Jordan Wu

U0900517

Homework Assignment 11

1. **Consider the following specifications for two hard drives:**

Drive A:

- **Drive diameter: 3.5"**
- **Platters: 3**
- **RPM: 7,500**
- **Average time needed to move the read heads to a specific track (seek time): 10ms**
- **Rate at which data passes under the read heads: 90 MB/s**
- **Rate at which the hard drive communicates with the computer: 100 MB/s**

Drive B:

- **Drive diameter: 2.5"**
- **Platters: 1**
- **RPM: 10,000**
- **Average time needed to move the read heads to a specific track (seek time): 7ms**
- **Rate at which data passes under the read heads: 40 MB/s**
- **Rate at which the hard drive communicates with the computer: 200 MB/s**

Assume that the OS requests a read of some block of data from one track on a drive. Assume the drive controller will read the entire block of data from the disk, then begin transmitting the data back to the OS. Answer the following questions twice, once for drive A and once for drive B.

(a) What is the average latency for reading a 1024 byte block of data?

The disk memory consist of a collection of platters which are spinning at a specific RPM. To read and write information on a hard disk, a read-write head is used. Each disk surface is divided into concentric circles called tracks and each track has thousands of sectors that contain information. The sectors at the outer tracks can a bigger bytes size. To access data there are three-stage processes.

First step: seek: the time to move the read-write head to the desired track also known as seek time.

Second Step: Next wait for the desired sector to rotate under the read-write head this is also known as rotational latency or rotational delay.

Last step: The last thing is the transfer time, the time to transfer a block of bits. This is a function of sector size, the rotation speed, and the recording density of a track.

$$latency = seek\ time + rotational\ delay + transfer\ time + controller\ time$$

$$RPM = \frac{rotation}{minute} \cdot \frac{minute}{60\ seconds} = Rotation\ per\ second$$

$$Avg.rotational\ latency\ for\ Drive\ A = \frac{0.5\ rotation}{7{,}500\ RPM/(60\frac{seconds}{minute})} = 4\ ms$$

$$Avg.rotational\ latency\ for\ Drive\ B = \frac{0.5\ rotation}{10{,}000\ RPM/(60\frac{seconds}{minute})} = 3\ ms$$

$$transfer\ time\ for\ Drive\ A = \frac{1024\ bytes\ block\ data}{90{,}000{,}000\ \frac{bytes}{second}} = 11.377\ \mu s$$

$$transfer\ time\ for\ Drive\ B = \frac{1024\ bytes\ block\ data}{40{,}000{,}000\ \frac{bytes}{second}} = 25.6\ \mu s$$

$$controller\ time\ for\ Drive\ A = \frac{1024\ bytes\ block\ data}{100{,}000{,}000\ \frac{bytes}{second}} = 10.24\ \mu s$$

$$controller\ time\ for\ Drive\ B = \frac{1024\ bytes\ block\ data}{200{,}000{,}000\ \frac{bytes}{second}} = 5.12\ \mu s$$

$$Avg\ latency\ for\ Drive\ A = 10ms + 4ms + 11.377\mu s + 10.24\mu s = 14.021617\ ms$$

$$Avg\ latency\ for\ Drive\ B = 7ms + 3ms + 25.6\mu s + 5.12\mu s = 10.03072\ ms$$

(b) What is the minimum latency for reading a 2048 byte block of data?

