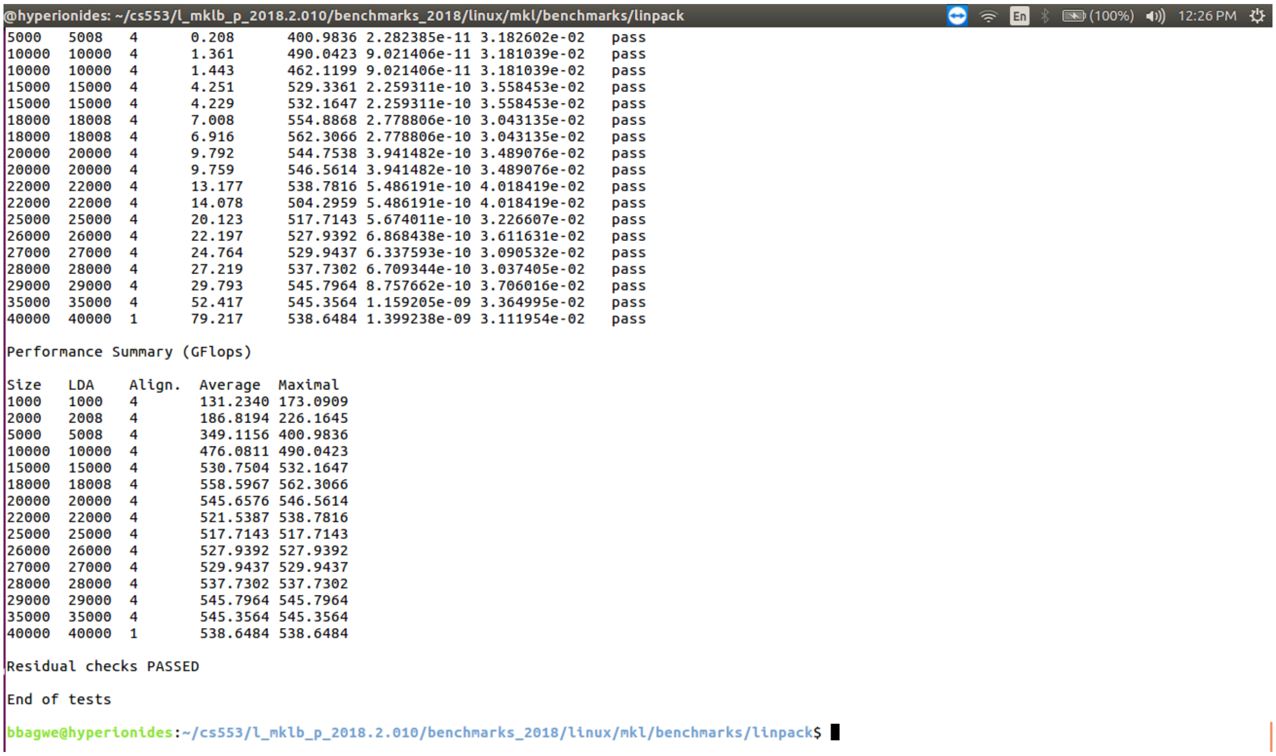
GFLOPS = 16 * 1 * 2.3 * 16 = 588.80

e. LINPACK

LINPACK benchmark was run on the same instance using double precision floating point and it gives below output:


