3. Upgrade your performance measurement setup now to measure **CPU utilization** and **Average number of active threads**.

a. For each load level, just before you start the experiment, start taking 10 second snapshots of CPU utilization using ***vmstat***, then stop after the experiment. (Needless to say, save this to a file.) Similarly use **ps eLf** or any other tool you like to take snapshots of **NLWP**

No. of clients:- 10

```
vedant@laptop:~/Practicals/DECS/lab_08$ vmstat 10
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 0  0 339736 2083656  92016 1381420    0    4    38    84   88   74 12  5 82  1  0
 1  0 339736 2087688  92048 1381516    0    0     0    78 3886 1607 29  5 66  0  0
 0  0 339736 2086136  92076 1381516    0    0     0    66 2440  867  1  1 98  0  0
```

```
UID        PID        PPID     LWP     C  NLWP STIME TTY          TIME CMD
vedant@laptop:~/Practicals/DECS/lab_08$ ps -eLf | grep gradingserver
vedant    282560  282553  282560  0   11 18:36 pts/1    00:00:00 ./gradingserver 8000
vedant    282560  282553  294231  0   11 19:17 pts/1    00:00:00 ./gradingserver 8000
vedant    282560  282553  294232  0   11 19:17 pts/1    00:00:00 ./gradingserver 8000
vedant    282560  282553  294233  0   11 19:17 pts/1    00:00:00 ./gradingserver 8000
vedant    282560  282553  294234  0   11 19:17 pts/1    00:00:00 ./gradingserver 8000
vedant    282560  282553  294238  0   11 19:17 pts/1    00:00:00 ./gradingserver 8000
vedant    282560  282553  294241  0   11 19:17 pts/1    00:00:00 ./gradingserver 8000
vedant    282560  282553  294247  0   11 19:17 pts/1    00:00:00 ./gradingserver 8000
vedant    282560  282553  294249  0   11 19:17 pts/1    00:00:00 ./gradingserver 8000
vedant    282560  282553  294251  0   11 19:17 pts/1    00:00:00 ./gradingserver 8000
vedant    282560  282553  294256  0   11 19:17 pts/1    00:00:00 ./gradingserver 8000
```

No. of clients:- 20

```
vedant@laptop:~/Practicals/DECS/lab_08$ vmstat 10
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 0  0 339736 2069004  92464 1388912    0    4    38    83   89   75 12  5 82  1  0
 0  0 339736 2072392  92472 1389532    0    0     0    80 3824 1712 60  8 32  0  0
 0  0 339736 2072356  92496 1388896    0    0     0    66  858  924  1  2 97  0  0
```

```
vedant@laptop:~/Practicals/DECS/lab_08$ ps -eLf | grep gradingserver
vedant    282560  282553  282560  0   21 18:36 pts/1    00:00:00 ./gradingserver 8000
vedant    282560  282553  294868  0   21 19:20 pts/1    00:00:00 ./gradingserver 8000
vedant    282560  282553  294870  0   21 19:20 pts/1    00:00:00 ./gradingserver 8000
vedant    282560  282553  294871  0   21 19:20 pts/1    00:00:00 ./gradingserver 8000
vedant    282560  282553  294877  0   21 19:20 pts/1    00:00:00 ./gradingserver 8000
vedant    282560  282553  294886  0   21 19:20 pts/1    00:00:00 ./gradingserver 8000
vedant    282560  282553  294887  0   21 19:20 pts/1    00:00:00 ./gradingserver 8000
vedant    282560  282553  294891  0   21 19:20 pts/1    00:00:00 ./gradingserver 8000
vedant    282560  282553  294892  0   21 19:20 pts/1    00:00:00 ./gradingserver 8000
vedant    282560  282553  294893  0   21 19:20 pts/1    00:00:00 ./gradingserver 8000
vedant    282560  282553  294896  0   21 19:20 pts/1    00:00:00 ./gradingserver 8000
vedant    282560  282553  294897  0   21 19:20 pts/1    00:00:00 ./gradingserver 8000
vedant    282560  282553  294901  0   21 19:20 pts/1    00:00:00 ./gradingserver 8000
vedant    282560  282553  294902  0   21 19:20 pts/1    00:00:00 ./gradingserver 8000
vedant    282560  282553  294914  0   21 19:20 pts/1    00:00:00 ./gradingserver 8000
vedant    282560  282553  294916  0   21 19:20 pts/1    00:00:00 ./gradingserver 8000
vedant    282560  282553  294917  0   21 19:20 pts/1    00:00:00 ./gradingserver 8000
vedant    282560  282553  294934  0   21 19:20 pts/1    00:00:00 ./gradingserver 8000
vedant    282560  282553  294935  0   21 19:20 pts/1    00:00:00 ./gradingserver 8000
vedant    282560  282553  294936  0   21 19:20 pts/1    00:00:00 ./gradingserver 8000
vedant    282560  282553  294955  0   21 19:20 pts/1    00:00:00 ./gradingserver 8000
```