There will be no seek time and rotational delay since this is looking for the minimum latency where the read-write head is already at the correct sector.

$$minimum\ latency = transfer\ time + controller\ time$$

$$transfer\ time\ for\ Drive\ A = \frac{2048\ bytes\ block\ data}{90{,}000{,}000\ \frac{bytes}{second}} = 22.755\ \mu s$$

$$transfer\ time\ for\ Drive\ B = \frac{2048\ bytes\ block\ data}{40{,}000{,}000\ \frac{bytes}{second}} = 51.2\ \mu s$$

$$controller\ time\ for\ Drive\ A = \frac{2048\ bytes\ block\ data}{100{,}000{,}000\ \frac{bytes}{second}} = 20.48\ \mu s$$

$$controller\ time\ for\ Drive\ B = \frac{2048\ bytes\ block\ data}{200{,}000{,}000\ \frac{bytes}{second}} = 10.24\ \mu s$$

$$minimum\ latency\ for\ Drive\ A = 22.755\ \mu s + 20.48\ \mu s = 43.235\ \mu s$$

$$minimum\ latency\ for\ Drive\ B = 51.2\ \mu s + 10.24\ \mu s = 61.62\ \mu s$$

(c) For each drive, determine the dominant factor that affects latency. Specifically, if you could make an improvement to any one aspect of that drive, what would you improve and why? If there is no dominant factor, explain why.

If my calculation are correct, the seek time and rotational delay play a bigger role in latency than the transfer time and controller time. The average seek time depends on the diameter of the disk, where having a bigger drive diameter increases the average seek time. Having a bigger drive diameter seem to increase the capacity of memory where more data can be stored. Rotational delay seem to be decrease as the RPM is increased, but having a higher RPM might increase power consumption. I would say that if memory size is not an issue than decrease the drive diameter but if power consumption is not an issue than increase the RPM. My choice would be to improve the RPM since 25-35 percent of the time the disk request are referencing to locality data which will decrease the seek time.

2. **Solid state drives also have a controller, but the controller reads data out of flash ram chips prior to sending it back to the OS. Assume these data rates for two flash drives:**

Drive A:

- **Drive width: 2.5"**
- **Rate at which data can be read from flash memory: 600 MB/s**
- **Rate at which the drive communicates with the computer: 300 MB/s**

Drive B:

- **Drive width: 2.5"**
- **Rate at which data can be read from flash memory: 100 MB/s**
- **Rate at which the drive communicates with the computer: 90 MB/s**

Answer these for both drives:

(a) What is the average latency for reading a 1024 byte block of data?

The solid state drives do not have any mechanical components, this mean there will be no rotational time delay. Also using the

$$latency = access\ time + transfer\ time + controller\ time$$

$$transfer\ time\ for\ Drive\ A = \frac{1024\ byte\ block\ of\ data}{600{,}000{,}000\,\frac{bytes}{second}} = 1.706\ \mu s$$

$$transfer\ time\ for\ Drive\ B = \frac{1024\ byte\ block\ of\ data}{100{,}000{,}000\,\frac{bytes}{second}} = 10.24\ \mu s$$

$$controller\ time\ for\ Drive\ A = \frac{1024\ bytes\ block\ data}{300{,}000{,}000\ \frac{bytes}{second}} = 3.413\ \mu s$$

$$controller\ time\ for\ Drive\ B = \frac{1024\ bytes\ block\ data}{90{,}000{,}000\ \frac{bytes}{second}} = 11.37\ \mu s$$

$$latency\ for\ Drive\ A = 0.1\ ms + 1.706\ \mu s + 3.413\ \mu s = 105.119\ \mu s$$

$$latency\ for\ Drive\ B = 0.1\ ms + 10.24\ \mu s + 11.37\ \mu s = 121.61\ \mu s$$

(b) What is the minimum latency for reading a 2048 byte block of data?

The only difference is that the byte block of data is doubled and is the same as multiplying by 2.

$$latency\ for\ Drive\ A = 0.1\ ms + 2 \cdot 1.706\ \mu s + 2 \cdot 3.413\ \mu s = 110.24\ \mu s$$

$$latency\ for\ Drive\ B = 0.1\ ms + 2 \cdot 10.24\ \mu s + 2 \cdot 11.37\ \mu s = 143.22\ \mu s$$

Then answer this general question:

(c) Would you expect that when reading flash ram, latency to access flash ram would increase as memory gets larger? If so, why? If not, why not? Is this significant? (Full credit for a well-reasoned discussion that relates latency, throughput, and the size of flash memory.)

