OS LAB 2

Karan Vaidya 130050019

Sivaprasad S 130050085

* Common reasons to observations in 2,3 and 4 are mentioned at the end under conclusion

1.

Client and server were on two machines connected to a router by Ethernet cable (Router is acting as switch).

Read bandwidth of the disk at the server : 28MBps = 14req/s (./disk saturates at this value)

Commands used :

server: iostat -dx 1

Network Bandwidth : 94.1 Mbps = 11.76 MBps = 5.8825 req/sec

Commands used :

server : iperf -s -i 100

client : iperf -n 1024M -i 100 -c <ip address>

Justification : The LAN / router is rated 100 Mbps and 94.1 Mbps is practically achievable value.

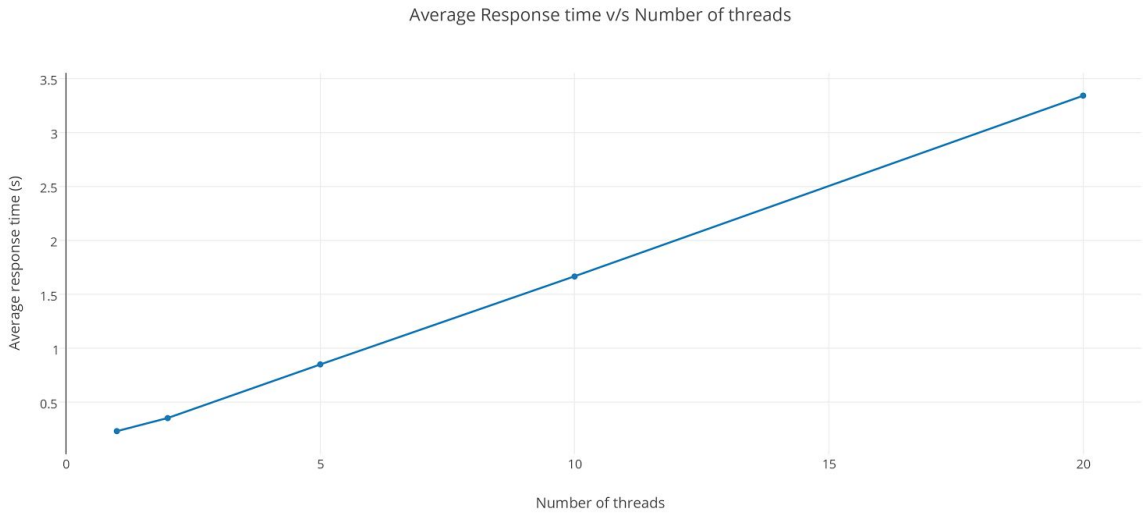
****

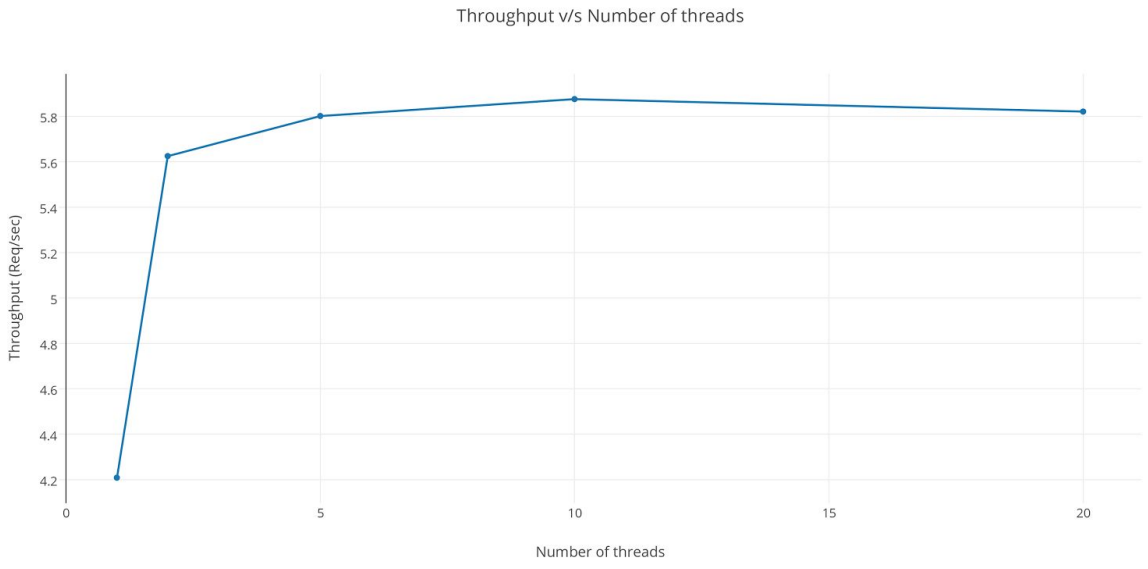
2.

