Pooja Patel
A20396099

PERFORMANCE REPORT

PROGRAMMING ASSIGNMENT 01
CS 553 CLOUD COMPUTING

- The performance report includes all the tables in which we are required to be filled by Mybenchmark values and also the standard benchmark values and also the efficiency is calculated and entered in the table
- It also consists of charts and graphs which I have analyzed them
- All the experiments are performed on Hyperion except the disk benchmark
- The following is the screenshot of the specifications of Hyperion:

```
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 42
model name      : Intel Xeon E312xx (Sandy Bridge)
stepping        : 1
microcode       : 0x1
cpu MHz         : 2299.998
cache size      : 4096 KB
physical id     : 0
siblings        : 1
core id         : 0
cpu cores       : 1
apicid          : 0
initial apicid  : 0
fpu             : yes
fpu_exception   : yes
cpuid level     : 13
wp              : yes
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_l
m abm invpcid_single retpoline kaiser fsgsbase bmi1 avx2 smep bmi2 erms invpcid xsaveopt
bugs            : cpu_meltdown spectre_v1 spectre_v2
bogomips        : 4599.99
clflush size    : 64
cache_alignment : 64
address sizes   : 46 bits physical, 48 bits virtual
power management:

processor       : 1
vendor_id       : GenuineIntel
cpu family      : 6
model           : 42
model name      : Intel Xeon E312xx (Sandy Bridge)
stepping        : 1
microcode       : 0x1
cpu MHz         : 2299.998
cache size      : 4096 KB
physical id     : 1
siblings        : 1
core id         : 0
cpu cores       : 1
apicid          : 1
initial apicid  : 1
```

Pooja Patel
A20396099

Benchmark Values for the following:

1. CPU

- **The following table shows all the values of MyCPUBench, HPL Measured and the theoretical values also the MyCPUBench and HPL efficiency**
- **For CPU benchmark experiments I have used Hyperion cluster**
- **The unit in which MyCPUBench, HPL and theoretical values are measured are in ops/sec(GigaOPS)**
- **The experiments were carried out multiple times**
- **Theoretical Ops/sec is calculated using the following formula :**
  Flops = no. of sockets * cores per socket * cycles per second * flops per cycle

  Flops for DP = 2* 1 * 2.299 * 16
  Flops for SP = 2* 1 * 2.299 * 32
  Flops for HP = 2* 1 * 2.299 * 64
  Flops for QP = 2* 1 * 2.299 * 128
- **MyCPUBench Efficiency is calculated as MyCPUBench Value / Theoretical Value**
- **HPL Efficiency is calculated as HPL Measured value / Theoretical Value**

CPU						
Workload	Concurrency	MyCPUBench Measured Ops/Sec (GigaOPS)	HPL Measured Ops/Sec (GigaOPS)	Theoretical Ops/Sec (GigaOPS)	MyCPUBench Efficiency (%)	HPL Efficiency (%)
QP	1	4.769879	N/A	588.544	0.810454104	N/A
QP	2	7.994183	N/A	588.544	1.358298275	N/A
QP	4	9.603246	N/A	588.544	1.631695506	N/A
HP	1	4.074414	N/A	294.272	1.384574135	N/A
HP	2	9.492386	N/A	294.272	3.225718383	N/A
HP	4	8.316971	N/A	294.272	2.826286905	N/A
SP	1	4.395861	N/A	147.136	2.987617578	N/A
SP	2	7.864433	N/A	147.136	5.345009379	N/A
SP	4	8.44972	N/A	147.136	5.742795781	N/A
DP	1	5.044236	33.4721	73.568	6.856562636	45.49818
DP	2	8.406412	57.0881	73.568	11.42672358	77.59909
DP	4	9.76965	68.7276	73.568	13.27975478	93.42051

Pooja Patel
A20396099

Table 1 : CPU BENCHMARK VALUES

- **Following is the screenshot of the HPL benchmark linpack for 1 thread which I executed :**

```
Sample data file lininput_xeon64.

Current date/time: Tue Mar 27 05:10:31 2018

CPU frequency:    3.083 GHz
Number of CPUs: 2
Number of cores: 2
Number of threads: 1

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 30000 35000 40000 45000
Leading dimension of array                  : 1000  2000  5008  10000 15000 18008 20016 22008 25000 26000 27000 30000 35000 40000 45000
Number of trials to run                     : 4     2     2     2     2     2     2     2     2     2     1     1     1     1     1
Data alignment value (in Kbytes)            : 4     4     4     4     4     4     4     4     4     4     4     1     1     1     1

Maximum memory requested that can be used=3202964416, at the size=20000

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

Size   LDA    Align. Time(s)    GFlops   Residual     Residual(norm) Check
1000   1000   4      0.028      23.6071  9.916512e-13 3.381785e-02   pass
1000   1000   4      0.026      26.1259  9.916512e-13 3.381785e-02   pass
1000   1000   4      0.025      26.3669  9.916512e-13 3.381785e-02   pass
1000   1000   4      0.026      26.1125  9.916512e-13 3.381785e-02   pass
2000   2000   4      0.171      31.2037  4.112481e-12 3.577355e-02   pass
2000   2000   4      0.162      32.8809  4.112481e-12 3.577355e-02   pass
5000   5008   4      2.354      35.4291  2.344680e-11 3.269467e-02   pass
5000   5008   4      2.425      34.3787  2.344680e-11 3.269467e-02   pass
10000  10000  4      18.666     35.7263  1.041791e-10 3.673460e-02   pass
10000  10000  4      18.453     36.1393  1.041791e-10 3.673460e-02   pass
15000  15000  4      61.068     36.8513  2.108050e-10 3.320215e-02   pass
15000  15000  4      62.615     35.9409  2.108050e-10 3.320215e-02   pass
18000  18008  4      114.463    33.9729  2.906587e-10 3.183070e-02   pass
18000  18008  4      112.355    34.6103  2.906587e-10 3.183070e-02   pass
20000  20016  4      155.533    34.2958  3.578535e-10 3.167788e-02   pass
20000  20016  4      163.381    32.6484  3.578535e-10 3.167788e-02   pass

Performance Summary (GFlops)

Size   LDA    Align.  Average  Maximal
1000   1000   4       25.5531  26.3669
2000   2000   4       32.0423  32.8809
5000   5008   4       34.9039  35.4291
10000  10000  4       35.9328  36.1393
15000  15000  4       36.3961  36.8513
18000  18008  4       34.2916  34.6103
20000  20016  4       33.4721  34.2958

Residual checks PASSED

End of tests
```

- **Following is the screenshot of the HPL benchmark linpack for 2 thread which I executed :**

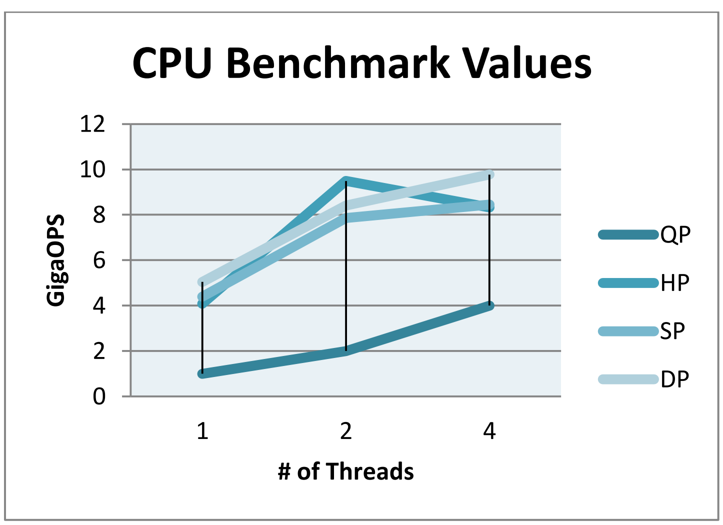
```
CPU frequency:    2.845 GHz
Number of CPUs: 2
Number of cores: 2
Number of threads: 2

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 30000 35000 40000 45000
Leading dimension of array                  : 1000  2000  5008  10000 15000 18008 20016 22008 25000 26000 27000 30000 35000 40000 45000
Number of trials to run                     : 4     2     2     2     2     2     2     2     2     2     1     1     1     1     1
Data alignment value (in Kbytes)            : 4     4     4     4     4     4     4     4     4     4     4     1     1     1     1

Maximum memory requested that can be used=3202964416, at the size=20000

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

Size   LDA    Align. Time(s)    GFlops   Residual     Residual(norm) Check
1000   1000   4      0.022      30.2522  9.394430e-13 3.203742e-02   pass
1000   1000   4      0.014      48.3450  9.394430e-13 3.203742e-02   pass
1000   1000   4      0.014      48.9039  9.394430e-13 3.203742e-02   pass
1000   1000   4      0.014      49.1671  9.394430e-13 3.203742e-02   pass
2000   2000   4      0.096      55.8980  4.085732e-12 3.554086e-02   pass
2000   2000   4      0.097      54.9130  4.085732e-12 3.554086e-02   pass
5000   5008   4      1.336      62.4118  2.262585e-11 3.154992e-02   pass
5000   5008   4      1.419      58.7757  2.262585e-11 3.154992e-02   pass
10000  10000  4      11.753     56.7404  9.187981e-11 3.239775e-02   pass
10000  10000  4      10.902     61.1664  9.187981e-11 3.239775e-02   pass
15000  15000  4      40.229     55.9411  2.219450e-10 3.495671e-02   pass
15000  15000  4      38.664     58.2051  2.219450e-10 3.495671e-02   pass
18000  18008  4      65.185     59.6556  2.886628e-10 3.161212e-02   pass
18000  18008  4      65.587     59.2902  2.886628e-10 3.161212e-02   pass
20000  20016  4      95.165     56.0512  3.669736e-10 3.248520e-02   pass
20000  20016  4      91.770     58.1250  3.669736e-10 3.248520e-02   pass

Performance Summary (GFlops)

Size   LDA    Align.  Average  Maximal
1000   1000   4       44.1671  49.1671
2000   2000   4       55.4055  55.8980
5000   5008   4       60.5938  62.4118
10000  10000  4       58.9534  61.1664
15000  15000  4       57.0731  58.2051
18000  18008  4       59.4729  59.6556
20000  20016  4       57.0881  58.1250

Residual checks PASSED

End of tests

Done: Tue Mar 27 05:18:42 UTC 2018
```

Pooja Patel
A20396099

- **Following is the screenshot of the HPL benchmark linpack for 4 thread which I executed :**

```
Sample data file lininput_xeon64.

Current date/time: Tue Mar 27 05:10:38 2018

CPU frequency:    1.388 GHz
Number of CPUs: 2
Number of cores: 2
Number of threads: 4

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 30000 35000 40000 45000
Leading dimension of array                  : 1000  2000  5008  10000 15000 18008 20016 22008 25000 26000 27000 30000 35000 40000 45000
Number of trials to run                     : 4     2     2     2     2     2     2     2     2     2     1     1     1     1     1
Data alignment value (in Kbytes)            : 4     4     4     4     4     4     4     4     4     4     4     1     1     1     1

Maximum memory requested that can be used=3202964416, at the size=20000

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

Size   LDA    Align. Time(s)    GFlops   Residual     Residual(norm) Check
1000   1000   4      0.040      16.7969  9.394430e-13 3.203742e-02   pass
1000   1000   4      0.032      21.1752  9.394430e-13 3.203742e-02   pass
1000   1000   4      0.028      23.7986  9.394430e-13 3.203742e-02   pass
1000   1000   4      0.029      23.3327  9.394430e-13 3.203742e-02   pass
2000   2000   4      0.183      29.2661  4.085732e-12 3.554086e-02   pass
2000   2000   4      0.183      29.1141  4.085732e-12 3.554086e-02   pass
5000   5008   4      1.464      56.9727  2.262585e-11 3.154992e-02   pass
5000   5008   4      1.429      58.3323  2.262585e-11 3.154992e-02   pass
10000  10000  4      9.596      69.4967  9.187981e-11 3.239775e-02   pass
10000  10000  4      9.572      69.6687  9.187981e-11 3.239775e-02   pass
15000  15000  4      33.529     67.1202  2.219450e-10 3.495671e-02   pass
15000  15000  4      32.393     69.4739  2.219450e-10 3.495671e-02   pass
18000  18008  4      56.479     68.8515  2.886628e-10 3.161212e-02   pass
18000  18008  4      57.088     68.1170  2.886628e-10 3.161212e-02   pass
20000  20016  4      77.483     68.8423  3.669736e-10 3.248520e-02   pass
20000  20016  4      77.742     68.6130  3.669736e-10 3.248520e-02   pass

Performance Summary (GFlops)

Size   LDA    Align.  Average  Maximal
1000   1000   4       21.2758  23.7986
2000   2000   4       29.1901  29.2661
5000   5008   4       57.6525  58.3323
10000  10000  4       69.5827  69.6687
15000  15000  4       68.2970  69.4739
18000  18008  4       68.4843  68.8515
20000  20016  4       68.7276  68.8423

Residual checks PASSED
```