@hyperionides: ~/cs553/l_mklb_p_2018.2.010/benchmarks_2018/linux/mkl/benchmarks/linpack
En
(100%)
12:24 PM

```
bbagwe@hyperionides:~/cs553/l_mklb_p_2018.2.010/benchmarks_2018/linux/mkl/benchmarks/linpack$ ./xlinpack_xeon64 < data_file
User-defined string

Current date/time: Sun Mar 25 12:13:41 2018

CPU frequency:    1.295 GHz
Number of CPUs: 16
Number of cores: 16
Number of threads: 16

Parameters are set to:

Number of tests: 15
Number of equations to solve (problem size) : 1000  2000  5000  10000 15000 18000 20000 22000 25000 26000 27000 28000 29000 35000 40000
Leading dimension of array                  : 1000  2008  5008  10000 15000 18008 20000 22000 25000 26000 27000 28000 29000 35000 40000
Number of trials to run                     : 4     4     2     2     2     2     2     2     1     1     1     1     1     1     1
Data alignment value (in Kbytes)            : 4     4     4     4     4     4     4     4     4     4     4     4     4     4     1

Maximum memory requested that can be used=12800801024, at the size=40000

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

Size   LDA    Align. Time(s)    GFlops   Residual     Residual(norm) Check
1000   1000   4      0.011      60.3878  9.298812e-13 3.171134e-02   pass
1000   1000   4      0.005      125.2889 9.298812e-13 3.171134e-02   pass
1000   1000   4      0.004      173.0909 9.298812e-13 3.171134e-02   pass
1000   1000   4      0.004      166.1685 9.298812e-13 3.171134e-02   pass
2000   2008   4      0.028      192.0365 4.911049e-12 4.272011e-02   pass
2000   2008   4      0.049      108.4427 4.911049e-12 4.272011e-02   pass
2000   2008   4      0.024      220.6340 4.911049e-12 4.272011e-02   pass
2000   2008   4      0.024      226.1645 4.911049e-12 4.272011e-02   pass
5000   5008   4      0.281      297.2477 2.282385e-11 3.182602e-02   pass
5000   5008   4      0.208      400.9836 2.282385e-11 3.182602e-02   pass
10000  10000  4      1.361      490.0423 9.021406e-11 3.181039e-02   pass
10000  10000  4      1.443      462.1199 9.021406e-11 3.181039e-02   pass
15000  15000  4      4.251      529.3361 2.259311e-10 3.558453e-02   pass
15000  15000  4      4.229      532.1647 2.259311e-10 3.558453e-02   pass
18000  18008  4      7.008      554.8868 2.778806e-10 3.043135e-02   pass
18000  18008  4      6.916      562.3066 2.778806e-10 3.043135e-02   pass
20000  20000  4      9.792      544.7538 3.941482e-10 3.489076e-02   pass
20000  20000  4      9.759      546.5614 3.941482e-10 3.489076e-02   pass
22000  22000  4      13.177     538.7816 5.486191e-10 4.018419e-02   pass
22000  22000  4      14.078     504.2959 5.486191e-10 4.018419e-02   pass
```


@hyperionides: ~/cs553/l_mklb_p_2018.2.010/benchmarks_2018/linux/mkl/benchmarks/linpack
En
(100%)
12:26 PM