a. N=5

b. The throughput is lower for number of threads lesser than n. It almost remains the same for higher values.

   The response time increases with increase in number of threads.





c. The bottleneck resource was identified to be Network bandwidth. The Interface utilisation obtained from sar -n DEV 1 5 is 98.8% for eth0 interface. The cpu usage disk usage at the server are low. The bandwidth reaches a value of 5.87 req/s

which is 99.8% utilisation of the bandwidth obtained from iperf command. Thus bandwidth is identified as the bottleneck resource.

d. The Maximum bandwidth received is 5.876033 req/sec. The value in turns out to be 94.016528 Mbps which is approximately equal to the bandwidth of 94.1(99.9%) we got using iperf.

n	Throughput (in req/s)	avg_resp_time
1	4.208333	0.231683
2	5.625000	0.352593
5	5.801653	0.850427
10	5.876033	1.665260
20	5.821138	3.342179

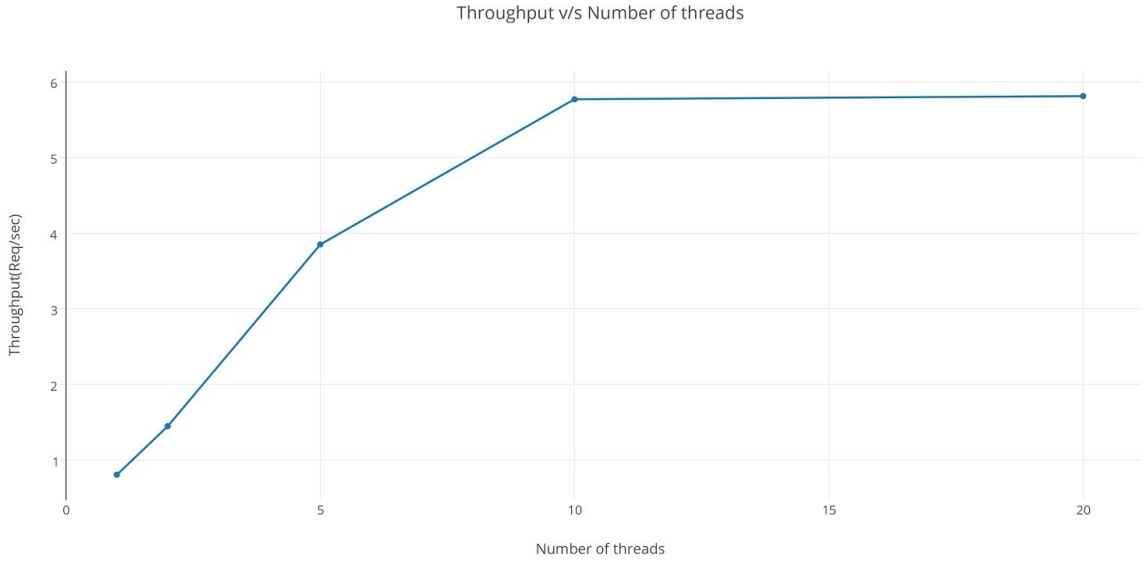