It would not make any sense if the latency to access flash ram increases as memory gets larger. The reason why this is illogical is that it should take longer to find the data when the memory size is large. The size of the flash memory might not significantly increase the latency too much since it can find the data pretty quick. The throughput of the flash ram shouldn't change too much if the size of the flash is increase since it does not significantly increase the latency. Conclusion, the latency of the flash ram will not increase significantly when the size is increase but the cost of it might increase significantly.

3. (a) Assume you have a 25 meter network cable and that you are transmitting bits across the cable at a rate of 100 Mhz. How many bits are "in-flight", or in transit down the cable at any particular moment?

To solve this problem we have to know how fast the signal is traveling down this cable.

Assume that electrical signals travel at 2*10^8 m/s in the copper wire.

The data transmit rate has a frequency of 100 Mhz which we can find the period of it using the reciprocal. The period would be 10 ns, **this means that the cable will receive a bit every 10 ns.** Now we have to find the time it takes the electrical signal to travel 25 meters.

$$\textit{time it takes signal to travel } 25 \textit{ meters} = \frac{25 \textit{ meters}}{2 \cdot 10^8 \frac{\textit{meters}}{\textit{seconds}}} = 125 \, ns$$

Now we can divide the time it takes the signal to travel 25 meters with 10 ns. This will give use the bits in the cable at any particular time which is 125/10 = 12.5 which is around 12 bits.

(b) Now assume that you have a 2,500 meter optical communications cable and that you are transmitting bits across the cable at a rate of 10 Ghz. How many bits are "in-flight" at any particular moment?

Optical communication uses light to carry information down the cable so in this problem the speed of the signal is the speed of light which is 3*10^8 m/s. Now changing the rate of the transmitting bits by doing the reciprocal, the cable will receive a bit every 1 ns. Now calculating the time it takes the signal to travel down this 2,500 meter cable.

$$\textit{time it takes signal to travel } 25 \textit{ meters} = \frac{2{,}500 \textit{ meters}}{3 \cdot 10^8 \frac{\textit{meters}}{\textit{seconds}}} = 8.\overline{33} \, \mu s$$

The bits in-flight will be $8.\overline{33} \, \mu s/$ 1 ns = 8333.33 which is about 8333 bits. MADE THE WRONG ASSUMPTION, NOW USING THE ASUMPTION BELOW.

Assume that optical signals travel at 2*10^8 m/s in the fiber.

$$\textit{time it takes signal to travel } 25 \textit{ meters} = \frac{2{,}500 \textit{ meters}}{2 \cdot 10^8 \frac{\textit{meters}}{\textit{seconds}}} = 12.5 \, \mu s$$

The bits in-flight will be $12.5 \, \mu s/$ 1 ns = 12500 bits.

(c) For both situations, what would be the expected latency for requesting and receiving a block of data? Assume the client on one end of the cable requests (with a 100 byte request message) a 100,000 byte data block from a server on the other end, and that it takes an average of 10 signaled bits to transmit each byte of data. Also assume that in addition to wire delays, the communications overhead is 0.02 ms on each end for each read or write of a message (or data block). Compute the total time from the start of the 100 byte request to the receiving the end byte of the 100,000 byte data packet for both cables from part (a) and (b).

Looking at this problem the client will receiving 100,000 byte data block from a server and it takes an average of 10 signal bits to transmit each byte of data. This signal bits will include the overhead of 2 bits. To setup this problem I would look at the request message of 100 bytes and the receiving block of data of 100,000 byte from the server.

Starting with part a)

The rate at which it is transmitting bits is around one bit per 10 ns. We need to send 100 byte which is 1000 bits (including the overhead which is 10 bits per byte), we would need 1000 bits * 10 ns = 10,000 ns to send 100 bytes to the server. Adding the communication overhead cause by wire delays to this we would get a latency for the request to be 30 μs. Now receiving the data from the server will take 1,000,000 bits times the 10 ns will give us 10,000,000 ns plus the 0.02 ms overhead will give us 10.02 ms. Now adding them together will give use the latency of 10.05 ms for part a.

