Instructions

There are six (6) questions on the exam. You may find questions that could have several answers and require an explanation or a justification. As we've said, many answers in storage systems are "It depends!". In these cases, we are more interested in your justification, so make sure you're clear. Good luck!

If you have several calculations leading to a single answer, please place a box around your answer.

Problem 1 : Short answer. [30 points]

(a) CD and DVD media have a continuous spiral track whereas hard drive media are organized into thousands of concentric tracks. Given a 4-head, 15,000 RPM hard drive with a 1.0 ms track switch time and 1.5 ms cylinder switch time, calculate the percentage improvement in the sustained media transfer rate between (i) the above design and (ii) an alternate design with a single head and a continuous spiral track.

(3 points). To read four "tracks" worth of sequential data in design (ii) the disk rotates 4 times for a total of 16 ms. In design (i) there are four rotations plus (3 × 1 ms) plus (1.5 ms) for a total of 20.5 ms. The percentage improvement is $\frac{20.5-16}{20.5}$ = *22%.*

(b) Disk-based video recording and playback systems use large I/O requests (e.g., 1 MB and up). Why?

(3 points). Video files tend to be very large and laid out sequentially on disk (because they are written in large chunks). Large disk requests utilize the streaming performance of the disk, amortizing positioning delays over substantial media transfers. Using smaller requests can be inefficient, particularly when servicing competing request streams.

(c) When is DMA a bad choice for communicating with an I/O controller (as compared to programmed I/O)?

(3 points). When the amount of data to be communicated is very small (e.g., several bytes). The time to set up a direct device-to-memory transfer for minimal amounts of data would exceed the benefit of offloading the transfer. Devices such as keyboards and mice tend to only send a few bytes at a time.

(d) Your friend is very excited about their new disk, because it has a one million hour MTBF rating. Explain why your friend can't expect to never have to buy another disk drive, even if they don't run out of capacity.

(3 points). Your friend should have taken 18-746. MTBF isn't a guarantee for the lifetime of an individual disk, it's a statement of the number of expected device failures when an entire manufacturing run is taken in the aggregate.

Also, people are living longer these days—who's to say your friend won't outlive his or her disk?

(e) Many file systems implement directories as unordered lists of entries. Explain why this is a problem when a directory has many entries.

(3 points). It is inefficient to search for individual entries. The search time of an unordered list is proportional to the number of entries in the list (O(n) with an average of $\frac{n}{2}$*). Also, the entire unordered list must be scanned when searching for files that don't exist.*

(f) Modern disks continue reading sectors into their on-board memory after fetching those requested by the host. Why does this improve performance in many systems?

(3 points). Many data references exhibit a high degree of spatial locality to previous references. In other words, if you request block N now, it's in general probable that you will eventually request block N+1.

(g) Why do the buffers in many disk drives use a single, dedicated segment for writes?

(3 points). Usually, data written to disk already live in the FS or database buffer cache, so there is no need to cache writes at the disk to improve read performance. At least one segment should be available for writes to use as a speed matching buffer with the bus. This also makes the cache management firmware simpler, as the disk need not worry about the ordering of write operations.

(h) You have been asked to buy a new desktop computer to be used as the main Carnegie Mellon web server. So far, you've selected a system with a 1 GHz processor and a 3.5-inch, 10,000 RPM disk. You have a little extra money that you can spend on one of three things: (i) a faster processor, (ii) a 2.5-inch disk (with the same capacity), or (iii) a 15,000 RPM disk (with the same capacity). Which do you upgrade? Justify your answer.

(3 points). The Carnegie Mellon web server contains a large number of small text and image files. Option (i) may allow for more concurrent connections, but Carnegie Mellon doesn't tend to have very much traffic. Option (iii) will improve streaming performance (unhelpful for small web files) and slightly improve the average rotational latency. Option (ii) will improve the seek time by reducing the distance between files (in order to retain the same capacity, the BPI or TPI must improve). The seek time for random requests generally dominates rotational latency, so this is the most important factor to improve. Therefore, Option (ii). *(Note: any answer with a reasonable justification would be accepted.)*

(i) What does *fairness* mean in the context of disk drive scheduling algorithms?

(3 points). A "fair" algorithm is one in which no disk request will wait unreasonably long to be serviced, compared to other requests.

(j) Is it possible to recalculate the information in a file system superblock if the block were corrupted? How, or why not?

