CMU 18-746/15-746 **Storage Systems** **23 Feb 2011**

Spring 2011 **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) Imagine a file system that uses write-ahead logging (WAL) to protect the integrity of its metadata. What is one reason that a "fsck" program would still be needed, even if the WAL implementation is perfect?

"Grown defects" on a disk or sector writes that are interrupted, such that the disk's ECC mechanism cannot recover them. Such kind of errors cannot be protected by WAL, but unaffected portions of the file system can be recovered by "fsck".

(b) Imagine a file system that uses synchronous writes (for update ordering) to protect the integrity of its metadata. What is the minimum number of disk writes that must be completed during a file create system call? Explain your answer.

During the file create system call, only the new initialized inode must be written (with a synchronous write) for integrity protection. It must be written before the directory entry that points to it.

Other data structures that get updated and must be written eventually include:

- *the inode bitmap (for allocation)*
- *the parent directory (for new entry)*
- *the superblock (for the inode count)*

(c) Imagine that you work for a large Internet services company that has 100,000 disks in its data center. How many disk failures would you tell your boss to expect in a one year period, if each disk has a MTBF of 100 years?

$Expected\ \#failures = \frac{\#disks \times period}{MTBF} = \frac{100{,}000 \times 1}{100} = 1000$

(d) Briefly describe a scenario in which Shortest-Positioning-Time-First disk scheduling could give lower service times than Shortest-Seek-Time-First for both requests, assuming only two disk requests are pending.

Positioning time includes seek time as well as rotation time. Imagine a senario where req1 has slightly shorter seek time but much longer rotation time than req2. For SSTF, req1 is served first, but the disk head has to wait for a whole rotation to hit the actual acquired sector; then req2 is served, but it requires much higher rotation time than what would have been if it was scheduled first. For SPTF, req2 is first served with little rotation time, and when the disk head seeks to the track of req1, it just hits the requested sector with also little rotation time. In this case, SPTF gives lower request time than SSTF for both requests.

(e) Imagine a system with an I/O workload described by a closed arrival process and zero think time. If the average service time doubles, what happens to the throughput?

The throughput halves.

(f) You buy a disk that rotates at 6000 RPM (100 rotations per second) and has 1000 512-byte sectors on every track. When reading data from it sequentially, as fast as possible, do you expect it to provide over 50 MB/s? Explain why or why not.

No. In theory, if the disk head reads or writes the same track over and over it could access 50 MB per second. However, in practice, this is not ture due to overhead of cylinder or track switches.

Problem 2 : More short answer. [48 points]

(a) Imagine a 5-disk disk array subsystem configured for RAID-5 with a 16 KB stripe unit size. If you were creating a log-structured file system on it, with a file system block size of 4 KB, what would be a good segment size to use? Justify your answer.

64KB (the stripe size) or multiples of 64KB. Log structured file systems write sequentially into log segments. If the segment size equals 64 KB (or multiples of 64 KB), the new parity can be computed directly from the new data. For segment size smaller than the stripe size or not an integer multiple of it, writing a full stripe will require reading some old data to update the parity.

(b) Imagine a 10-disk disk array configured to use RAID-4, with 8 data disks, one parity disk, and one spare disk. When using this disk array, Joe notices that his performance drops dramatically for about 30 minutes after a disk failure. Suggest a configuration knob that Joe might try changing in order to reduce the performance drop when he next encounters a disk failure. Explain.

The performance dropped because the RAID-4 disk array was actively reconstructing the failed disk onto the spare disk, consuming much disk bandwidth. One possible configuration change to reduce the interference is to lower the priority of the reconstruction process.

(c) Fred has constructed a perfect fsck program. When he uses it on his file system, after a crash, it reports a regular file inode with a link count of two that has no directory entries pointing to it. Describe a possible set of file system operations, performed just before the crash, that would explain how such a situation could arise.

One possible set of operations:

1. link("f1", "f2");

2. unlink("f1");

3. unlink("f2");

After operation 1, the inode pointed by f1 and f2 has a link count of 2. Operation 2 removes the directory entry of f1 and reduces the link count to 1. Operation 3 removes the directory entry of f2 and reduces the link count to 0 (the file is removed). The crash happens after removal of directory entries is reflected to disk but before the inode is updated.

(d) Given an ext-2 filesystem that supports 4KB blocks, what is the largest filesize supported if inodes contain 10 direct blocks, 1 indirect block, and 1 double-indirect block? Assume block pointers are 32-bits.

10 direct blocks $\rightarrow 10 \times 4KB = 40KB$

1 indirect block $\rightarrow \frac{4KB}{32b} = 1K$ *(direct block pointers)* $\rightarrow 1K \times 4KB = 4MB$

1 double-indirect block $\rightarrow$ *1K indirect block pointers* $\rightarrow$ *1M direct block pointers* $\rightarrow 1M \times 4MB = 4GB$

The largest filesize supported is $40KB + 4MB + 4GB \approx 4GB$

(e) Imagine a mirrored pair of disks, where each disk can service 100 I/Os per second. Given a workload that issues requests at an exponentially distributed rate w/mean 20 I/Os per second, what is the average disk response time if every request is a write?

Since every request is a write, each request uses both disks.

$T_s = \frac{1000ms}{100} = 10ms$

$util = \frac{20}{100} = 0.2$

$T_q = T_s \cdot \frac{util}{1-util} = 2.5ms$

$T_r = T_q + T_s = 12.5ms$

T_s*: service time*

T_q*: queue time*

T_r*: response time*

(f) Your fsck program finds a directory with a link count of 5. How many sub-directories should that directory have? Explain you answer.

3 sub-directories. Each sub-directory addes to 1 link count to the directory. The other 2 links come from the parent directory and the "." self-link.

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

(a) Which instructor is on vacation?
Garth

(b) Which school's sports teams does Greg cheer for (most strongly)?
Michigan

(c) Where should Swapnil work, after he finishes his PhD?
Wherever...

(d) What color is Lianghong's hair?
black