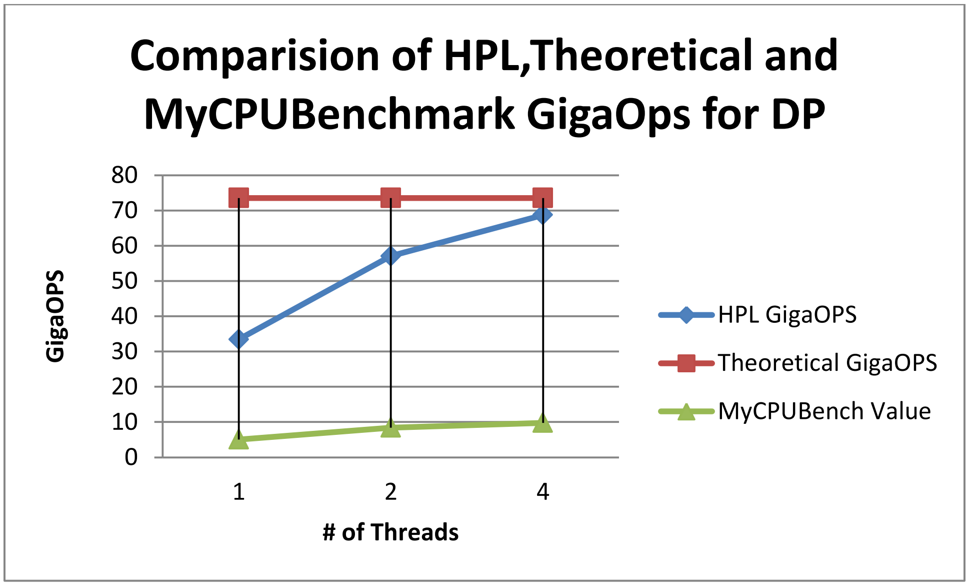
- **The below shows all graphs and charts for CPU Benchmark :**



Graph 1

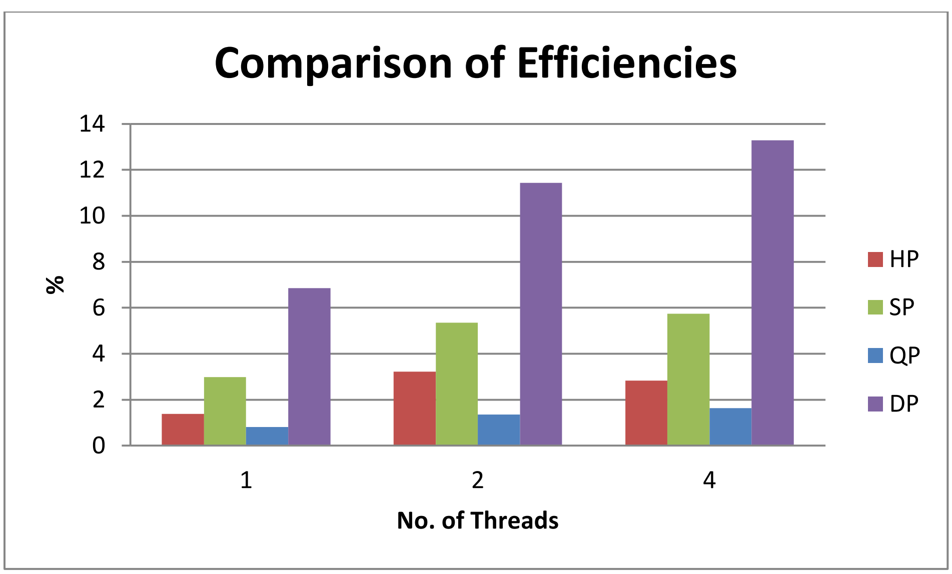
Pooja Patel
A20396099

- **Analysis for Graph 1 :** The above charts shows the my CPU Benchmark Values for QP,SP,HP,DP with 1,2,4 threads and I analyzed according to my experiments that as the no. of threads increases the ops/sec i.e. GigaOPs also increases except for HP .For HP, if we go from 1 thread to 2 thread GigaOps increases but from 2 thread to 4 threads it decreases.



Graph 2

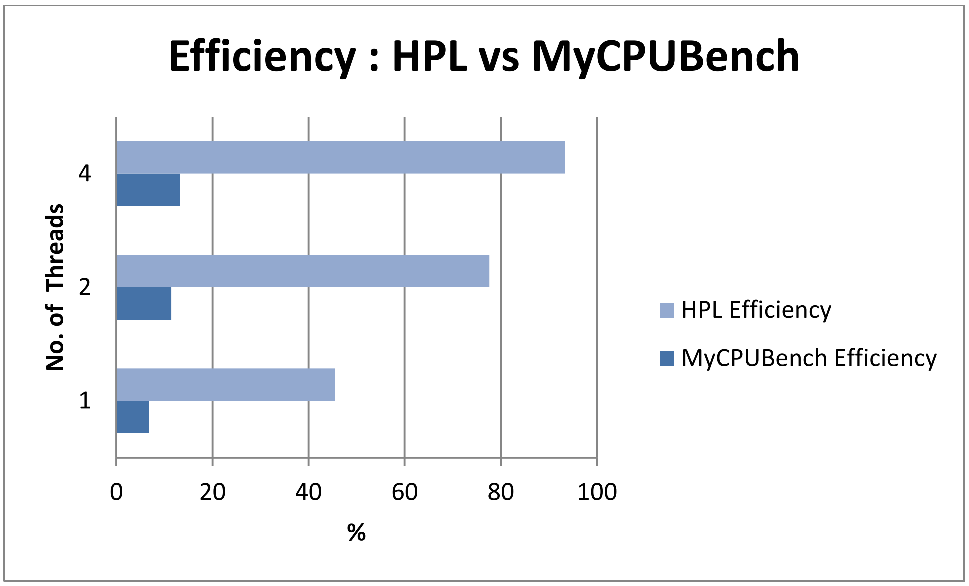
- **Analysis for Graph 2 :** In the above graph I am comparing the values which I got for MyCPUBenchmark , HPL and theoretical value for DP. I observed that for HPL and MyCPUBench as the no. of threads increases GigaOps also increases but the theoretical values remains the same for 1,2,4 threads



Pooja Patel
A20396099

Graph 3

- **Analysis for Graph 3 :** This graph shows the efficiency comparison for all precision and each threads.For, 1 thread DP has highest precision and then followed by SP then HP and lastly QP. Also the same is scenario for 2 and 4 threads.Also I observe that for each precision the highest efficiency is for 4 threads only , for instance, say for DP among 1,2,4 threads the highest efficiency is for 4 thread and then 2 followed by 1.The same is implied with other precisions too.



Graph 4

- **Analysis for Graph 4:** This graph compares the efficiency for HPL benchmark and MYCPUBench for different no. of threads. As per my analysis thread 4 has the highest efficiency for both HPL and MyCPUBench among all the threads.

2. MEMORY

- The memory experiments I have executed on Hyperion cluster
- I have calculated Theoretical throughput for memory (GB/sec) as using below formula:
  = clocks per sec * lines per clock * 64 bits per line * no. of interfaces
  = 2.133 Ghz * 2 * 64 * 2 / 8
  = 68.256 GB/sec
- MyRAMBench efficiency (%)= (MyRAMBench Measure throughput / Theoretical throughput) * 100
- pmbw efficiency (%) = (pmbw Measure throughput / Theretical throughput) * 100

Pooja Patel
A20396099

- **The following table 2 shows the memory throughput values (GB/sec):**

MEMORY							
Workload	Concurrency	Block Size	MyRAMBench Measured Throughput (GB/sec)	pmbw Measured Throughput (GB/sec)	Theoretical Throughput (GB/sec)	MyRAMBench Efficiency (%)	pmbw Efficiency (%)
RWS	1	1KB	27.682604	17.0934	68.256	40.55703	25.04307
RWS	1	1MB	27.795428	16.2943	68.256	40.72232	23.87233
RWS	1	10MB	24.785082	16.6644	68.256	36.31195	24.41456
RWS	2	1KB	54.339056	30.0942	68.256	79.61067	44.09019
RWS	2	1MB	50.513955	32.9942	68.256	74.00661	48.3389
RWS	2	10MB	43.611729	29.8839	68.256	63.89435	43.78209
RWS	4	1KB	50.548527	38.3133	68.256	74.05727	56.13177
RWS	4	1MB	45.817716	35.3751	68.256	67.12628	51.82709
RWS	4	10MB	33.039632	32.3993	68.256	48.40546	47.46733
RWR	1	1KB	6.669413	5.2105	68.256	9.771175	7.633761
RWR	1	1MB	9.097803	0.4836	68.256	13.32894	0.708509
RWR	1	10MB	8.7268	0.3138	68.256	12.7854	0.45974
RWR	2	1KB	4.320288	9.6309	68.256	6.329536	14.10997
RWR	2	1MB	17.347014	1.1654	68.256	25.41464	1.707396
RWR	2	10MB	13.639483	0.6789	68.256	19.98283	0.994638
RWR	4	1KB	3.912593	24.9305	68.256	5.732233	36.52499
RWR	4	1MB	16.148176	2.1791	68.256	23.65825	3.19254
RWR	4	10MB	14.537208	0.8075	68.256	21.29807	1.183046

Table 2 : MEMORY THROUGHPUT VALUES

- **Below table shows the latency part of memory:**

MEMORY LATENCY							
Workload	Concurrency	Block Size	MyRAMBench Measured Latency (microsec)	pmbw Measured Latency (microsec)	Theoretical Latency (microsec)	MyRAMBench Efficiency (%)	pmbw Efficiency (%)
RWS	1	1	0.006484	0.0004201	0.01406	46.11664	2.987909
RWS	2	1	0.003313	0.00029109	0.01406	23.5633	2.070341
RWS	4	1	0.003981	0.00022483	0.01406	28.31437	1.599075
RWR	1	1	0.125448	0.0015563	0.01406	892.2333	11.06899
RWR	2	1	0.294136	0.00082712	0.01406	2092.006	5.882788
RWR	4	1	0.331272	0.00030633	0.01406	2356.131	2.178734