****

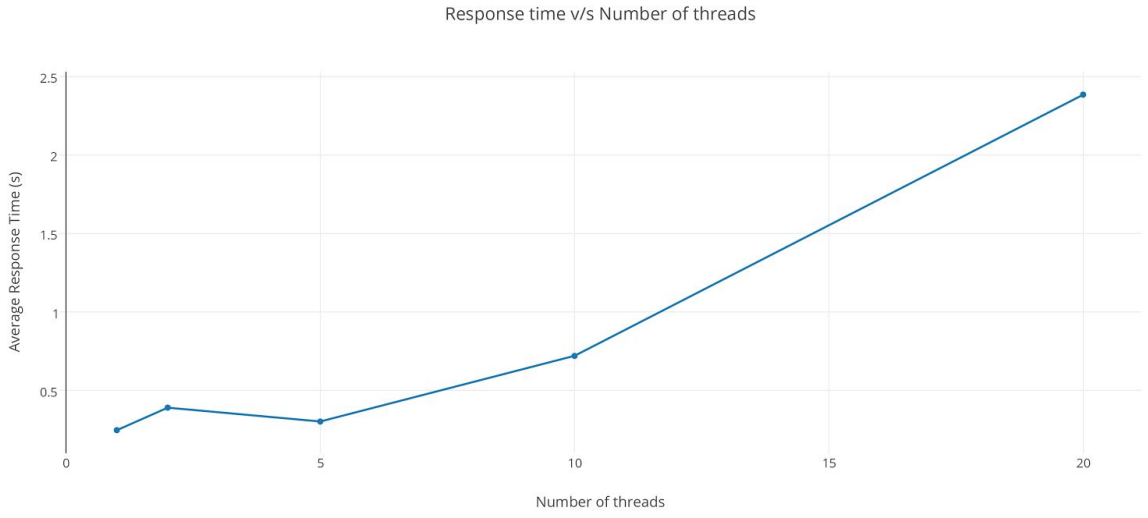
3.

a. N=10

b. The throughput is lower for number of threads lesser than n. It almost remains the same for higher values.

The response time increases with increase in number of threads.





n	Throughput (in req/s)	avg_resp_time
1	0.808333	0.247423
2	1.450000	0.390805
5	3.852459	0.302128
10	5.770492	0.720170
20	5.813008	2.384615

c. The bottleneck resource was identified to be Network bandwidth. The Interface utilisation obtained from sar -n DEV 1 5 is 98.55% for eth0 interface. The cpu usage disk usage at the server are low. The bandwidth reaches a value of 5.8 req/s which is 98.6% utilisation of the bandwidth obtained from iperf command. Thus bandwidth is identified as the bottleneck resource.

d. The Maximum bandwidth received is 5.876033 req/sec. The value in turns out to be 94.016528 Mbps which is approximately equal to the bandwidth of 94.1(99.9%) we got using iperf.