No. of clients:- 30

```
vedant@laptop:~/Practicals/DECS/lab_08$ vmstat 10
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 0  0 339736 2050380  92628 1393268    0    4    38    83   90   75 12  5 82  1  0
 0  0 339736 2059112  92652 1393348    0    0     0    14 1858  818  1  1 98  0  0
30  0 339736 1605740  92692 1394544    0    0     1   195 4218 2226 73 10 17  0  0
 0  0 339736 2113856  92728 1384652    0    0     0   192 1960 1648 15  4 81  0  0
```

```
vedant@laptop:~/Practicals/DECS/lab_08$ ps -eLf | grep gradingserver
vedant    282560  282553  282560  0   31 18:36 pts/1    00:00:00 ./gradingserver 8000
vedant    282560  282553  296063  0   31 19:21 pts/1    00:00:00 ./gradingserver 8000
vedant    282560  282553  296064  0   31 19:21 pts/1    00:00:00 ./gradingserver 8000
vedant    282560  282553  296066  0   31 19:21 pts/1    00:00:00 ./gradingserver 8000
```

No. of clients:- 40

```
vedant@laptop:~/Practicals/DECS/lab_08$ vmstat 10
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 0  0 339736 2103352  92860 1380776    0    4    38    83   90   75 12  5 82  1  0
 0  0 339736 2098184  92972 1387232    0    0     0    60 3610 1373  1  1 96  2  0
41  0 339736 1298480  93036 1388532    0    0     0   215 2852 2109 69 10 21  0  0
 1  0 339736 2114104  93056 1384980    0    0     0   131 1890 2055 45  8 47  0  0
 0  0 339736 2109036  93080 1384972    0    0     0    75  526  447  0  0 99  0  0
```

```
vedant@laptop:~/Practicals/DECS/lab_08$ ps -eLf | grep gradingserver
vedant     282560  282553  282560  0   41 18:36 pts/1    00:00:00 ./gradingserver 8000
vedant     282560  282553  297830  0   41 19:21 pts/1    00:00:00 ./gradingserver 8000
vedant     282560  282553  297831  0   41 19:21 pts/1    00:00:00 ./gradingserver 8000
vedant     282560  282553  297833  0   41 19:21 pts/1    00:00:00 ./gradingserver 8000
vedant     282560  282553  297838  0   41 19:21 pts/1    00:00:00 ./gradingserver 8000
vedant     282560  282553  297841  0   41 19:21 pts/1    00:00:00 ./gradingserver 8000
vedant     282560  282553  297843  0   41 19:21 pts/1    00:00:00 ./gradingserver 8000
vedant     282560  282553  297853  0   41 19:21 pts/1    00:00:00 ./gradingserver 8000
vedant     282560  282553  297854  0   41 19:21 pts/1    00:00:00 ./gradingserver 8000
```