Table 3 : MEMORY LATENCY VALUES

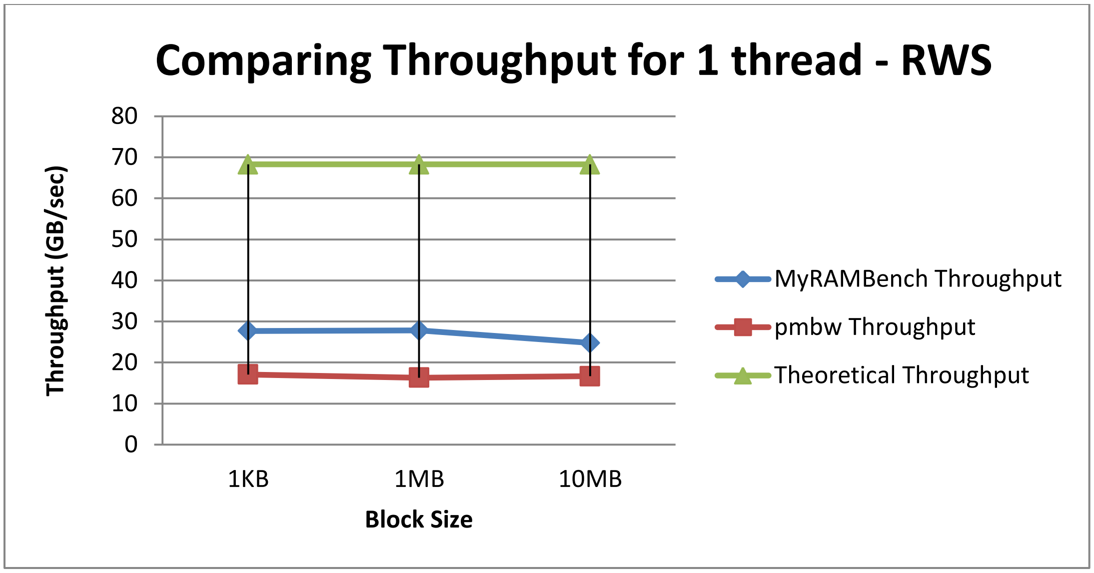
Pooja Patel
A20396099

- **The following is the screenshot of pmbw benchmark:**

```
ppatel115@bluecompute-10:~$ pmbw -p 1 -P 1 -s 1024 -S 1024 -M 1G -f ScanWrite64PtrSimpleLoop
Running benchmarks with at least 1 threads.
Running benchmarks with up to 1 threads.
Running benchmarks with array size at least 1024.
Running benchmarks with array size up to 1024.
Setting memory limit to 1073741824.
Running only functions containing 'ScanWrite64PtrSimpleLoop'
CPUID: mmx sse avx
Detected 3951 MiB physical RAM and 2 CPUs.

Allocating 512 MiB for testing.
Running nthreads=1 factor=1073741824 areasize=1024 thrsize=1024 testsize=1024 repeats=1048576 testvol=1073741824 testaccess=134217728
run time = 0.0999286 -> rerunning test with repeat factor=16117635518
Running nthreads=1 factor=16117635518 areasize=1024 thrsize=1024 testsize=1024 repeats=15739879 testvol=16117636096 testaccess=2014704512
run time = 0.888612 -> rerunning test with repeat factor=27206978145
Running nthreads=1 factor=27206978145 areasize=1024 thrsize=1024 testsize=1024 repeats=26569315 testvol=27206978560 testaccess=3400872320
run time = 1.59166 -> next test with repeat factor=25640132586
RESULT  datetime=2018-03-28 02:13:17    host=bluecompute-10     version=0.6.2   funcname=ScanWrite64PtrSimpleLoop       nthreads=1      areasize=1024   threa
dsize=1024      testsize=1024   repeats=26569315        testvol=27206978560     testaccess=3400872320   time=1.5916636820184066892      bandwidth=17093421724
.304548264      rate=4.6801630059972575738e-10
Skipping ScanWrite64PtrSimpleLoop test with 2048 maximum array size due to -S 1024.
Skipping ScanWrite64PtrSimpleLoop test with 3072 maximum array size due to -S 1024.
Skipping ScanWrite64PtrSimpleLoop test with 4096 maximum array size due to -S 1024.
Skipping ScanWrite64PtrSimpleLoop test with 6144 maximum array size due to -S 1024.
Skipping ScanWrite64PtrSimpleLoop test with 8192 maximum array size due to -S 1024.
Skipping ScanWrite64PtrSimpleLoop test with 12288 maximum array size due to -S 1024.
Skipping ScanWrite64PtrSimpleLoop test with 16384 maximum array size due to -S 1024.
Skipping ScanWrite64PtrSimpleLoop test with 20480 maximum array size due to -S 1024.
Skipping ScanWrite64PtrSimpleLoop test with 24576 maximum array size due to -S 1024.
Skipping ScanWrite64PtrSimpleLoop test with 28672 maximum array size due to -S 1024.
Skipping ScanWrite64PtrSimpleLoop test with 32768 maximum array size due to -S 1024.
Skipping ScanWrite64PtrSimpleLoop test with 40960 maximum array size due to -S 1024.
Skipping ScanWrite64PtrSimpleLoop test with 49152 maximum array size due to -S 1024.
Skipping ScanWrite64PtrSimpleLoop test with 65536 maximum array size due to -S 1024.
Skipping ScanWrite64PtrSimpleLoop test with 98304 maximum array size due to -S 1024.
```

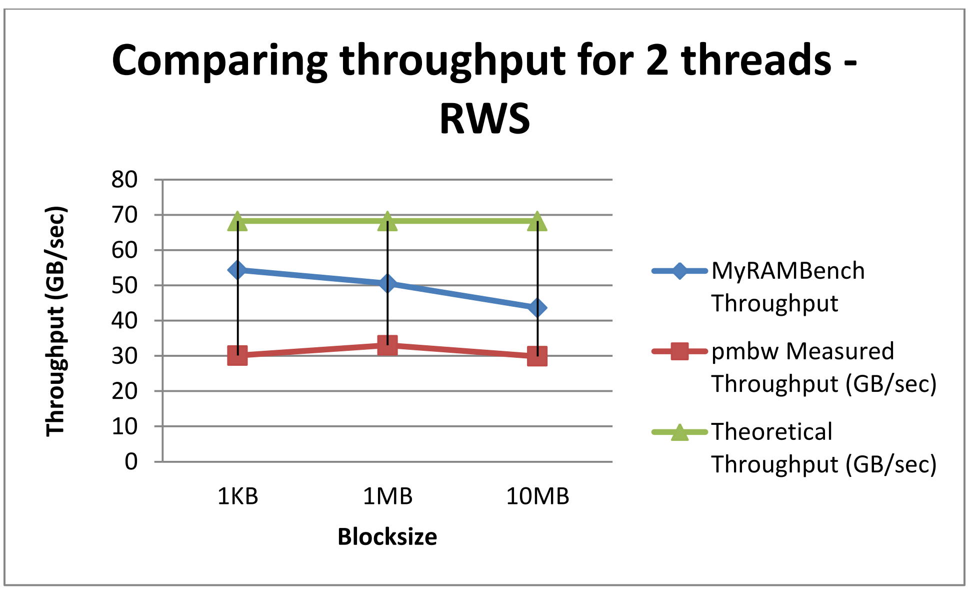
```
--- bluecompute-2 ping statistics ---
26 packets transmitted, 26 received, 0% packet loss, time 25118ms
rtt min/avg/max/mdev = 0.919/1.925/2.864/0.408 ms
ppatel115@hyperionides:~$ srun -n 1 -p interactive --pty /bin/bash
ppatel115@bluecompute-3:~$ pmbw -p 4 -P 4 -s 1048576 -S 1048576 -M 1G -f ScanWrite64PtrSimpleLoop
Running benchmarks with at least 4 threads.
Running benchmarks with up to 4 threads.
Running benchmarks with array size at least 1048576.
Running benchmarks with array size up to 1048576.
Setting memory limit to 1073741824.
Running only functions containing 'ScanWrite64PtrSimpleLoop'
CPUID: mmx sse avx
Detected 3951 MiB physical RAM and 2 CPUs.

Allocating 512 MiB for testing.
Skipping ScanWrite64PtrSimpleLoop test with 1024 minimum array size due to -s 1048576.
Skipping ScanWrite64PtrSimpleLoop test with 2048 minimum array size due to -s 1048576.
Skipping ScanWrite64PtrSimpleLoop test with 3072 minimum array size due to -s 1048576.
Skipping ScanWrite64PtrSimpleLoop test with 4096 minimum array size due to -s 1048576.
Skipping ScanWrite64PtrSimpleLoop test with 6144 minimum array size due to -s 1048576.
Skipping ScanWrite64PtrSimpleLoop test with 8192 minimum array size due to -s 1048576.
Skipping ScanWrite64PtrSimpleLoop test with 12288 minimum array size due to -s 1048576.
Skipping ScanWrite64PtrSimpleLoop test with 16384 minimum array size due to -s 1048576.
Skipping ScanWrite64PtrSimpleLoop test with 20480 minimum array size due to -s 1048576.
Skipping ScanWrite64PtrSimpleLoop test with 24576 minimum array size due to -s 1048576.
Skipping ScanWrite64PtrSimpleLoop test with 28672 minimum array size due to -s 1048576.
Skipping ScanWrite64PtrSimpleLoop test with 32768 minimum array size due to -s 1048576.
Skipping ScanWrite64PtrSimpleLoop test with 40960 minimum array size due to -s 1048576.
Skipping ScanWrite64PtrSimpleLoop test with 49152 minimum array size due to -s 1048576.
Skipping ScanWrite64PtrSimpleLoop test with 65536 minimum array size due to -s 1048576.
Skipping ScanWrite64PtrSimpleLoop test with 98304 minimum array size due to -s 1048576.
Skipping ScanWrite64PtrSimpleLoop test with 131072 minimum array size due to -s 1048576.
Skipping ScanWrite64PtrSimpleLoop test with 196608 minimum array size due to -s 1048576.
Skipping ScanWrite64PtrSimpleLoop test with 262144 minimum array size due to -s 1048576.
Skipping ScanWrite64PtrSimpleLoop test with 393216 minimum array size due to -s 1048576.
Skipping ScanWrite64PtrSimpleLoop test with 524288 minimum array size due to -s 1048576.
Skipping ScanWrite64PtrSimpleLoop test with 786432 minimum array size due to -s 1048576.
Running nthreads=4 factor=1073741824 areasize=1048576 thrsize=262144 testsize=1048576 repeats=4096 testvol=4294967296 testaccess=536870912
run time = 0.169859 -> rerunning test with repeat factor=9482065132
Running nthreads=4 factor=9482065132 areasize=1048576 thrsize=262144 testsize=1048576 repeats=36172 testvol=37929091072 testaccess=4741136384
run time = 1.07008 -> next test with repeat factor=13291922495
RESULT  datetime=2018-03-29 05:01:22    host=bluecompute-3      version=0.6.2   funcname=ScanWrite64PtrSimpleLoop       nthreads=4      areasize=1048576        t
hreadsize=262144        testsize=1048576        repeats=36172   testvol=37929091072     testaccess=4741136384   time=1.0700791520066559315      bandwidth=354
45126653.363746643      rate=2.2570098502499773404e-10
Skipping ScanWrite64PtrSimpleLoop test with 1310720 maximum array size due to -S 1048576.
```

- I have used –p and –P for the number of threads, -s and –S for the blocksize , -M for maximum size and –f for the files which I am using. The files are ScanWrite64PtrUnrollLoop,ScanRead64PtrUnrollLoop, PermRead64UnrollLoop



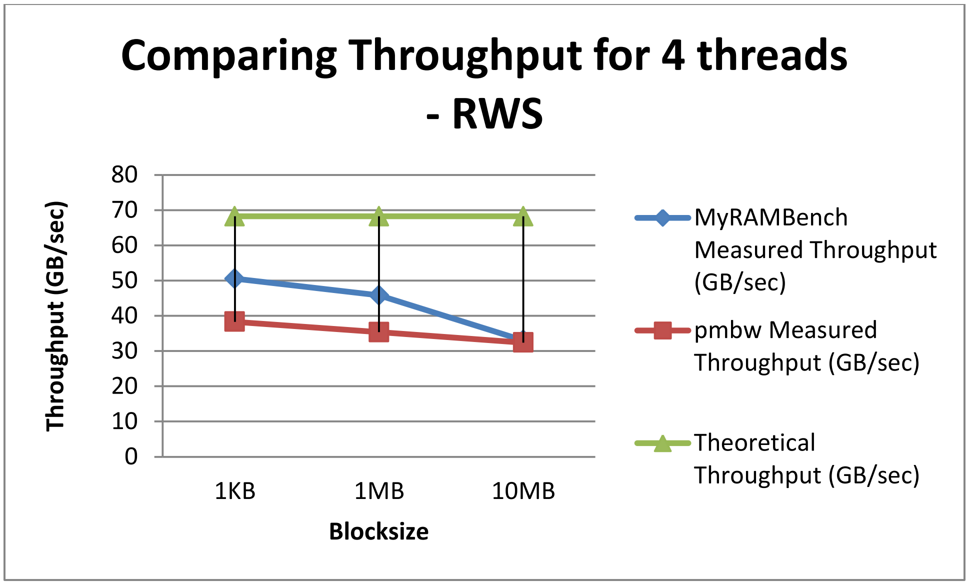
Graph 5

- **Analysis for Graph 5 :** Here,I am comparing the throughputs in GB/sec for 1 thread and for different blocksize 1KB, 1MB,10MB.The theoretical values are more than MyRAmBench and pmbw throughputs but the pmbw throughputs are less than MyRAMBench throughputs.Hence, the efficiency of MyRAMBench is better than pmbw throughput values.