(3 points). Likely not. Given lots of guessing (block size, inode bitmap location, number of blocks, etc.) and enough analysis of the remainder of the disk, it may be possible to reconstruct enough of the superblock to produce a working file system. However, it's much easier to replicate the super block (as is done in most file systems) in the event of a catastrophic loss of the primary super block.

Problem 2 : Parallel transfer disk drives. [16 points]

Most disks transfer from only one head at a time, because they are engineered that way. Every once in a while, though, a company designs a drive that transfers from/to all read/write heads in parallel. Let's analyze the design trade-off, calling the former "single-head" and the latter "parallel-head."

Assume a disk that with the following characteristics: 10000 cylinders, 10 heads, 100 sectors per track, 10000 RPMs, 0 ms head switch time, 0–15 ms for a seek (based on a linear seek curve). Assume the

following component costs: $10 per platter, $10 for the controller logic, $10 per parallel read/write head for the channel and servo functionalities, $20 for the spindle and actuator.[1] Answer the following questions.

(a) What are the streaming media bandwidths for the single-head and parallel-head versions of this drive?

(4 points) Single-head:

$$\frac{100 \text{ sectors} \times 512 \text{ B/sector}}{0.006 \text{ s}} = 8,533,333 \text{ B/s} = \boxed{8.138 \text{ MB/s}}$$

Parallel-head:

$$\frac{100 \text{ sectors} \times 512 \text{ B/sector}}{0.006 \text{ s}} \times 10 = 85,333,333 \text{ B/s} = \boxed{81.38 \text{ MB/s}}$$

(b) What are the throughputs for random 4KB requests for the single-head and parallel-head versions of this drive?

(4 points) The average seek distance for random requests is 1/3 the full throw distance, so the average seek time is 5 ms. The average rotational latency is 1/2 the full rotation time, or 3 ms. To read 4 KB, 8 sectors must be read.

Single-head:

$$\frac{1}{3}(15 \text{ ms}) + \frac{1}{2}(6 \text{ ms}) + \frac{8}{100}(6 \text{ ms}) = 8.48 \text{ ms/req} = \boxed{117.9 \text{ req/s = 483,018 B/s = 471.7 kB/s}}$$

Parallel-head:

$$\frac{1}{3}(15 \text{ ms}) + \frac{1}{2}(6 \text{ ms}) + \frac{1}{100}(6 \text{ ms}) = 8.06 \text{ ms/req} = \boxed{124.1 \text{ req/s = 508,189 B/s = 496.3 k/s}}$$

(c) What are the costs for the single-head and parallel-head disks?

(3 points) Single-head: $50 (5 platters) + $10 (logic) + $10 (one channel) + $20 (spindle) = $90.

Parallel-head: $50 (5 platters) + $10 (logic) + $100 (10 channels) + $20 (spindle) = $180.

(d) What must the request size be for this disk to provide a performance improvement that matches the increase in cost?

(5 points) Let N be the number of sectors in a request. The per-request performance improvement must be a factor of 2 ($\frac{\$180}{\$90} = 2$).

$$2 \times (\text{single head performance}) = \text{parallel head performance}$$

$$2 \times \left(\frac{1}{8 + 0.06N}\right) = \frac{1}{8 + \frac{0.06N}{10}}$$

$N = 166.67$ which must be rounded up to $\boxed{\text{N = 167 sectors}}$

[1] Note that, in reality, a parallel-head disk is more difficult than these numbers suggest, because the different heads would need to do fine-grain positioning independently.

Problem 3 : Circuit switching vs. packet switching. [18 points]

In this problem, assume that k = kilo = 10^3 and M = mega = 10^6. So 1 MB = 1,000,000 bytes.

The Fibre Channel framing layer (FC-2) describes how data is transferred between nodes and includes the definition of the frame format, frame sequences, communications protocols, and service classes. The basic unit of data transmission in Fibre Channel is a variable-sized Frame. Frames contain 0–2,048 bytes of user data and 36 bytes of overhead for framing, source and destination port addressing, service type, and error detection information. A single higher layer protocol message may be larger than a Frame's payload capacity; in that case, the message will be fragmented into a series of frames called a sequence.

FC-2 defines three classes of service. The one we're interested in for this problem is Class 1: a connection-oriented (virtual circuit or *circuit switched*) service, where two nodes must establish a logical connection prior to any transfer of data. Once the connection is set up it guarantees adequate network resources to transmit all data without further delays (including congestion problems). For the purposes of this problem, assume that connection setup is accomplished by sending a special zero-data-byte frame round-trip from the sender to the receiver and back to the sender.

(a) Electrical signals travel at approximately 300,000,000 m/s. What is the virtual circuit setup time for a 3-km Fibre Channel link?