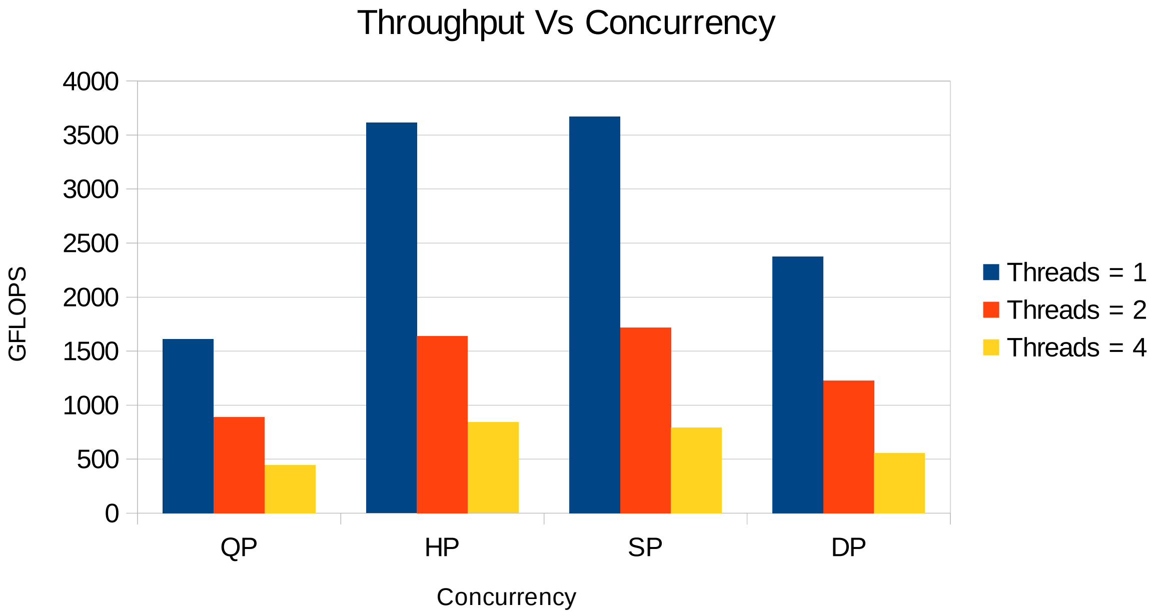
```
5000   5008   4      0.208      400.9836 2.282385e-11 3.182602e-02   pass
10000  10000  4      1.361      490.0423 9.021406e-11 3.181039e-02   pass
10000  10000  4      1.443      462.1199 9.021406e-11 3.181039e-02   pass
15000  15000  4      4.251      529.3361 2.259311e-10 3.558453e-02   pass
15000  15000  4      4.229      532.1647 2.259311e-10 3.558453e-02   pass
18000  18008  4      7.008      554.8868 2.778806e-10 3.043135e-02   pass
18000  18008  4      6.916      562.3066 2.778806e-10 3.043135e-02   pass
20000  20000  4      9.792      544.7538 3.941482e-10 3.489076e-02   pass
20000  20000  4      9.759      546.5614 3.941482e-10 3.489076e-02   pass
22000  22000  4      13.177     538.7816 5.486191e-10 4.018419e-02   pass
22000  22000  4      14.078     504.2959 5.486191e-10 4.018419e-02   pass
25000  25000  4      20.123     517.7143 5.674011e-10 3.226607e-02   pass
26000  26000  4      22.197     527.9392 6.868438e-10 3.611631e-02   pass
27000  27000  4      24.764     529.9437 6.337593e-10 3.090532e-02   pass
28000  28000  4      27.219     537.7302 6.709344e-10 3.037405e-02   pass
29000  29000  4      29.793     545.7964 8.757662e-10 3.706016e-02   pass
35000  35000  4      52.417     545.3564 1.159205e-09 3.364995e-02   pass
40000  40000  1      79.217     538.6484 1.399238e-09 3.111954e-02   pass

Performance Summary (GFlops)

Size   LDA    Align.  Average  Maximal
1000   1000   4       131.2340 173.0909
2000   2008   4       186.8194 226.1645
5000   5008   4       349.1156 400.9836
10000  10000  4       476.0811 490.0423
15000  15000  4       530.7504 532.1647
18000  18008  4       558.5967 562.3066
20000  20000  4       545.6576 546.5614
22000  22000  4       521.5387 538.7816
25000  25000  4       517.7143 517.7143
26000  26000  4       527.9392 527.9392
27000  27000  4       529.9437 529.9437
28000  28000  4       537.7302 537.7302
29000  29000  4       545.7964 545.7964
35000  35000  4       545.3564 545.3564
40000  40000  1       538.6484 538.6484

Residual checks PASSED

End of tests

bbagwe@hyperionides:~/cs553/l_mklb_p_2018.2.010/benchmarks_2018/linux/mkl/benchmarks/linpack$ █
```

Maximum throughput given by LINPACK = 558.59 GFLOPS

Efficiency of LINPACK = (558.59 *100)/ 588.80 = 94.86 %

g. Processor performance table

Workload	Concurrency	MyCPUBench Measured Ops/Sec (GigaOps)	HPL Measured Ops/Sec (GigaOps)	Theoretical Ops/Sec (GigaOps)	MyCPUBench Efficiency (%)	HPL Efficiency (%)
QP	1	1610.58	N/A	588.80	273.54	N/A
QP	2	889.18	N/A	588.80	151.02	N/A
QP	4	445.09	N/A	588.80	75.59	N/A
HP	1	3612.99	N/A	588.80	613.62	N/A
HP	2	1637.70	N/A	588.80	278.14	N/A
HP	4	840.27	N/A	588.80	142.71	N/A
SP	1	3672.05	N/A	588.80	623.65	N/A
SP	2	1716.91	N/A	588.80	291.60	N/A
SP	4	793.90	N/A	588.80	134.83	N/A
DP	1	2375.17	558.59	588.80	403.39	94.86
DP	2	1226.00	558.59	588.80	208.22	94.86
DP	4	556.96	558.59	588.80	94.59	94.86



As no of threads increases the performance decreases due to strong scaling.