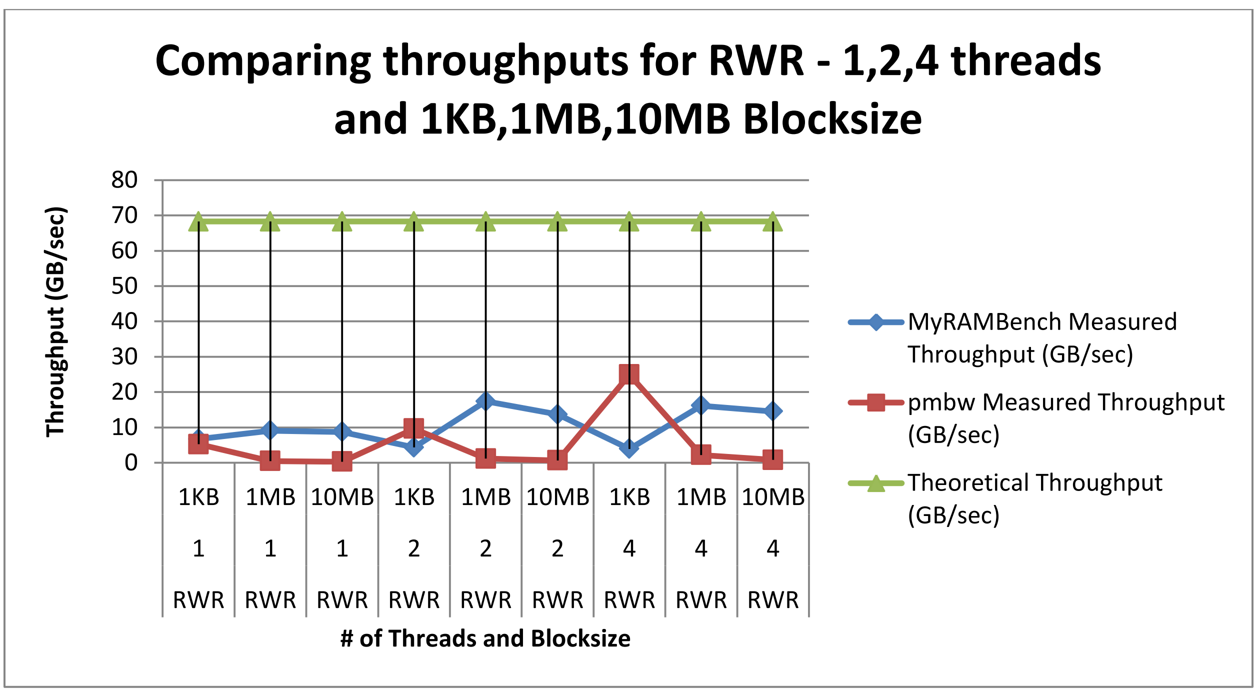
Graph 6

- **Analysis for Graph 6:** I compare the blocksize with throughput values of MyRAMBench, pmbw , theoretical. As the blocks size increased from 1MB to 10 MB for both pmbw and MyRAMBench the throughput values increased. Also I deduced that MyRAMBench throughputs are better than the pmbw throughputs. The theoretical values are more than the other two throughputs.



Graph 7

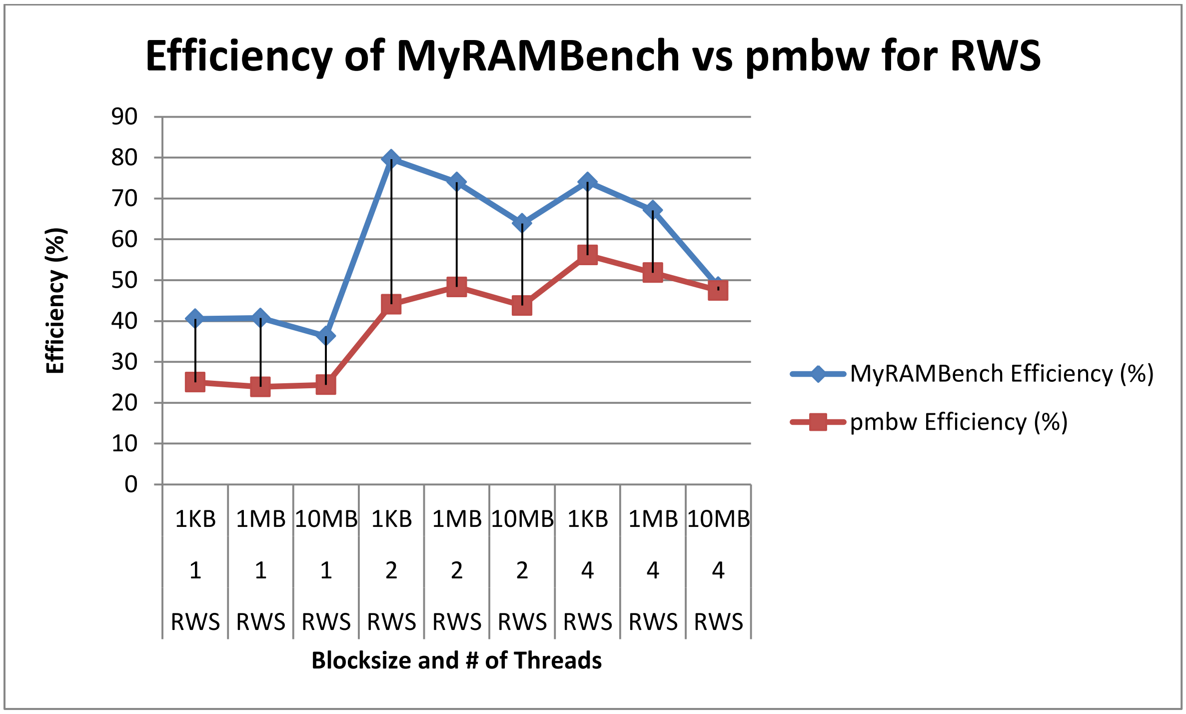
- **Analysis for Graph 7 :** I analyzed that for 4 threads , RWS and for different blocksize MyRAMBench and pmbw values decreased from increasing blocksize from 1MB to 10 MB. Also in this case pmbw measured values are lower than the ones I got.



Graph 8

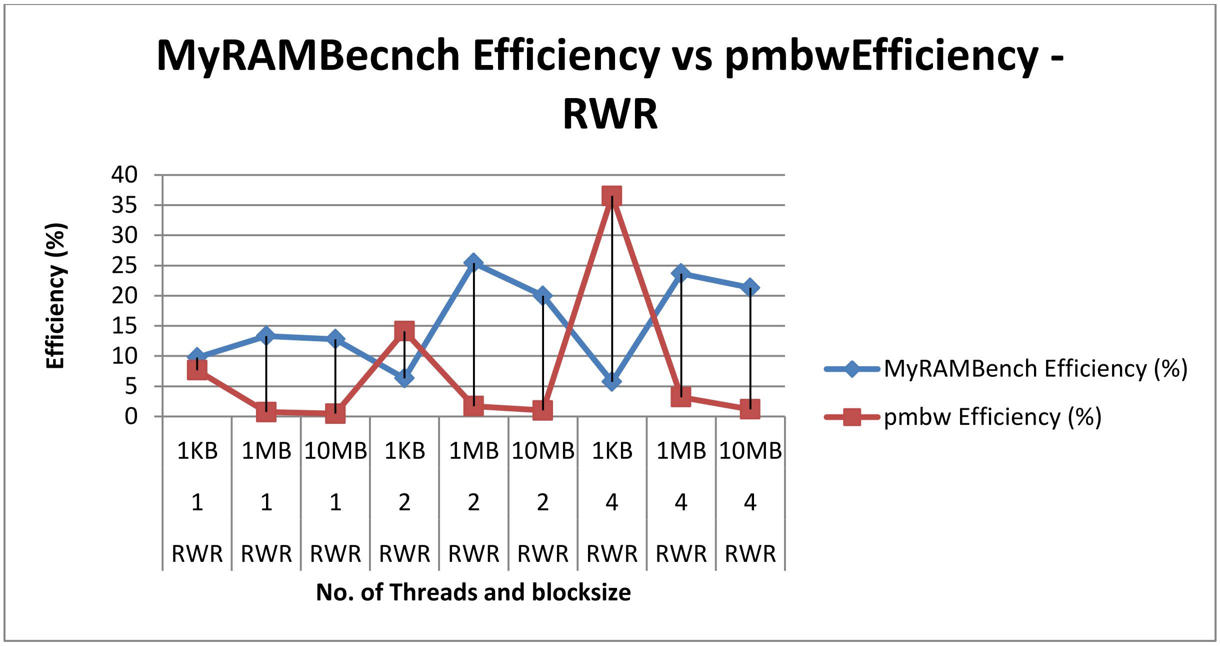
Pooja Patel
A20396099

- **Analysis for Graph 8 :** This graph compares throughput for 1,2,4 threads and also for different blocksize. It also shows MyRAMBench values , theoretical values and pmbw measured values. My analysis for this graph is that as the no. of threads increases the throughput also increases for both MyRAMBench and pmbw. But for my measured values when the blocksize increases the throughput values have decreased for 1,2,4 threads. The same happened in pmbw case from blocksize 1MB to 10MB for all the threads.