(4 points) An acceptable answer:

$$2 \times \frac{3,000 \text{ m}}{300,000,000 \text{ m/s}} = 0.000,02 \text{ s} = \boxed{20\ \mu\text{s}}$$

A more accurate answer includes both the propagation delay and the time to get the bits on the wire, assuming the data rate is 100 MB/s:

$$2 \times \frac{3,000 \text{ m}}{300,000,000 \text{ m/s}} + 2 \times \frac{36 \text{ B}}{100,000,000 \text{ B/s}} = 0.000,020,72 \text{ s} = 20.72\ \mu\text{s}$$

(b) An optical 3-km Fibre Channel link supports a 100 MB/s data rate. What is the total transmission time (including setup) for sending the following over a 3-km Fibre Channel link: (i) one 512-byte sector, (ii) 1 MB?

(5 points) (i) Count the propagation delay and the frame transmission time.

$$20\ \mu\text{s} + \frac{512 + 36 \text{ B}}{100,000,000 \text{ B/s}} = \boxed{25.48\ \mu\text{s}}$$

(ii) Determine how many frames must be sent: $\frac{1,000,000\ B}{2048\ B}$ *= 488 full frames plus one 576-data-byte frame.*

$$20\ \mu\text{s} + \frac{(2048 + 36 \text{ B})}{100,000,000 \text{ B/s}} \times 488 + \frac{(576 + 36 \text{ B})}{100,000,000 \text{ B/s}} = \boxed{10.19 \text{ s}}$$

Local area networks (LANs) avoid the connection setup delay by transferring data immediately without first establishing a connection. This is known as a *packet-switched* service. Because network resources are not reserved for any individual connection, it is possible for data to be lost en route from the sender to the

receiver (e.g., because of congestion). When this happens, the data must be retransmitted by the sender after a timeout.

The maximum TCP/IP packet size using Ethernet is 1500 bytes: 0–1460 bytes for user data and 40 bytes of overhead (20 bytes for the TCP headers and 20 bytes for the IP headers). Assume the underlying network is a 1 Gbit/s switched Ethernet. Assume also that the retransmission timeout is five times the round-trip time between the sender to the receiver, and that any packets following a lost packet must be retransmitted as well.

(c) What is the total transmission time (including setup) for sending the following over a 3-km TCP/IP link: (i) one 512-byte sector, (ii) 1 MB?

(5 points) (i) There is no "delay." 1 Gbit/s = 125,000,000 MB/s.

$$\frac{512 + 40 \text{ B}}{125,000,000 \text{ B/s}} = \boxed{4.416\ \mu\text{s}}$$

(ii) Determine how many packets must be sent: $\frac{1{,}000{,}000\ B}{1460\ B}$ *= 684 full packets plus one 1360-data-byte packet.*

$$\frac{(1460 + 40 \text{ B})}{125,000,000 \text{ B/s}} \times 684 + \frac{(1360 + 40 \text{ B})}{125,000,000 \text{ B/s}} = \boxed{8.2192 \text{ ms}}$$

(d) When sending 1 MB of data, what is the maximum error rate on the TCP/IP-based network (as a percentage of packets sent) before the use of Fibre Channel would be preferred?

(4 points) Each packet error can be thought of as a "bubble" in the stream of packets. This bubble adds two times: the retransmission timeout, plus the time to retransmit the packet.

$$\text{Retransmit time } t_R = 5 \times 20\mu s + \frac{1500 \text{ B}}{125,000,000 \text{ B/s}} = 0.112 \text{ ms}$$

The total time spent retransmitting cannot be greater than the difference in times between parts (b) and (c).

$$(10.196 - 8.219 \text{ ms}) \div t_R = 17.7 \text{ packets} = 17 \text{ packets}$$

Therefore, $\frac{17\ packets}{684\ packets}$ = $\boxed{2.5\%.}$

Problem 4 : Metadata integrity with write-ahead logging. [20 points]

Write-ahead logging helps a file system protect its metadata from corruption caused by unfortunately timed system crashes. Joe FS designer needs your help with parts of his design.

(a) Joe understands how to add entries to the log and make sure that they are flushed before the corresponding metadata changes. But, he doesn't understand how crash recovery works. Explain what he should do with the log after a crash.

(5 points) Joe must apply all of the updates in the log (since the last metadata checkpoint) to the on-disk metadata structures. For example, if there is a log entry to rename a file, he should rename it. The operations must be executed in order. After this completes, the file system metadata will be in a stable, mountable state.