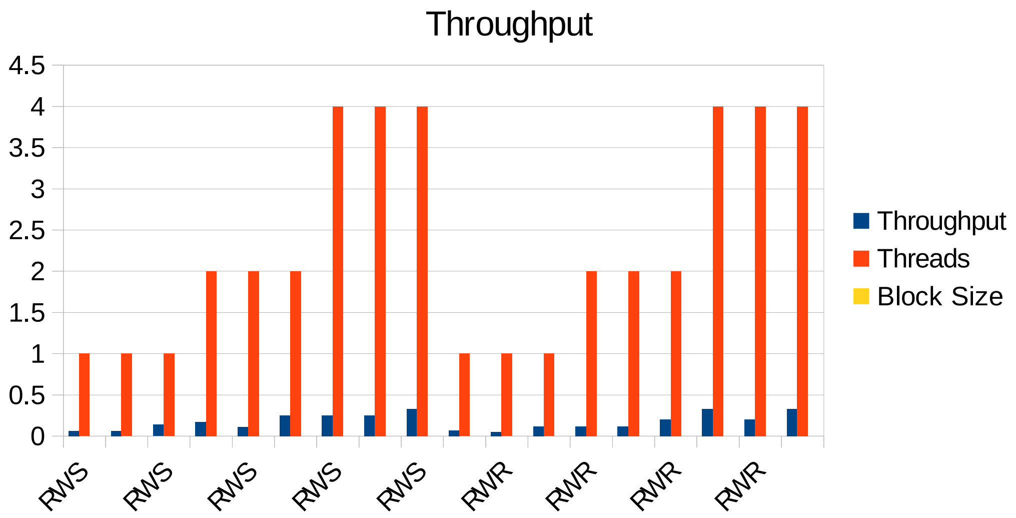
2. Memory

a. Source Code file : MyRAMBench.c

f. Theoretical throughput = 12.8 GB/s [Ref: https://en.wikipedia.org/wiki/DDR4_SDRAM]

g. Memory throughput table:

Workload	Concurrency	Block Size	MyRAMBench Measured Throughput (GB/sec)	Pmbw Measured Throughput (GB/sec)	Theoretical Throughput (GB/s)	MyRAMBench Efficiency (%)	Pmbw Efficiency (%)
RWS	1	1KB	0.06	3.80	12.8	0.47	29.68
RWS	1	1MB	0.06	3.80	12.8	0.47	29.68
RWS	1	10MB	0.14	3.80	12.8	1.09	29.68
RWS	2	1KB	0.17	1.91	12.8	1.33	14.92
RWS	2	1MB	0.11	1.91	12.8	0.86	14.92
RWS	2	10MB	0.25	1.91	12.8	1.95	14.92
RWS	4	1KB	0.25	9.52	12.8	1.95	74.38
RWS	4	1MB	0.25	9.52	12.8	1.95	74.38
RWS	4	10MB	0.33	9.52	12.8	2.58	74.38
RWR	1	1KB	0.07	3.68	12.8	0.55	28.75
RWR	1	1MB	0.05	3.68	12.8	0.39	28.75
RWR	1	10MB	0.12	3.68	12.8	0.94	28.75
RWR	2	1KB	0.12	1.88	12.8	0.94	14.68
RWR	2	1MB	0.12	1.88	12.8	0.94	14.68
RWR	2	10MB	0.20	1.88	12.8	1.56	14.68
RWR	4	1KB	0.33	9.96	12.8	2.58	77.82
RWR	4	1MB	0.20	9.96	12.8	1.56	77.81
RWR	4	10MB	0.33	9.96	12.8	2.58	77.81