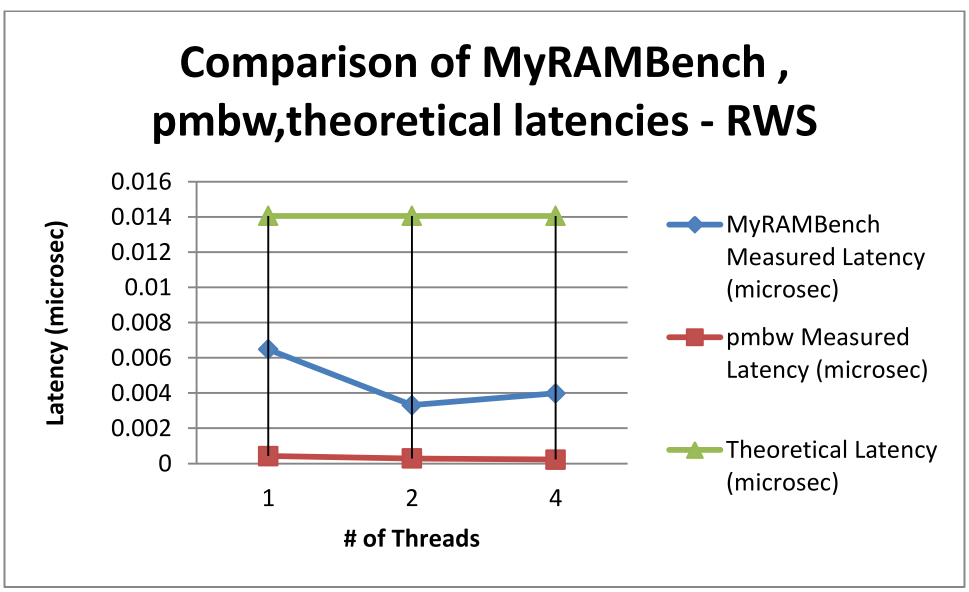
Graph 9

- **Analysis for Graph 9** : I compare the efficiency of my measured values and pmbw for RWS access pattern. My measured values had higher efficiency than the pmbw's efficiency. Also the efficiency tends to decrease from 1KB blocksize to 1MB and also from 1 MB to 10 MB. Also for 4 threads and 10MB my measured efficiency and the pmbw efficiency are almost near to each other.



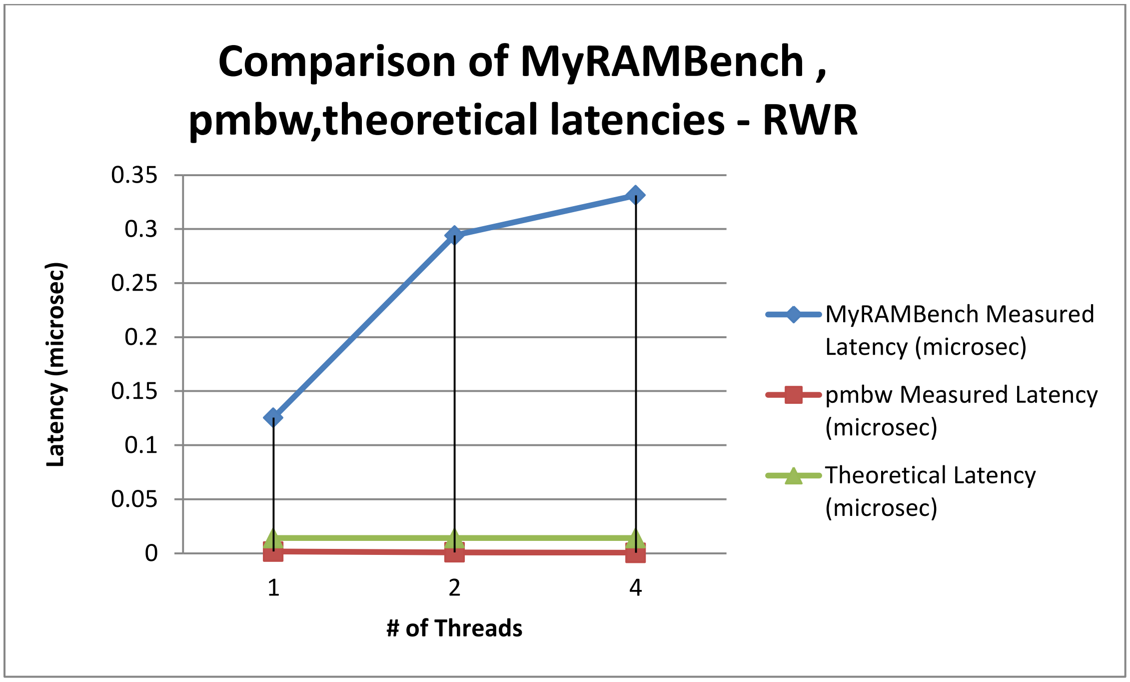
Graph 10

- **Analysis for Graph 10 :** This graph compared the efficiency of MyRAMBench vs pmbw for RWR. I observed that the efficiency values increased by increasing the no. of threads for both MyRAM and pmbw. Also MyRAM efficiency are higher than the pmbw efficiencies.



Graph 11

- **Analysis for Graph 11 :** This graph shows the latency values which I got through my experiments , pmbw measured and theoretical values. MyRAMBench latencies are more than the pmbw values.



Graph 12

- **Analysis for Graph 12:** Here I compare the latencies for RWR access pattern where MyRAMBench value for latency increased by increasing the no. of threads. RWR I got more latency for MyRAMBench compared to pmbw and theoretical latency values.

3. DISK

- **This experiments is done on Prometheus cluster**
- **The disk on Prometheus is :** Disk type is Micron 5100 PRO 2.5"480GB,SATA,6Gb/s,3D NAND,7mm,1.5DWPD 2
- **The theoretical values for disk throughput are calculated in MB/sec using following :**

Pooja Patel
A20396099

Micron 5100 PRO 2.5", 480GB, SATA, 6Gb/s, 3D NAND, 7mm, 1.5DWPD

MTFDDAK480TCB-1AR1ZABYY

Features

Order Information

Ordering Part #:	MTFDDAK480TCB-1AR1ZABYY

SSD/HDD Specification

Bauhöhe:	7mm
Interface:	SATA 3.0
MTBF/MTTF:	2M hrs
Power Idle:	2.5W
Power R/W:	4.5W
SSD Capacity:	480GB
Xfer Rate:	6Gbps
Form Factor:	2.5"
DWPD:	1.5

Storage Performance

Sequential Read (up to):	540MB/s
Sequential Write (up to):	410MB/s
Random Read (up to):	93000 IOPS
Random Write (up to):	43000 IOPS
Latency - Read:	500 µs
Latency - Write:	500 µs

Figure : Micron 5100 PRO 2.5"480GB,SATA,6Gb/s,3D NAND,7mm,1.5DWPD 2 (Taken from https://www.sysgen.de/micron-5100-pro-2.5-480gb-sata-6gb-s-3d-nand-7mm-1.5dwpd.html)

Pooja Patel
A20396099

- **Below Table 4 shows the throughput and efficiency values :**

Disk Throughput							
Work-load	**Con-currency**	**Block Size**	**MyDisk Bench Measured Throughput(MB /sec)**	**IOZoneMeasured Throughput (MB/sec)**	**Theoretical Throughput (MB/sec)**	**MyDiskBench Efficiency(%)**	**IOZone Efficiency (%)**
RS	1	1MB	324.437852	413.56	540	60.08108	76.58519
RS	1	10MB	321.747282	377.45	540	59.58283	69.89815
RS	1	100MB	314.449234	350.89	540	58.23134	64.97963
RS	2	1MB	224.396378	300.34	1080	20.77744	27.80926
RS	2	10MB	622.626206	756.45	1080	57.65057	70.04167
RS	2	100MB	638.944322	803.23	1080	59.16151	74.37315
RS	4	1MB	918.0215	1032.34	2160	42.501	47.79352
RS	4	10MB	1152.253968	1204.53	2160	53.34509	55.76528
RS	4	100MB	1127.415125	1137.56	2160	52.19514	52.66481
WS	1	1MB	316.660454	340.34	410	77.23426	83.00976
WS	1	10MB	308.943431	329.52	410	75.35206	80.37073
WS	1	100MB	214.403484	250.23	410	52.29353	61.03171
WS	2	1MB	310.540089	423.67	820	37.87074	51.66707
WS	2	10MB	311.351668	345.34	820	37.96972	42.11463
WS	2	100MB	303.033841	333.73	820	36.95535	40.69878
WS	4	1MB	327.497134	387.28	1640	19.96934	23.61463
WS	4	10MB	167.264938	190.63	1640	10.19908	11.62378
WS	4	100MB	309.187423	423.34	1640	18.85289	25.81341
RR	1	1MB	263.933309	311.63	372	70.94981	83.77151
RR	1	10MB	85.30059	179.56	372	22.93027	48.26882
RR	1	100MB	273.388753	299.34	372	73.4916	80.46774
RR	2	1MB	244.747861	355.46	744	32.89622	47.77688
RR	2	10MB	618.666284	709.45	744	83.15407	95.35618
RR	2	100MB	310.926355	634.34	744	41.79118	85.26075
RR	4	1MB	470.654602	984.23	1488	31.63001	66.14449
RR	4	10MB	1202.755964	1345.34	1488	80.83037	90.41263
RR	4	100MB	97.809626	232.63	1488	6.573228	15.63374
WR	1	1MB	194.506518	187.34	172	113.0852	108.9186
WR	1	10MB	190.97221	198.45	172	111.0304	115.3779
WR	1	100MB	185.01286	160.74	172	107.5656	93.45349
WR	2	1MB	207.197665	323.45	344	60.23188	94.02616
WR	2	10MB	308.819832	315.34	344	89.77321	91.6686
WR	2	100MB	170.838829	298.44	344	49.66245	86.75581
WR	4	1MB	126.46198	234.65	688	18.3811	34.1061

WR	4	10MB	309.960709	456.34	688	45.05243	66.32849
WR	4	100MB	315.168917	498.25	688	45.80944	72.42006

Table 4 : DISK THROUGHPUT VALUES

- **The following table 5 shows the disk latency values in ms for MyDiskBench , IoZone , Theoretical values and the efficiencies**

Disk Latency(Measured in ms)							
Work-load	**Con-currency**	**Block Size**	**MyDiskBench Measured Latency(ms)**	**IOZone Measured Latency(ms)**	**Theoretical Latency(ms)**	**MyDiskBench Efficiency(%)**	**IOZone Efficiency(%)**
RR	1	1KB	0.001	0.0005	0.5	0.2	0.1
RR	2	1KB	0.002	0.0007	0.5	0.4	0.14
RR	4	1KB	0.001	0.00032	0.5	0.2	0.064
RR	8	1KB	0.001	0.00035	0.5	0.2	0.07
RR	16	1KB	0.008	0.00015	0.5	1.6	0.03
RR	32	1KB	0.766	0.0001	0.5	153.2	0.02
RR	64	1KB	0.266	0.00002	0.5	53.2	0.004
RR	128	1KB	0.068	0.00045	0.5	13.6	0.09
WR	1	1KB	0.001	0.003	0.5	0.2	0.6
WR	2	1KB	0.0015	0.0002	0.5	0.3	0.04
WR	4	1KB	0.001	0.00045	0.5	0.2	0.09
WR	8	1KB	0.003	0.00027	0.5	0.6	0.054
WR	16	1KB	0.006	0.00025	0.5	1.2	0.05
WR	32	1KB	0.009	0.00013	0.5	1.8	0.026
WR	64	1KB	0.012	0.00007	0.5	2.4	0.014
WR	128	1KB	0.028	0.00003	0.5	5.6	0.006

Table 5 : DISK LATENCY VALUES (ms)

- **Table 6 shows the Disk latency in terms of IOPS**

Disk Latency (Measured in IOPS)							
Work-load	**Con-currency**	**Block Size**	**MyDisk Bench Measured IOPS**	**IOZone Measured IOPS**	**Theoretical IOPS**	**MyDiskBench Efficiency(%)**	**IOZone Efficiency(%)**
RR	1	1KB	70544	80445	93000	75.85376	86.5
RR	2	1KB	75456	77455	93000	81.13548	83.28495
RR	4	1KB	67990	72345	93000	73.10753	77.79032