No. of clients:- 45

```
vedant@laptop:~/Practicals/DECS/lab_08$ vmstat 10
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 0  0 339736 2091552  93308 1387552    0    4    38    83   91   75 12  5 82  1  0
 1  0 339736 2090320  93332 1385376    0    0     0    31 3035 1253  1  1 98  0  0
45  0 339736 1416916  93364 1387276    0    0     0   184 2713 2198 70 10 19  0  0
 0  0 339736 2077368  93408 1387024    0    0   142  1210 2289 2571 59  9 31  0  0
 0  0 339736 2075704  93424 1387072    0    0     5    94  550  477  1  0 99  0  0
```

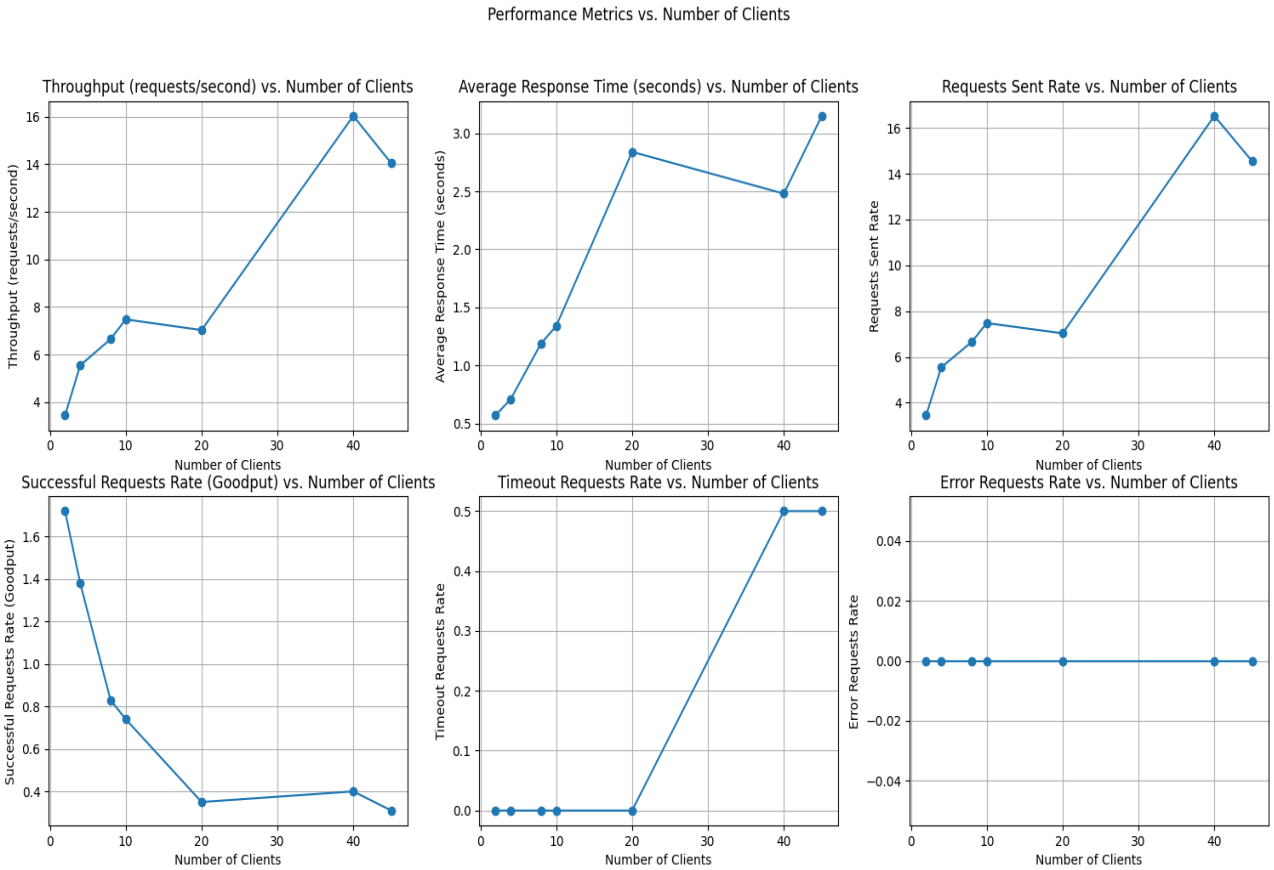
```
vedant@laptop:~/Practicals/DECS/lab_08$ ps -eLf | grep gradingserver
vedant     282560  282553  282560  0   46 18:36 pts/1    00:00:00 ./gradingserver 8000
vedant     282560  282553  300169  0   46 19:22 pts/1    00:00:00 ./gradingserver 8000
vedant     282560  282553  300170  0   46 19:22 pts/1    00:00:00 ./gradingserver 8000
vedant     282560  282553  300172  0   46 19:22 pts/1    00:00:00 ./gradingserver 8000
vedant     282560  282553  300173  0   46 19:22 pts/1    00:00:00 ./gradingserver 8000
vedant     282560  282553  300174  0   46 19:22 pts/1    00:00:00 ./gradingserver 8000
vedant     282560  282553  300184  0   46 19:22 pts/1    00:00:00 ./gradingserver 8000
vedant     282560  282553  300187  0   46 19:22 pts/1    00:00:00 ./gradingserver 8000
vedant     282560  282553  300188  0   46 19:22 pts/1    00:00:00 ./gradingserver 8000
vedant     282560  282553  300192  0   46 19:22 pts/1    00:00:00 ./gradingserver 8000
vedant     282560  282553  300193  0   46 19:22 pts/1    00:00:00 ./gradingserver 8000
vedant     282560  282553  300202  0   46 19:22 pts/1    00:00:00 ./gradingserver 8000
vedant     282560  282553  300205  0   46 19:22 pts/1    00:00:00 ./gradingserver 8000
vedant     282560  282553  300206  0   46 19:22 pts/1    00:00:00 ./gradingserver 8000
vedant     282560  282553  300208  0   46 19:22 pts/1    00:00:00 ./gradingserver 8000
vedant     282560  282553  300220  0   46 19:22 pts/1    00:00:00 ./gradingserver 8000
vedant     282560  282553  300223  0   46 19:22 pts/1    00:00:00 ./gradingserver 8000
vedant     282560  282553  300224  0   46 19:22 pts/1    00:00:00 ./gradingserver 8000
```