Part b)

The rate at which it is transmitting bits is around one bit per 1 ns. We need to send 100 byte which is 1000 bits (including the overhead which is 10 bits per byte), we would need 1000 bits * 1 ns = 1,000 ns to send 100 bytes to the server. Adding the communication overhead cause by wire delays to this we would get a latency for the request to be 21 μs. Now receiving the data from the server will take 1,000,000 bits times the 1 ns will give us 1,000,000 ns plus the 0.02 ms overhead will give us 1.02 ms. Now adding them together will give use the latency of 1.041 ms for part b.

(d) From (c), what term dominates the latency in each situation? In other words, in each situation what could you improve to best decrease the latency of the data block request?

The dominate factor is the data rate in which bits are transmitted through the cable. With a higher data rate the latency will be improved since more bits are traveling down the cable at a particular time. There are a few factors that can increase the data rate, one would be to increase the data rate of the cable and the other would be to increase the signal speed. Using optical cable usually uses light to carry information down the cable. Nothing in the universe is faster than the speed of light which is probably the best choice for the electrical signal.

4. Read through (but don't yet complete) problems 6.3 and 6.5, then do the following:

In both cases you may assume that the code will be changed suitably such that the original algorithm will now run on multiple cores, and results will be coordinated appropriately. You should state any assumptions about how this is done (as required below), but don't write any code. These problems are abstract and it will be tempting to go overboard with the analysis. Don't. Simply explore the relationship between algorithms and parallel performance, and use algebra to justify your results.

For 6.3, explain why it is difficult to use a large number of cores to speed up the algorithm, and estimate the ideal number of cores for speeding up the algorithm. (State any assumptions and give a general formula for the ideal number of cores.)

Looking at 6.2 on the difficulty of creating parallel processing program it states that have more cores does not always speed-up the program. The Binary search is already fast and it would be difficult to write a parallel processing program for it. Using Amdahl’s Law from chapter 1 and the in term of speed-up versus the original execution time.

$$Speed\ up = \frac{Execution\ time\ before}{(Execution\ time\ before - Execution\ time\ affected) + \frac{Execution\ time\ affected}{Amount\ of\ improvement}}$$

We can rewrite this equation assuming that the execution time before is 1 for some unit of time and the execution time affected by improvement is considered the fraction of the original execution time.

$$Speed\ up = \frac{1}{(1 - Fraction\ time\ affected) + \frac{Fraction\ time\ affected}{100}}$$

To find the ideal number of cores I would think to consider the worst case where it will take # of iteration in the while to find the correct value. For an N-element array the max number of iteration is $\log_2 N$.

$$Execution\ time\ after\ improvement = \frac{\log 2\mathrm{Nt}}{\#\ cores}$$

Assuming that it will take $\log_2$Nt for a single processor to complete the task. We have to find the best # cores for the highest execution time after improvement. The best ideal number of core for this binary search depends on the number of element in the array and should be less and or equal to $\log_2$N.

For 6.5, do both 6.5.1 and 6.5.2. For 6.5.2, explain how the parallel version of mergesort might work, and give a general formula for the expected speedup. Finally, what would the ideal number of cores be?

For problem 6.5.1 the number of cores is much smaller than the length of the array. The speed up will increase as the number of cores are increased. The reason is that the mergesort will need to keep doing the same operation until we have a list of size 1 in length. Having more cores will makes this process faster. I am assuming the plot will be exponential growth.

For problem 6.5.2 I think the parallel version of mergesort will divide the array m evenly between the numbers of cores. Then they will combine their array until they get one sorted array. Not sure on how to get the general formula expect speedup I am assuming it will involve the Amdahl's Law.

Sorry for a lack of context still need to read the chapter on parallelism.