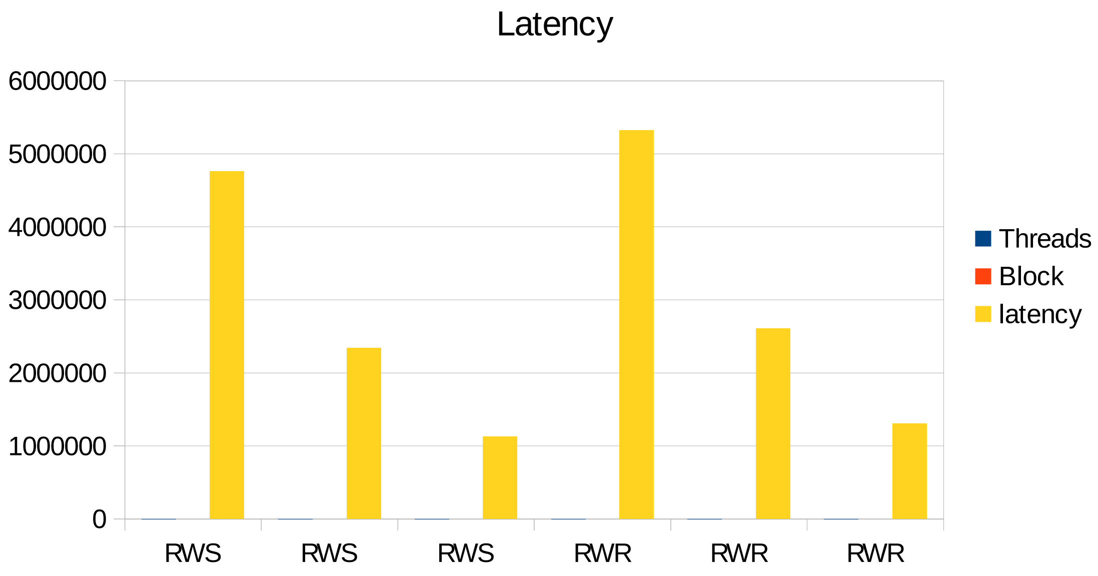
Memory Latency Table:

Workload	Concurrency	Block Size	MyRAMBench Measured Latency (us)	Pmbw Measured Latency (us)	Theoretical Latency (us)	MyRAMBench Efficiency (%)	Pmbw Efficiency (%)
RWS	1	1B	4760000	150000	13390	0.28	8.92
RWS	2	1B	2340000	132000	13390	0.57	10
RWS	4	1B	1130000	150000	13390	1.18	8.92
RWR	1	1B	5320000	160000	13390	2.51	8.36
RWR	2	1B	2610000	148000	13390	0.51	9.04
RWR	4	1B	1310000	11200	13390	1.02	11.9

Efficiency increases with increased no of threads

Latency decreases with increased no of threads.

Thus it is safe to say that for memory benchmark, increasing concurrency of the program improves the performance of the program.



e. Link to PMBW output in stats.txt:

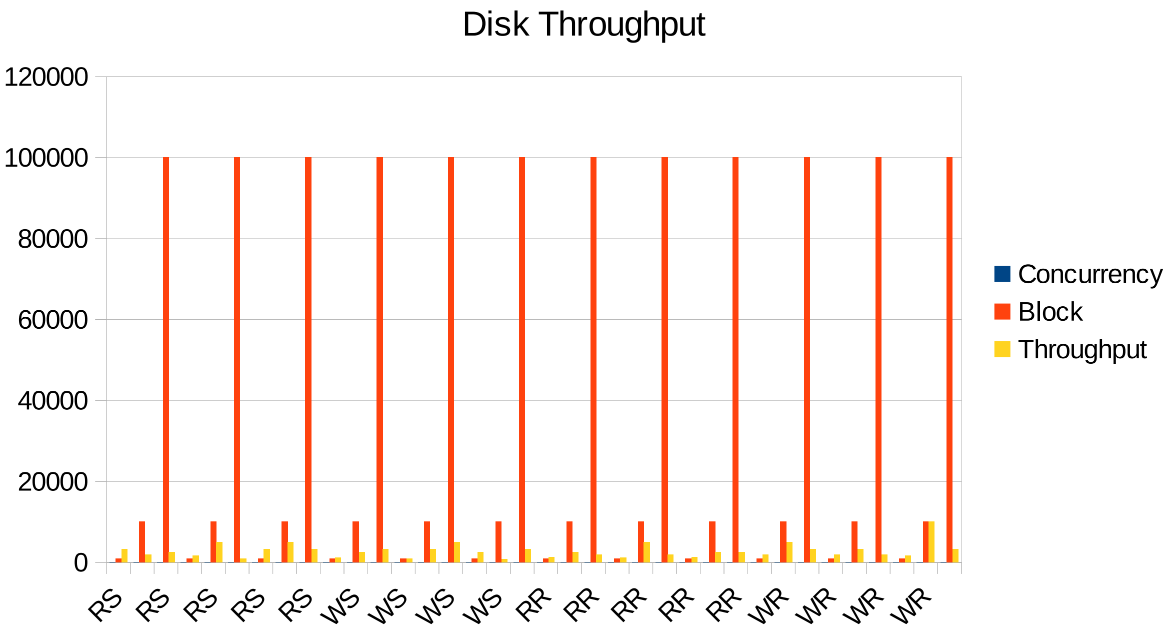
```
RESULT	datetime=2018-03-27 22:13:05 host=hyperionides version=0.6.2
	funcname=ScanWrite64PtrSimpleLoop nthreads=1 areasize=1024
	threadsize=1152 testsize=1152 repeats=[illegible] testvol=[illegible]
	testaccess=[illegible] time=[illegible]
	bandwidth=[illegible] rate=[illegible]e-10
RESULT	datetime=2018-03-27 22:13:08 host=hyperionides version=0.6.2
	funcname=ScanWrite64PtrSimpleLoop nthreads=1 areasize=2048
	threadsize=2176 testsize=2176 repeats=[illegible] testvol=[illegible]
	testaccess=[illegible] time=[illegible]
	bandwidth=[illegible] rate=[illegible]e-10
```

3. Disk

Source code file : MyDiskBench.c
Theoretical throughput : 1030 MB/sec
Theoretical Latency: 4.16
[Ref: https://www.newegg.com/Product/Product.aspx?Item=N82E16822148703,
https://en.wikipedia.org/wiki/Hard_disk_drive]

Workload	Concurrency	Block Size (Bytes)	MyDiskBench Measured Throughput (MB/sec)	Theoretical Throughput (MB/sec)	MyDiskBench Efficiency
RS	1	1000	3333.33	1030	323.62
RS	1	10000	2000.00	1030	194.17
RS	1	100000	2500.00	1030	242.72
RS	2	1000	1666.67	1030	161.90
RS	2	10000	5000.00	1030	485.43
RS	2	100000	1000.00	1030	97.08
RS	4	1000	3333.33	1030	323.62

RS	4	10000	5000.00	1030	485.43
RS	4	100000	3333.33	1030	323.62
WS	1	1000	1111.11	1030	107.87
WS	1	10000	2500.00	1030	242.72
WS	1	100000	3333.33	1030	323.62
WS	2	1000	1000.00	1030	97.08
WS	2	10000	3333.33	1030	323.62
WS	2	100000	5000.00	1030	485.43
WS	4	1000	2500.00	1030	242.72
WS	4	10000	833.33	1030	80.95
WS	4	100000	3333.33	1030	323.62
RR	1	1000	1250.00	1030	121.35
RR	1	10000	2500.00	1030	242.72
RR	1	100000	2000.00	1030	194.17
RR	2	1000	1111.11	1030	107.87
RR	2	10000	5000.00	1030	485.43
RR	2	100000	2000.00	1030	194.17
RR	4	1000	1250.00	1030	121.35
RR	4	10000	2500.00	1030	242.72
RR	4	100000	2500.00	1030	242.72
WR	1	1000	2000.00	1030	194.17
WR	1	10000	5000.00	1030	484.43
WR	1	100000	3333.33	1030	323.62
WR	2	1000	2000.00	1030	191.17
WR	2	10000	3333.33	1030	323.62
WR	2	100000	2000.00	1030	191.17
WR	4	1000	1666.67	1030	161.90
WR	4	10000	10000.00	1030	970.87
WR	4	100000	3333.33	1030	323.62