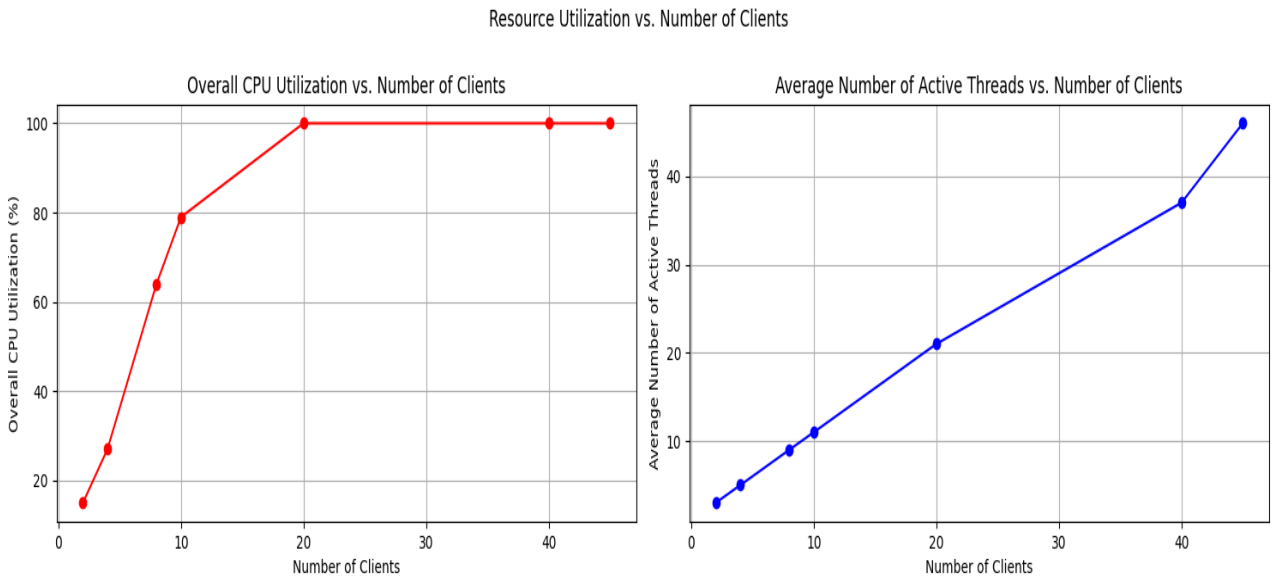
b. Calculate the average CPU utilization and average number of active threads (ensure you do not use ‘warm up’ and ‘cool down’ values).