(b) After running his system for about two weeks, Joe experienced his first system crash. It took 3 hours to recover. Jane FS designer told him that he forgot to include support for checkpointing and explained how to do it. His system executes 10 metadata operations per second. How frequently should his system checkpoint if he wants a one second recovery time? (Assume his system can clean up 1000 metadata operations per second during crash recovery.)

(5 points) $\frac{1000\ operations}{10\ operations/sec}$ = *100 s between checkpoints.*

(c) Joe heard that sometimes disk drives grow media defects and thereby destroy one or more sectors of existing disk content. He thinks that he doesn't have to worry, because he uses write-ahead logging. Is he correct? If so, explain why. If not, explain why not and propose an additional crash recovery step for addressing the problem.

(5 points) Joe should have taken 18-746. If a disk block is lost, the data contained in that disk block is lost forever. Updates in the log will not help recover what the initial contents were (if a directory entry is lost, and Joe's log entry says delete a file, how can Joe know which inode corresponds to the lost directory entry?) Even worse, a block from the log could be corrupted.

A block lost under a logging file system may be recovered after a crash by using a `fsck`*-style tool to make multiple passes over the metadata structures and making the best effort at recreating the file system.*

Two methods to prevent blocks from being lost in the first place include replication (on the same disk or a different disk) and redundancy (keep parity information across different blocks).

(d) After recovering from one particular crash, Joe discovered a file that contained random data from a previously deleted file. Explain how this could happen in a write-ahead logging system, and suggest a mechanism Joe can use to prevent this integrity problem from happening again.

Joe's log, like most, only records metadata updates, not data updates. Normally, when new blocks are assigned to a file, those blocks are zeroed out or initialized with immediate data. After a crash, if a log entry shows that new blocks should be assigned to a file, the file system has no way of knowing whether or not the corresponding disk locations were updated with real user data. The recovery routine could zero the blocks out, but then it is running the risk of deleting valid data that was written out successfully before the crash.

A solution to this problem is to add log entries noting when data are written to disk. This would increase the log size, but would prevent the problem that Joe encountered. Data leaks can be extremely undesirable.

Problem 5 : Use of indices in database systems. [16 points]

Assume a relational database table with 100 million records, each 64 bytes in size. The table is sorted by its key attribute. Assume the table is stored on a disk that streams data at 10 MB per second and performs random single-block requests in 10 ms. Answer the following questions.

(a) How long does it take to search for a particular record, if it must be identified by an attribute other than the key??

(4 points) The record must be found by scanning records, which (on average) requires looking at half of the 6.4 GB table. So, $\frac{3.2\ GB}{10\ MB/sec}$ = *320 sec*.

(b) Assume an index exists for the relevant attribute, allowing a specific record to be identified with 2 random requests (one for the leaf index block and one for a block of table data). Now, how long does the search for a particular record take?

(4 points) 2 random requests at 10 ms each, or 20 ms.

(c) Assume a query that searches the table for records whose value for the given attribute match a particular function. If 1% of the records match, how long does the query take if one uses the index? If one does not use the index?

(4 points) Assumption: using the index, each record lookup requires the standard 2 random requests. (This ignores the likely case of multiple records being described by each index block, and of index blocks being cached.)

So, 2 random requests are needed for each of 1,000,000 records (1% of 100,000,000), or 20,000 seconds.

If not using the index, one scans the entire table, which takes twice as long as in part (a), or 640 seconds.

(d) What is the crossover match percentage? (That is, at what percentage is use of the index to check specific records equal to simply scanning the entire table.)

(4 points) The full table scan takes 640 seconds. For index-based lookup to take the same amount of time, with the assumptions in part (c), only

$\frac{640\ sec}{20\ ms/record}$ *= 32,000 records of the 100,000,000 can match, or 0.032%*.

Problem 6 : Instructor trivia. [up to 2 bonus points]

Each correct answer received one bonus point, to a maximum of 2.

(a) How many sons does Dr. Ganger have, and what are their names?

Dr. Ganger has 2 sons: Timothy Jacob and William David.

(b) In which year did Dr. Riedel matriculate at Carnegie Mellon, and in which year did he finally receive his Ph.D.?

Matriculate: to be admitted into a group, especially a college or university. Dr. Riedel matriculated at Carnegie Mellon in 1989 and received the degree Doctor of Philosophy in 1999.

(c) From where did Mr. Griffin receive his undergraduate degree?

Mr. Griffin hails from Auburn University, home of the Auburn Tigers.