RR	8	1KB	73456	89545	93000	78.98495	96.28495
RR	16	1KB	65444	67634	93000	70.36989	72.72473
RR	32	1KB	78445	88333	93000	84.34946	94.98172
RR	64	1KB	56766	75566	93000	61.03871	81.25376
RR	128	1KB	86433	89845	93000	92.93871	96.60753
WR	1	1KB	9567	12566	43000	22.24884	29.22326
WR	2	1KB	23565	35565	43000	54.80233	82.7093
WR	4	1KB	27656	37545	43000	64.31628	87.31395
WR	8	1KB	12787	25445	43000	29.73721	59.17442
WR	16	1KB	34556	36566	43000	80.36279	85.03721
WR	32	1KB	22454	30565	43000	52.2186	71.0814
WR	64	1KB	27567	32454	43000	64.1093	75.47442
WR	128	1KB	19566	24457	43000	45.50233	56.87674

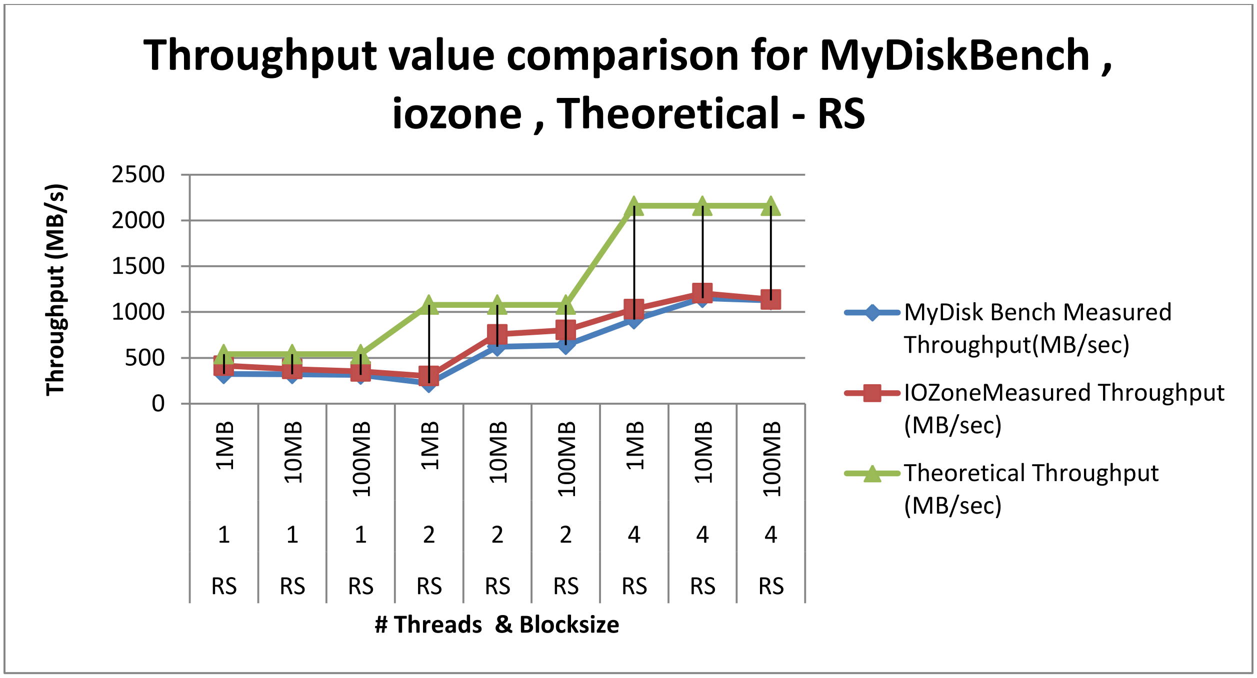
Table 6 : DISK LATENCY VALUES (IOPS)

- **Following is the screenshot of IoZone which I executed for different threads and blocksize :**

```
Iozone: Performance Test of File I/O
                Version $Revision: 3.471 $
                Compiled for 64 bit mode.
                Build: linux-AMD64

        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: Tue Mar 27 08:27:21 2018

        Excel chart generation enabled
        File size set to 10485760 kB
        Record Size 1024 kB
        O_DIRECT feature enabled
        OPS Mode. Output is in operations per second.
        O_DIRECT feature enabled
        Command line used: iozone -R -i 0 -i 1 -s 10g -r 1M -t 1 -F /tmp/g1 -u 1 -I -O -I /tmp/g2 /tmp/g3 /tmp/g4 /tmp/g5 /tmp/g6 /tmp/g7 /tmp/g8 /tmp/g9 /tmp/g10
        /tmp/g11 /tmp/g12 /tmp/g13 /tmp/g14 /tmp/g15 /tmp/g16 1
        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.
        Min process = 1
        Max process = 1
        Throughput test with 1 process
        Each process writes a 10485760 kByte file in 1024 kByte records

        Children see throughput for  1 initial writers  =     290.20 ops/sec
        Parent sees throughput for  1 initial writers   =     290.18 ops/sec
        Min throughput per process                      =     290.20 ops/sec
        Max throughput per process                      =     290.20 ops/sec
        Avg throughput per process                      =     290.20 ops/sec
        Min xfer                                        =   10240.00 ops
```

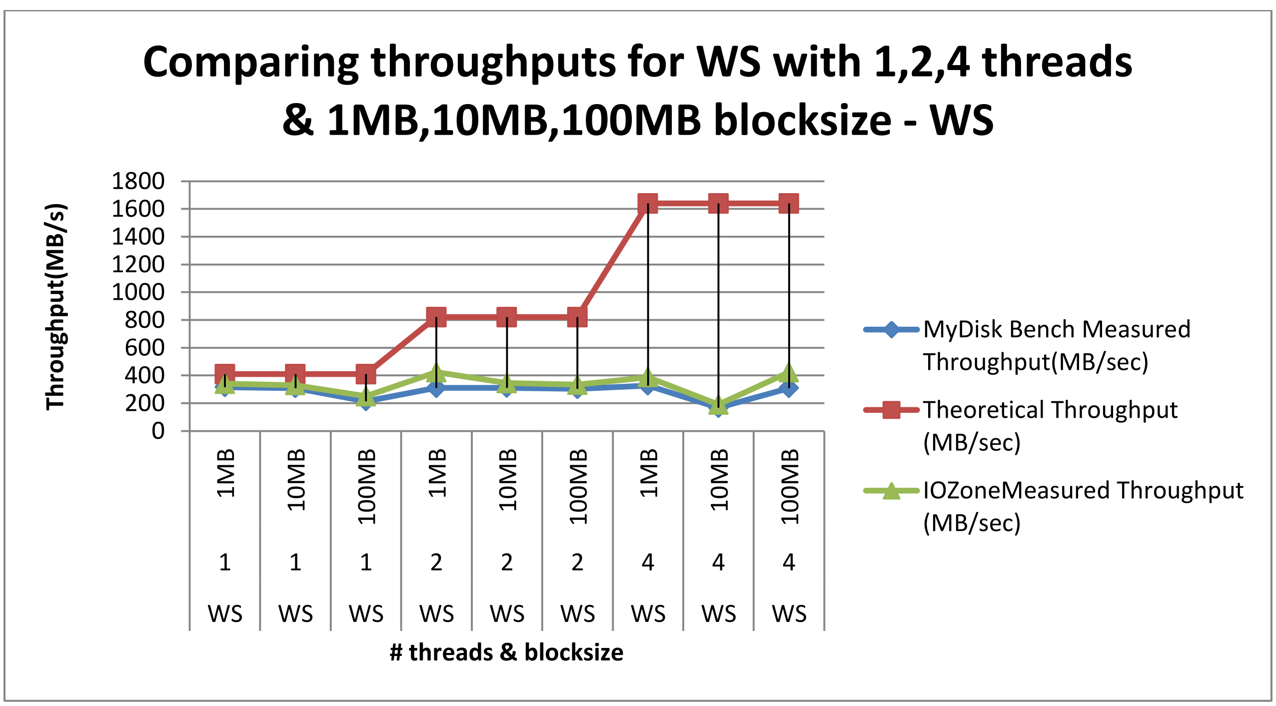
- **Following shows the my analysis from the graphs:**

Pooja Patel
A20396099



Graph 13

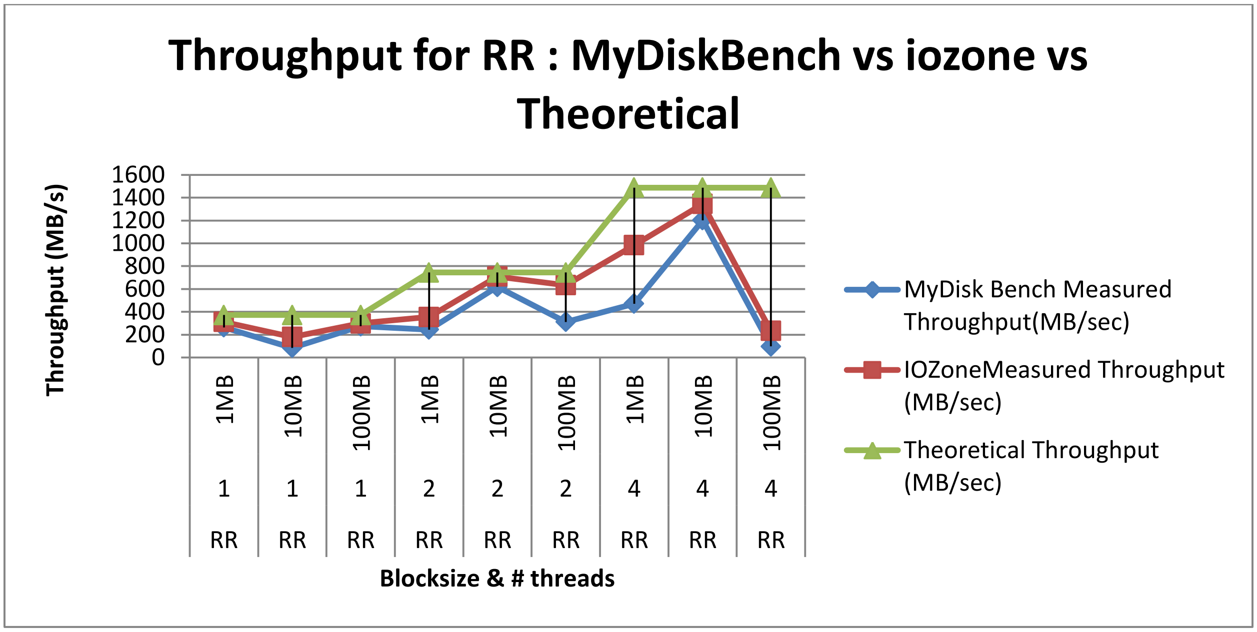
- **Analysis for Graph 13:** From this graph my analysis follows as, as the no. of threads increased for RS the throughput values also increased for all MyDisk, IOZone and theoretical values. If I changed the blocksize then for IOZone and MyDiskBench the values changed but the theoretical values for different blocksize and same no. of threads remained the same.