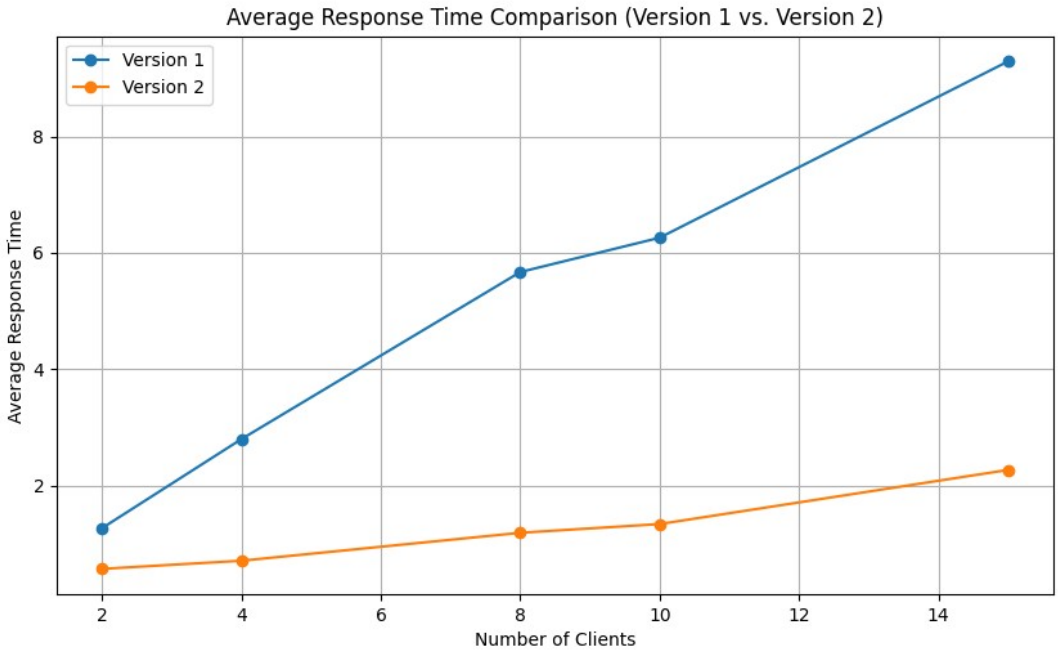
Num Clients	Throughput	Avg Response Time	Requests Sent Rate	Successful Requests Rate	Timeout Requests Rate	Error Requests Rate	Average Active Threads	Overall CPU Utilization
2	3.45	0.57	3.45	1.72	0	0	3	15
4	5.55	0.71	5.55	1.38	0	0	5	27
8	6.65	1.19	6.65	0.83	0	0	9	64
10	7.48	1.34	7.48	0.74	0	0	11	79
20	7.03	2.84	7.03	0.35	0	0	21	100
40	16.03	2.48	16.53	0.4	0.5	0	37	100
45	14.04	3.15	14.54	0.31	0.5	0	46	100