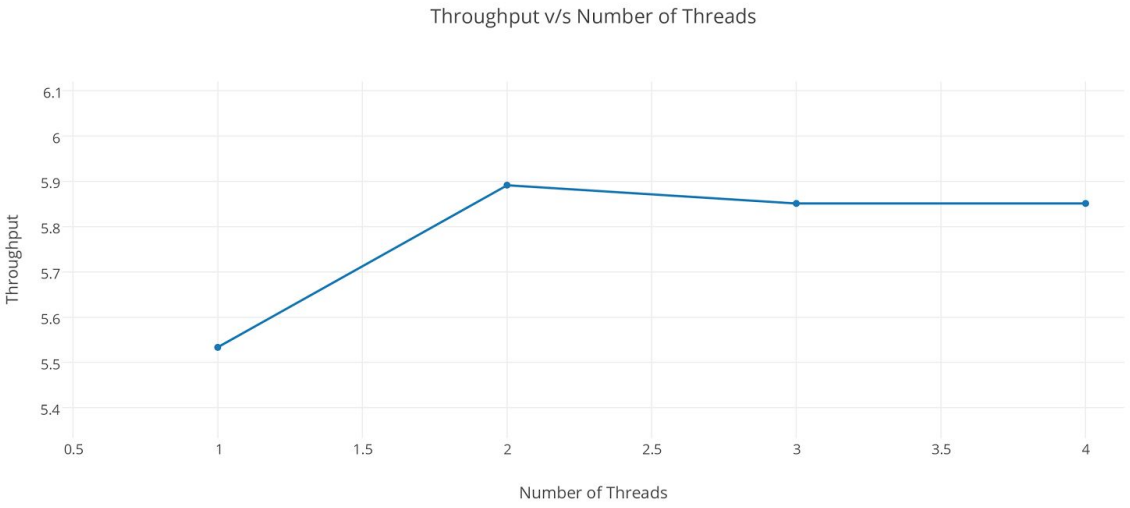
****

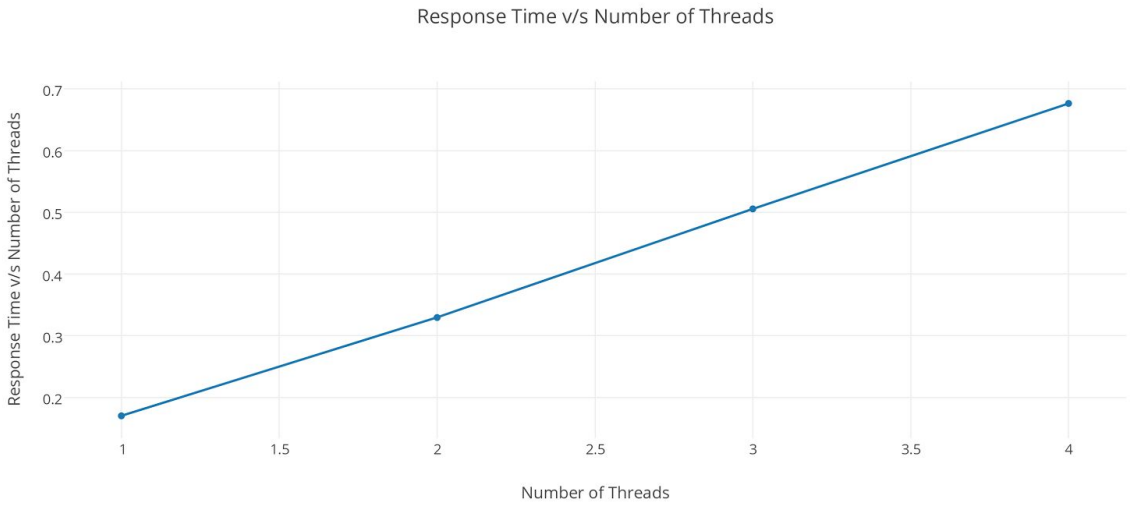
4.

a. N=2

b. The throughput is lower for number of threads lesser than n. It almost remains the same for higher values.

   The response time increases with increase in number of threads.





n	Throughput (in req/s)	avg_resp_time
1	5.533333	0.170181
2	5.891667	0.329562
3	5.851240	0.505650
4	5.851240	0.676554

c. The bottleneck resource was identified to be Network bandwidth. The Interface utilisation obtained from sar -n DEV 1 5 is 98.5% for eth0 interface. The cpu usage disk usage at the server are low. The bandwidth reaches a value of 5.89 req/s which is almost 100% utilisation of the bandwidth obtained from iperf command. Thus bandwidth is identified as the bottleneck resource.

d. The Maximum bandwidth received is 5.891667 req/sec. The value in turns out to be 94.26662 Mbps which is approximately equal to the bandwidth of 94.1, we got using iperf.

****

Conclusion:-

When there is only one user, the user uses only a share of the link bandwidth. As the number of threads increases the bandwidth utilisation increases and saturates at near 100%. Also concurrency increases the effective utilisation of bandwidth.

Response time increases with number of users as the process has to wait for more processes to be scheduled by the CPU scheduler.

The throughput increases with number of users (load) till the Network bandwidth bottleneck is achieved and then remains constant. Comparing the three cases, saturation is achieved at lesser users in fixed mode (2) than random mode(5). It is because in fixed mode the same file is requested for by each user and hence will be cached and increases the throughput. Comparing 0 think time and that of 1s, throughput is achieved later in latter as the active number of users at any time is lesser than that in former.

Response time with think time is lesser than that without think time as the number of active users at a time is lesser in former and concurrency gives better response time. Response time in random mode is better than fixed mode as the file will be cached and there is no delay in reading from the disk.

****
**