Graph 14

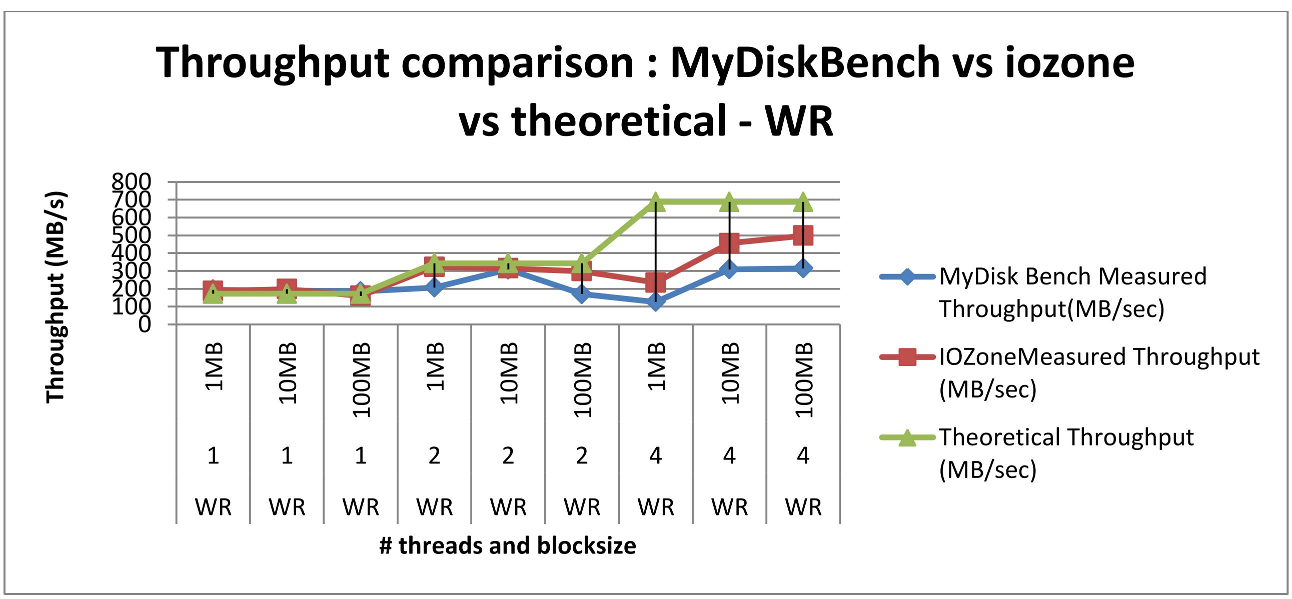
Pooja Patel
A20396099

- **Analysis for Graph 14 :** This graph compares throughput for all MyDisk , IOZone and theoretical values for WS access pattern and for different no. of threads and various blocksizes.



Graph 15

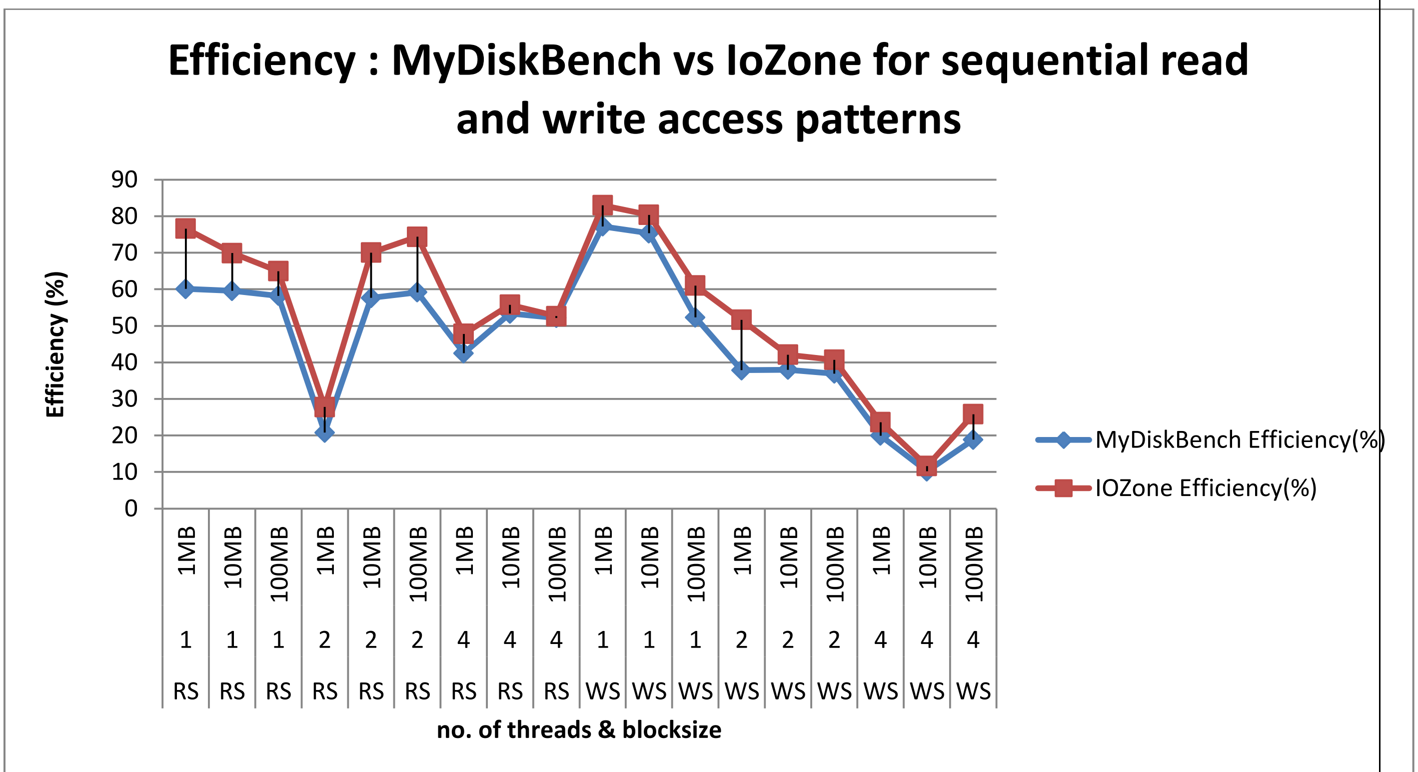
- **Analysis for Graph 15:** In this graph I compare the throughput for MyDiskBench , IOZone and theoretical value for RR access pattern. As the no. of threads increased the throughput values also increased. Say for 10MB blocksize the throughput for 4 threads is more than 2 threads and the throughput for 2 threads is more than the throughput for 1 thread.



Pooja Patel
A20396099

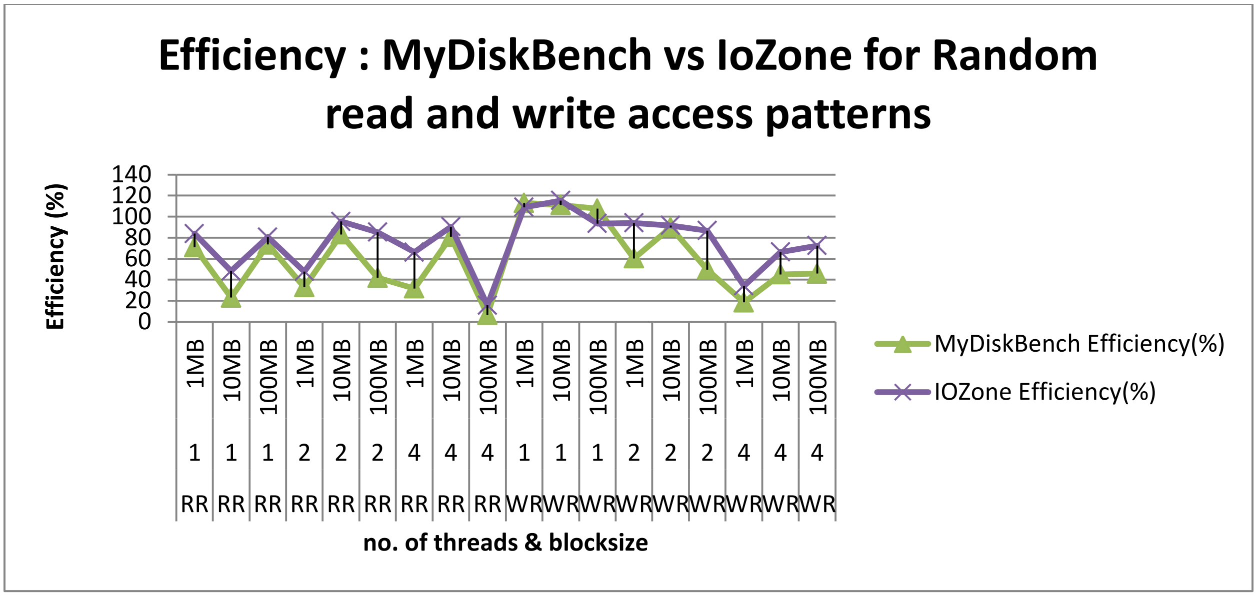
Graph 16

- **Analysis for Graph 16 :** For the access pattern WR, according to my analysis and the values I have obtained for 1 thread MyDiskBench , IOZone and Theoretical values are almost near to one another. For WR the throughput values are ranging from 100 – 700 MB/sec.



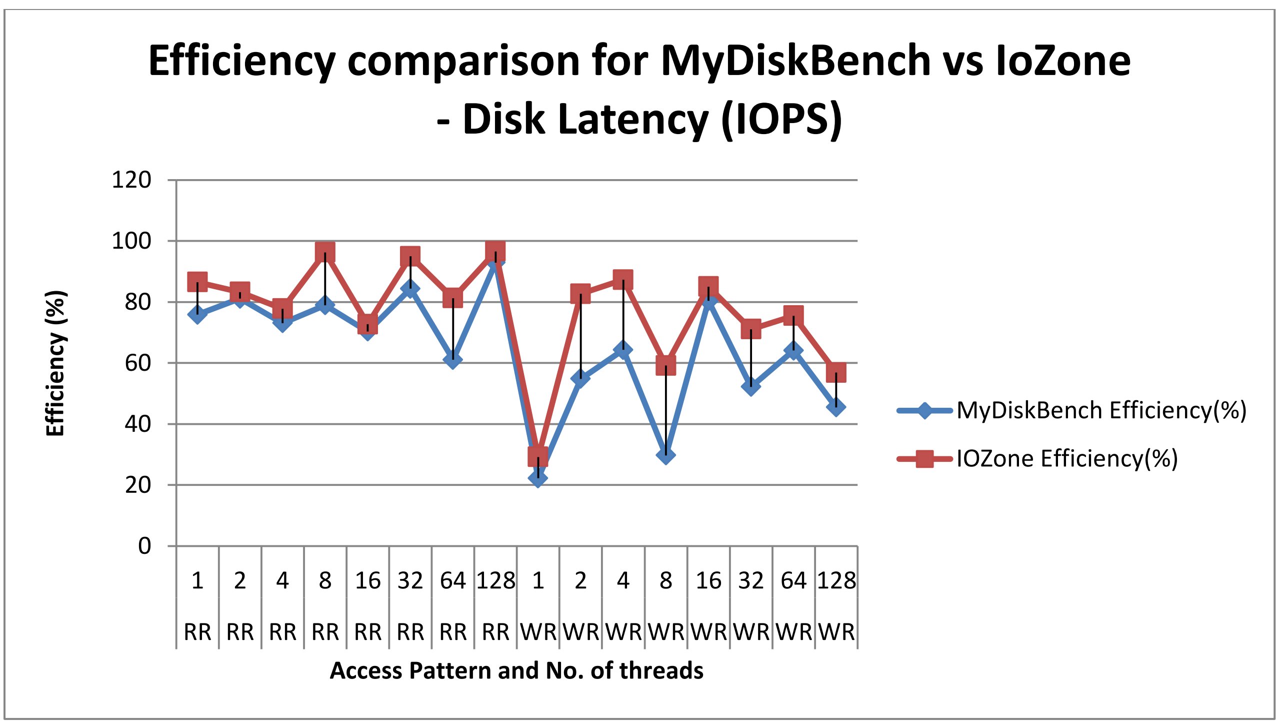
Graph 17

- **Analysis for Graph 17 :** Here, I compare the efficiencies of MyDiskBench and IOZone for both read and write sequential access pattern. For Read sequential pattern the efficiency was high for 1 thread and then it dropped for 2 threads and then again it raised for 4 threads for both MyDisk and IOZone. On the other hand for write sequential the efficiency was high for 1 thread and then for 2 then it went down and again for 4 threads it was dropped for MyDisk and IOZone too.