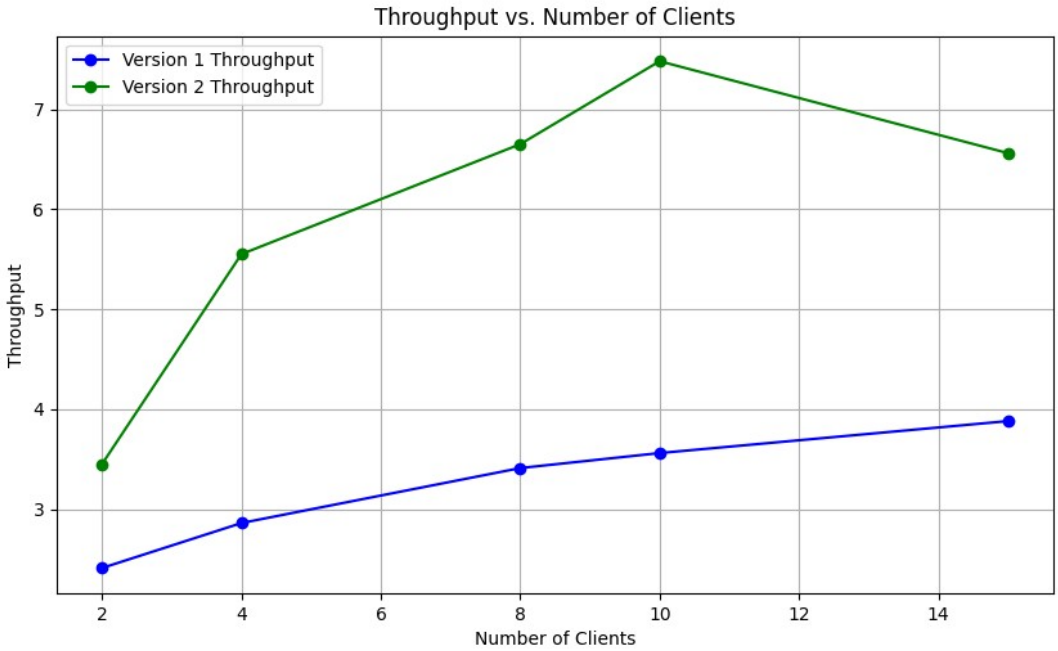
4. Now do the same performance experiments as before. First do these experiments by keeping the number of cores small (2 cores). Ensure the load is such that you are able to drive the CPU to 100%.

1. Now for each load level (i.e. number of clients), apart from throughput and response time, also calculate request sent rate, and timeout rate (so request sent should be = throughput + timeout + error rate if any)
2. Do the experiments for various load levels and plot these curves vs M:
    1. Request rate sent
    2. Successful request rate (we will now call this **goodput)**
    3. Timeout rate
    4. Error rate if any (connection refused etc)
    5. CPU utilization
    6. Average number of active threads
3. Compare the performance metrics with the Version 1 metrics (plot them on the same graph)




Resource Utilization vs. Number of Clients
Overall CPU Utilization vs. Number of Clients
Overall CPU Utilization (%)
Number of Clients
Average Number of Active Threads vs. Number of Clients
Average Number of Active Threads
Number of Clients


Average Response Time Comparison (Version 1 vs. Version 2)
Version 1
Version 2
Average Response Time
Number of Clients


Throughput vs. Number of Clients
Version 1 Throughput
Version 2 Throughput
Throughput
Number of Clients
2
4
6
8
10
12
14
3
4
5
6
7