HOMEWORK 4 PART –A

Jeshalraj R. Thakaria

UFID: 83213912

1) The reason performance did not increase as expected maybe because transit time was not the bottleneck in the system. It might not be the most significant part of the latency in the system at the first place. There are many steps in RPC implementation like marshalling, service processing, system's dependency on server's response etc. It is possible that any of these steps maybe the actual bottleneck. For e.g. marshalling in the implementation maybe the slowest step in the chain and hence the performance would be largely governed by it or the implementation of RPC would be such that a client cannot send another request until it receives the acknowledgment or response from the server for previous request or the processor on the server side might not be that fast for processing the requests. Also when transit time is decreased to half, there are more chances of the system actually queuing the requests as the arrival rate has increased. The throughput in that case would also depend on what kind of scheduling is implemented on the system to process all new threads for e.g. shortest job first may work better than first come first serve type of scheduling in case of small RPC requests which is the case for which the system is tested for.
2) Page size increase performance:
    - More page size would mean lesser page tables which would occupy less memory space.
    - Larger page size would mean a large amount of data and instructions would be in a sequentially allotted in physical memory, which would help retrieving data form disk easier as rotational latency of disk would be reduced.
    - Also if a TLB cache is used for faster references to memory, larger page size would enable a TLB of given size to address to a larger portion of physical memory.

    Page size decrease performance:

    - Larger page size would mean more data will be needed to be transferred to main memory from disk on a page miss.
    - Larger page size would decrease memory utilization as smaller programs or threads would not require huge memories and most of the page would remain empty and useless.
    - If the size of the main memory is kept the same and the page size is increased, now we can keep lesser number of pages in the memory which would increase the page miss rate.
3) Data transfer rate = 120 tracks/sec = 120*1.5 MBps = 180MBps

    i) Average latency random sector read Ls= average seek time + average rotational time + average transfer time

Average rotational time = 0.5* 1/ (7200/60) =0.5*8.33ms = 4.167ms

Ls = 8ms + 4.167ms + (4/(1024*180*1000)) = 12.19ms

ii) Average throughput = (1 / average read latency of a sector)* 4KB = (1/ 12.19ms)*4 = 328.13 KB/sec

iii) Average latency for a read of entire track Lt = Average seek time + time for one full rotation

Lt = 8ms + 8.33ms = 16.33ms

iv) Average throughput for read of entire track = (1/ average read latency of track)* 1.5MB
= (1/16.33ms)*1.5 = 91.85 MB/sec

v) Utilization = Arrival rate * Service time(Average latency for reading a sector)
0.8 = Arrival rate * 12.19ms
Arrival rate = 0.8/12.19ms
= 0.0656 requests/ms = 65.6 requests/sec

vi) Average time in queue = service time * utilization/ (1-utilizaiton)
= 12.19ms * 0.8/0.2
= 48.76ms

Length of Queue = time in queue * arrival rate = 48.76 *0.0656 = 3.2 requests