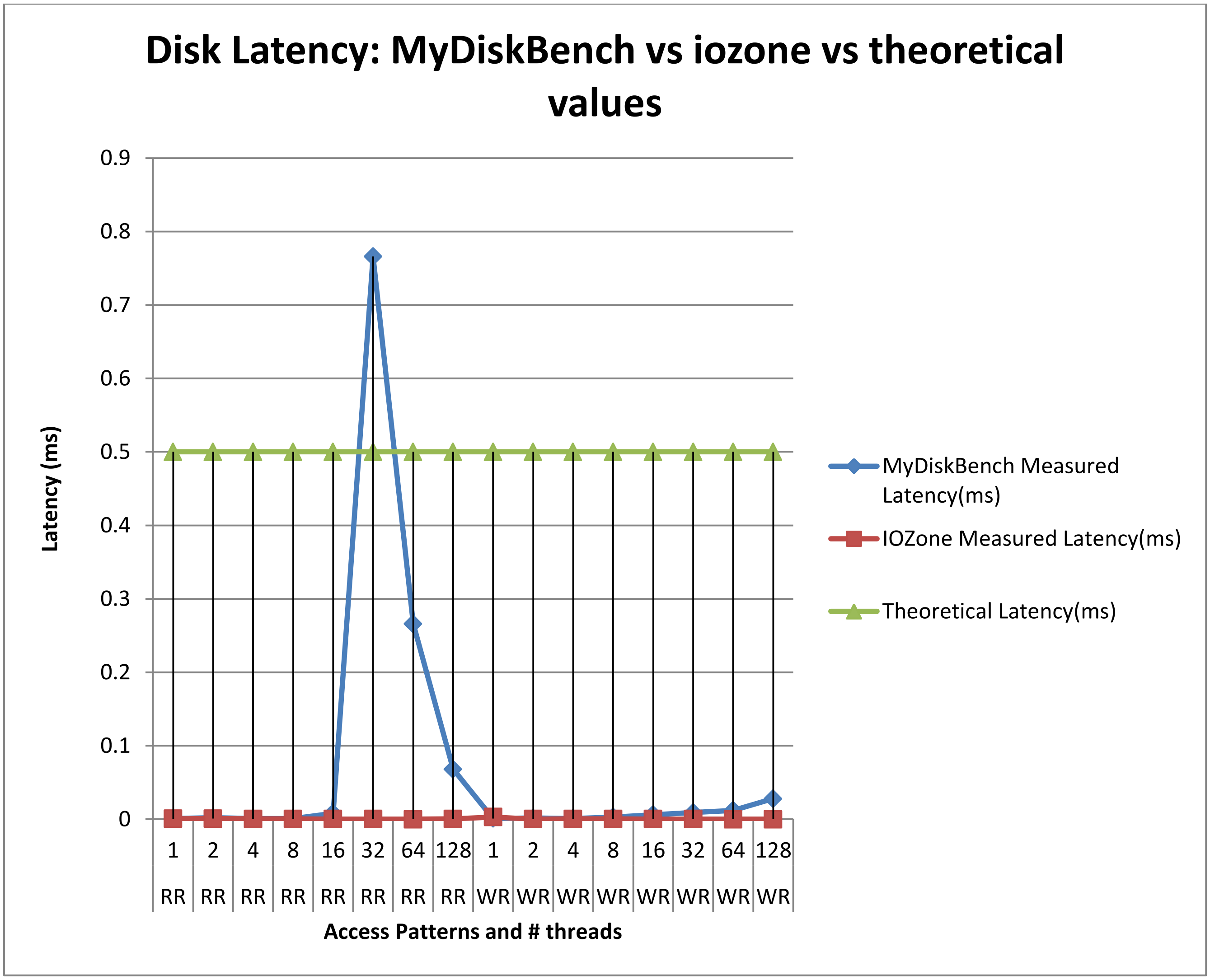
Graph 18

- **Analysis for Graph 18:** Here, I compare the efficiencies of MyDiskBench and IOZone for both read and write random access pattern. For Read random pattern the efficiency kept fluctuating for both MyDisk and IOZone. On the other hand for write sequential the efficiency was high for 1 thread and then for 2 then it went down as shown in graph for MyDisk and IOZone too.



Graph 19

- **Analysis for Graph 19 :** As the no. of threads increases for both RR and WR the IOPS sometimes increased and sometimes decreased but I observed that the IOPS for WR are less than the IOPS for RR



Graph 20

- **Analysis for Graph 20 :** This graph shows the latency for MyDisk, IOZone and Theoretical latency in ms.According to my analysis as the threads kept increasing for my measured values the latency highly increased from increasing no. of threads from 16 to 128.Rather the latency for IOZone is too low.

Pooja Patel
A20396099

4. NETWORK

- The network experiments are performed on hyperion and it uses Fourteen Data Rate (FDR) Infiniband network (56Gbps)
- Hence the theoretical throughput is 56000Mbps and the theoretical latency is 0.0007 ms
- Following is the screenshot :

Network

Networking is changing rapidly, and the network fabric is as much a part of the research focus of Chameleon as the compute or storage. For the Chameleon network, every switch in the research network is a fully OpenFlow compliant programmable Dell S6000-ON switch. Each node connects to this network at 10 Gbps, and each unit uplinks with 40Gbps per rack to the Chameleon core network. The core switches (Dell S6000-ON) are connected by 40 Gbps Ethernet links, which connect to the backbone 100Gbps services at both UC and TACC. A Fourteen Data Rate (FDR) Infiniband network (56Gbps) is also deployed on one SCU to allow exploration of alternate networks.

Figure : Shows hyperion network spec – FDR Infiniband network
(Taken from : https://www.chameleoncloud.org/about/hardware-description/)

Characteristics

	SDR	DDR	QDR	FDR10	FDR	EDR	HDR	NDR	XDR
Signaling rate (Gbit/s)	2.5	5	10	10.3125	14.0625[6]	25.78125	50	100	250
Theoretical effective throughput, Gbs, per 1x[7]	2	4	8	10	13.64	25	50	100	250
Speeds for 4x links (Gbit/s)	8	16	32	40	54.54	100	200	400	1000
Speeds for 8x links (Gbit/s)	16	32	64	80	109.08	200	400	800	2000
Speeds for 12x links (Gbit/s)	24	48	96	120	163.64	300	600	1200	3000
Encoding (bits)	8/10	8/10	8/10	64/66	64/66	64/66	64/66	?	?
Adapter latency (microseconds)[8]	5	2.5	1.3	0.7	0.7	0.5	less?	?	?
Year[9]	2001, 2003	2005	2007	2011	2011	2014[7]	2017[7]	after 2020	future (2023?)

Figure : Shows FDR Infiniband network latency
(Taken from : https://en.wikipedia.org/wiki/InfiniBand)

- **Network Throughput:** Below table 7 shows the throughput for iperf for TCP and UDP

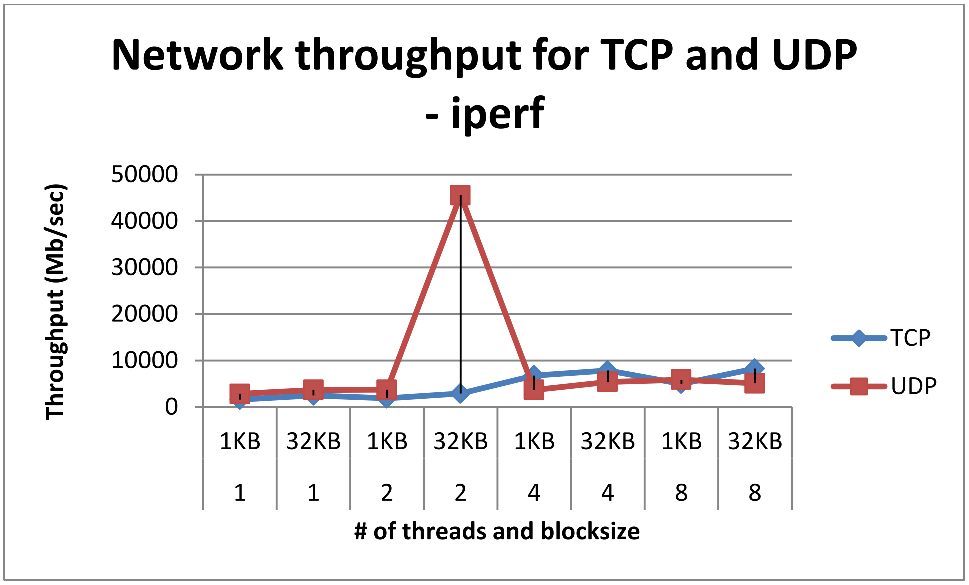
Network Throughput					
Protocol	Con-currency	Block Size	iperf Measured Throughput(Mb/sec)	Theoretical Throughput (Mb/sec)	iperf Efficiency (%)
TCP	1	1KB	1564.32	56000	2.793428571
TCP	1	32KB	2445.56	56000	4.367071429
TCP	2	1KB	1834.45	56000	3.275803571
TCP	2	32KB	2877.98	56000	5.13925
TCP	4	1KB	6763.34	56000	12.07739286
TCP	4	32KB	7839.67	56000	13.99941071
TCP	8	1KB	4978.34	56000	8.889892857
TCP	8	32KB	8234.09	56000	14.70373214
UDP	1	1KB	2784.76	56000	4.972785714
UDP	1	32KB	3673.79	56000	6.560339286
UDP	2	1KB	3682.57	56000	6.576017857
UDP	2	32KB	45567.34	56000	81.37025
UDP	4	1KB	3674.92	56000	6.562357143
UDP	4	32KB	5376.45	56000	9.600803571
UDP	8	1KB	5863.57	56000	10.47066071
UDP	8	32KB	5063.95	56000	9.042767857

Table 7 : NETWORK THROUGHPUT

- **Network Latency :** The following table shows the ping measured latency in ms for TCP and UDP protocols

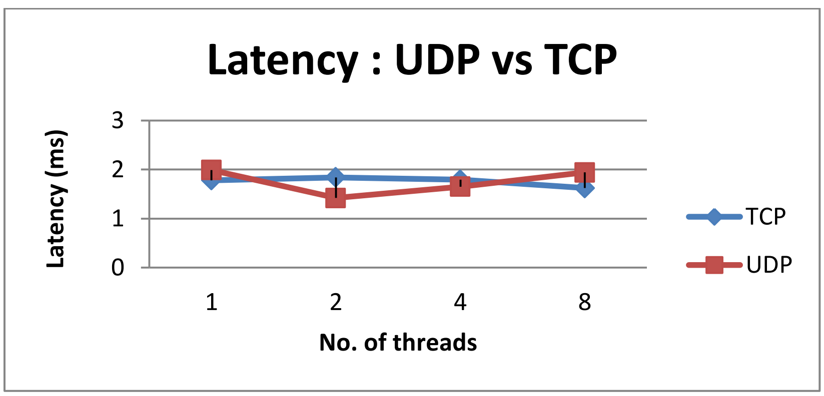
Network Latency					
Protocol	Con-currency	Message Size	ping Measured Latency (ms)	Theoretical Latency (ms)	iperf Efficicency(%)
TCP	1	1B	1.78	0.0007	254285.7143
TCP	2	1B	1.84	0.0007	262857.1429
TCP	4	1B	1.79	0.0007	255714.2857
TCP	8	1B	1.62	0.0007	231428.5714
UDP	1	1B	1.99	0.0007	284285.7143
UDP	2	1B	1.42	0.0007	202857.1429
UDP	4	1B	1.65	0.0007	235714.2857
UDP	8	1B	1.94	0.0007	277142.8571

Table 8: NETWORK LATENCY



Graph 21

- **Analysis for Graph 21:** The above chart depicts the variance of throughput with respect to the number of threads and block size. For UDP 2 threads 32 KB the throughput reached peak



Graph 22

- **Analysis for Graph 22:** This chart shows the latency for TCP and UDP for 1B block size. As seen above, the network latency for both the protocols varied slightly with the increase in the number of threads.