CMU 18-746/15-746 **Storage Systems** **3 Mar 2104**

Spring 2014 **Exam 1**

Name: _____

Instructions

There are three (3) 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. [48 points]

(a) Traditionally, when a file's contents are copied to another file, twice as many disk blocks are used (i.e., one for each of the files). What critical additional metadata would be needed to allow the same disk blocks to be used safely for both files? Explain your answer.

Reference counts on file blocks. Like with link counts in inodes, which allow multiple names to refer to a single file, link counts on file blocks would allow one to know when there are pointers to a given block from multiple inodes.

More info: Such a block refcount serves multiple functions. As with inode link counts, a block refcount would allow the file system to know when it is safe to mark the block as "free" (only when the reference count reaches zero). Also, if a file tries to update the block, and the refcount is more than 1, it means the block is shared and must be replaced with a dedicated block (newly allocated and initialized with the shared contents); this technique is generally referred to as "copy on write".

(b) Jeff likes to buy used storage devices, to save money. Before using a new storage device, Jeff always writes blocks of zeroes to every LBN in order to avoid the possibility of reading any previous owner's data. His most recent purchase is an SSD. Why could his practice of initializing all LBNs harm the SSD's performance? Explain your answer.

The SSD would consider all LBNs to be "occupied", minimizing the overprovisioning property and maximizing the cleaning work needed to clear erase regions, which in turn degrades performance for future writes. (The TRIM command exists to reduce the set of LBNs that are considered occupied.)

Simply saying "causes write-amplification" or "reduces SSD lifetime" were not considered complete answers. While true of all SSD writes, they do not make clear the specific reason that Jeff's initialization harms subsequent performance.

(c) Imagine that you work for a large Internet services company that has 100,000 disks in its data center. If three replicas of any given file block are stored on a random three of the disks, and each disk has an MTBF of 100 years, how many data loss events would you tell your boss to expect in a one-year period (assume no rebuild)?

Assuming that there are enough data blocks, any three disk failures would result in data loss. So, the mean time until the first data loss would be $\frac{100years}{100000} + \frac{100years}{99999} + \frac{100years}{99998}$*, which is close to* $\frac{3years}{1000}$*. Each subsequent disk failure would result in additional data loss, at approximately* $\frac{1year}{1000}$*. Overall, about 998 in a year (one in the first 0.003 of a year plus one every 0.001 of a year thereafter).*

Many people interpreted the question as being about 3-way mirrors (with one left over of the 100,000 ;)), which we gave substantial partial credit born of a wish that we had explicitly noted that that's not what was described (to eliminate any such confusion).

(d) Imagine a redundant disk array that used a parity disk and performs regular scrubbing to both find defective sectors and verify that the parity matches the data. If the scrubbing process discovers that the parity does not match the data, what should the disk array controller do?

Fix the parity, using the data from the data disks, so that any subsequent rebuild operation will use matching parity to reconstruct missing data.

Some people noted that the right thing to do would be to discard the entire stripe, since it might be one or more of the data stripe units that are wrong. Such an approach could be used, but it would have the effect of discarding valid data stripe units, so it is not common.

Some people described how to recover from a defective sector being discovered, but not how to deal with a parity mismatch. These are two different situations: a defective sector is an erasure, whereas a parity mismatch indicates that one of the disks (possibly the parity disk) has the wrong content without giving any indication of which.

Some people seem to have confused parity at the disk array level with per-sector ECC at the disk level. These are very different forms of redundancy addressing different types of failures.

(e) Fred wants to increase the maximum file size without changing either the file system block size or the total number of block pointers in the inode. (He is willing to change how that number of block pointers are used and backward compatibility is not a concern.) What is the easiest way for him to achieve his goal? Explain your answer.

Increase the level of indirection for at least one of the inode's block pointers. For example, the last direct pointer could be made a single indirect pointer, or a triple indirect pointer could be made a quadruple indirect pointer. The easiest change, in terms of file system code modification, might be to increase the level of indirection of the current "most indirect" pointer by one (e.g., from 3 to 4).

Most people proposed some variant of the above, but few explained why their specific change would be "easiest".