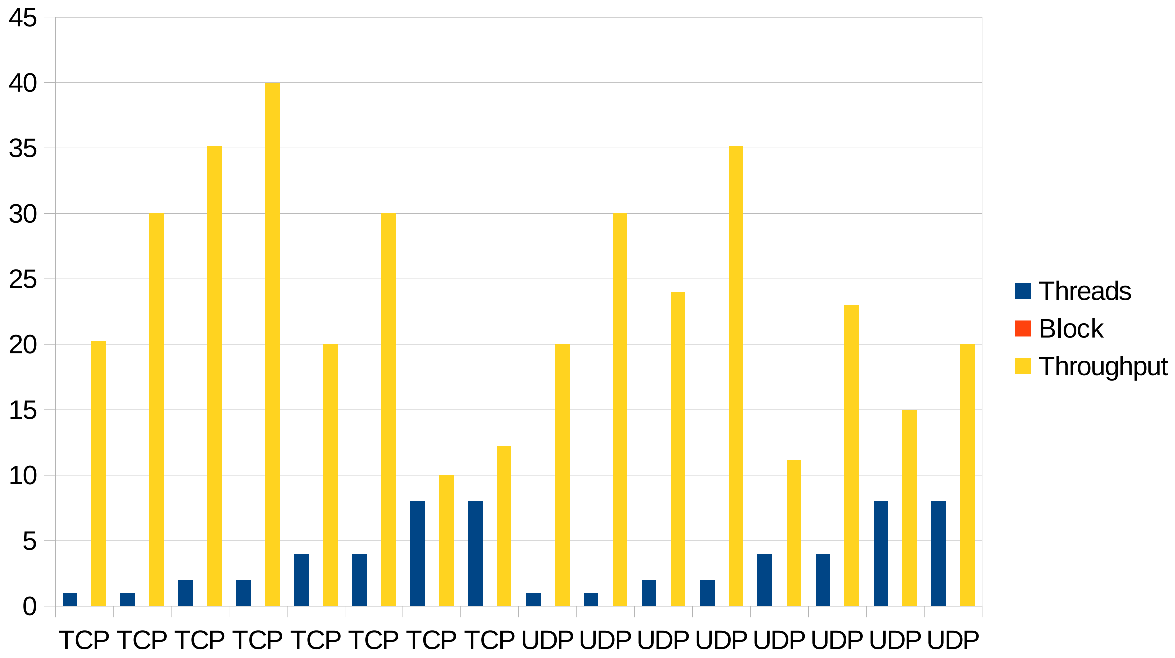


Disk performance improves with increased concurrency.

4. Network

Protocol	Concurrency	Block Size	MyNetBench Measured Throughput (Mb/sec)	Theoretical Throughput	MyNetBench Efficiency (%)
TCP	1	1KB	20.23	130	15.56
TCP	1	32KB	30.00	130	23.07
TCP	2	1KB	35.12	130	27.01
TCP	2	32KB	40.00	130	30.76
TCP	4	1KB	20.00	130	15.38
TCP	4	32KB	30.00	130	23.07
TCP	8	1KB	10.00	130	7.69
TCP	8	32KB	12.23	130	9.40
UDP	1	1KB	20.00	130	15.38
UDP	1	32KB	30.00	130	23.07
UDP	2	1KB	24.00	130	18.46
UDP	2	32KB	35.12	130	27.01
UDP	4	1KB	11.12	130	8.55

UDP	4	32KB	23.00	130	17.69
UDP	8	1KB	15.00	130	11.53
UDP	8	32KB	20.00	130	15.38


TCP TCP TCP TCP TCP TCP TCP TCP UDP UDP UDP UDP UDP UDP UDP UDP
Threads
Block
Throughput