(f) Imagine a system with an I/O workload described by a closed arrival process with zero think time. If the average service time is cut in half (reduced by 50%), what happens to the average response time?

It would also be cut in half. Given zero think time, the average number of queued requests would remain unchanged, while the average time a request waits for each request serviced while it waits would be cut in half.

Detailed derivation, for clarity: Response time (R) = Queue time (Q) + Service time (S)

When arrivals are closed, there are always the same number of requests in the system (N), either thinking, queued or in service.

When think time is zero, the number of requests in service is 1 and the rest are queued, N-1.

*With FIFO ordering (by default), the Queue time (Q) = (N-1) * Service time (S), so*

*Response time R = (N-1) * S + S = N * S*

If just the service time changes, S = S / 2

*Then it is still true that Q = (N-1) * S*

*So Q = (N-1) * S / 2*

*And R = Q + S = (N-1) * S / 2 + S / 2 = N * S / 2 = R / 2*

So the response time is reduced by half.

Problem 2 : More short answer. [40 points]

(a) Imagine an operating system with a log-structured file system that uses a segment size exactly equal to the Flash erase size of the SSD used for the actual storage. If the file system does not use TRIM, would you ever expect the SSD firmware to need to copy live data in order to clear regions to be flashed? Explain.

No. Writes from the log-structured file system to the SSD would always overwrite LBN ranges corresponding to entire Flash erase regions, clearing the previous locations of those LBNs without any SSD-level copying. Essentially, the cleaning happens within the log-structured file system, even though the two devices are not explicitly coordinating.

(b) Harold developed a file system that uses write-ahead logging to ensure metadata integrity, but is concerned that it may experience integrity problems when using disks that aggressively employ write-back caching to improve performance. What is the most significant concern that you would expect Harold to have? Explain your answer.

If the disk allows writes to be reordered, as they propogate from the disk's write-back cache to the persistent media, log writes could end up being saved after metadata writes that they describe. This breaks the write-ahead logging foundation of writing the log first, making it an ineffective tool for metadata integrity.

Many students noted that power loss could result in loss of the write-back cache contents. While true, such loss is a concern for metadata integrity only if it results in inconsistent metadata (rather than just loss of the most recent updates, just like any power loss event that affects the main memory file cache).

(c) Jonah has enjoyed the benefits of the clever shortest-seek-time-first request scheduler that he implemented for his operating system. But, Diane tells him that he shouldn't expect it to provide much benefit (over simple first-come-first-served scheduling) once the system is using a Flash-based SSD instead of a traditional mechanical disk. Do you agree with Diane? Explain your answer.

Yes. SSD service times do not have a "distance between locations accessed" correlation akin to the seek distance for mechanical disks.

(d) Imagine a disk array that uses parity to tolerate disk failures and uses 5 disks that can each service 100 I/Os per second. Assuming a read-only workload, how many more I/Os per second can the array service if it uses striped parity (RAID 5) instead of a dedicated parity disk (RAID 4)?

An additional 100 I/Os per second, or 25% more, because all five disks store both data and parity, enabling all five to service data reads instead of just four.

(e) Imagine a disk array with a 16 KB stripe unit size. Would you expect file system performance to increase or decrease if the file system block size were changed from 16 KB to 32 KB? Explain your answer.

Decrease. Each I/O for a single file block would now require use of two disks instead of just one, cutting peak throughput for the array in half because no significant benefit would be expected from having each of the two devices transfer half of the 32KB. (The request sizes are too small.)

Some noted that performance would be unlikely to change for very large files, assuming that the file system streams from all disks for such files in either case. While true, this answer alone would be incomplete.

Problem 3 : Bonus questions. [up to 2 bonus points]

(a) What do we call the online system used to view the lab 1 writeup and submit solutions?

Autolab

(b) Which instructor delivered the lecture on file system integrity?

Actually, Greg and Garth both did part of it, so we took either answer. (No, we didn't ask the question that way on purpose... we forgot that we had handed it off midway.)

(c) What percentage of lab 1 do you estimate that you have completed so far?

Hopefully 100% ;)

(d) Are any of the TAs related to one another?

Yes, the Jaltade's