CMU 18–746 **Storage Systems** **12 Oct 2006**

Fall 2006 **Exam 1**

Name: _____

Instructions

There are four (4) 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. [63 points]

(a) Identify two benefits of frequent checkpointing in a journaling file system?

A:

Recall that, during a checkpoint all the modified filesystem blocks (data and metadata) are written to stable storage, allowing the log to be truncated.

The primary advantage is that recovery is faster: The system needs to replay the log starting from the most recent checkpoint.

Another advantage is that the space dedicated to the log can be smaller.

(b) Some file systems use disk space allocation policies that try to place large files toward the beginning of the LBN space. Why would they do this?

A:

The beginning of the LBN space typically corresponds to the outer tracks of the disk, where the data transfer rates are higher. This makes large sequential accesses (such us those used to read a large file) faster.

(c) Most disk drives maintain a cache, even though the block cache in most systems' main memory has much larger capacity. Why?

A:

The disk drive cache is used for low-level optimizations performed by the disk controller such as prefetching and zero-latency access.

(d) Project 1 involved direct access to the raw disk interface rather than use of the file system. Why not use the file system interface instead?

A:

Project 1 required direct access in order to avoid interference from the file system buffer cache. Using the filesystem cache would have made the extraction of disk characteristics impossible.

(e) Most file systems keep multiple copies of the "superblock" on the disk. By doing so, what disk-level problem are they protecting against?

A:

Redundant superblock storage protects against sector defects, that could otherwise render the filesystem unusable by taking away this critical structure.

(f) It is difficult, in most systems, to perform disk request scheduling that simultaneously considers low-level positioning costs (e.g., Shortest-Positioning-Time-First) and high-level priorities among requests (e.g., because they were generated by applications of differing importance). Explain why.

A:

Most systems don't have a way to get both types of information to the same place in the system. OS device drivers don't have detailed information about disk internals hidden behind the storage interface. Most disk interfaces don't allow for priority information to be conveyed with requests. Thus, there is no place in the system that is able to consider both.

(g) Imagine that you have two large database tables and you want to satisfy the query "select * from R,S where R.id=S.id". Under what circumstance would you expect a sort-merge join to perform particularly well?

A:

A sort-merge join would work very well when R and S are already sorted on their 'id' attribute.

(h) Explain why soft updates does not provide the crash recovery semantics most naturally desired for RENAME operations that move a file from one directory to another.

A:

A rename operation would ideally be atomic. Soft updates allows a deterministic ordering on the updates, but not atomicity. With renames, soft updates can ensure that the original pointer to the file is not deleted until after the new pointer is created. But, with soft updates, it could be the case that after a crash, both the old and the new directories point to the file.

(i) Web sites that sell disk drives usually report their capacity (in GBs) and their rotation speed (in RPMs). Why should someone buying a disk not expect the 15000 RPM disk to be twice as fast as the 7200 RPM disk?

A:

The rotation speed affects only the rotational latency and transfer time components of disk request service time. Since seek time is not affected by doubling the rotational speed, one should expect the disk performance improvement to be less than 2x.

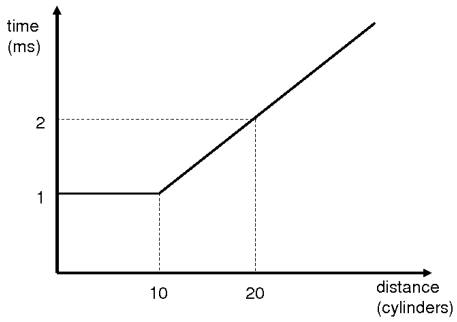

Figure 1: Disk Seek Curve

Problem 2 : Optimizing Disk Access Patterns . [13 points]

Consider a disk with the following characteristics:

- 10000 RPM
- Single zone with 700 sectors per track
- The seek curve shown in Figure 1.

(a) Compute the time it takes to read a directory and ten single-block files contained within it, assuming that all data and metadata are stored in the same cylinder group (of exactly 10 cylinders in size) and that no two inodes are in the same inode block. (Note: please be sure to state any assumptions.)

A:

We need to access the following blocks:

- *directory inode*
- *directory data block*
- *10 file inodes*
- *10 file data blocks*

Assuming a filesystem that does not optimize the placement of inodes and blocks within the cylinder group, we can expect each request to involve a seek, rotational latency and data transfer components.

- *Seek time:Since the seek distance is at most 10, we're going to assume that the seek time for every request is 1ms (from Figure 1). (Note: this ignores the possibility of two requests happening to be on the same track; so, lets call that an assumption.)*

- *Rotational latency:The average rotational latency is* $0.5 \times 60/10000s = 3ms$.
- *Transfer time:Assuming each filesystem block is a single disk sector, it takes* $6/700ms = 0.008ms$ *to transfer a block.*

Thus the total time per request is $1ms + 3ms + 0.008ms = 4.008ms$. *The entire operation consists of 22 requests, thus the total time will be* $22 \times 4.008ms = 88.176ms$.

(b) Which component of disk access time dominates the time in (a)?

A: *Rotational latency*

Problem 3 : Short design questions. [24 points]

(a) In an LFS implementation, it is important to identify the last log segment written before a crash when recovering from that crash. Assuming a static order in which the segments should be used, explain one approach a system could use to do this.

A

One can use a simple monotonically increasing number and store the next value from it somewhere (e.g., a segment header) in each segment that is written out. But, the key challenge is determining whether the last segment, as identified by the number, was completely written or only partially written (e.g., because it was cut off by the crash). One way to address this challenge is with a checksum computed over the entire segment contents, stored in the segment header with the number, and verified when trying to determine whether it was fully written.

(b) You are considering the use of a secondary index on the main table in your database. Identify one performance benefit and one performance penalty associated with the choice. [Hint: think about different request types.]

A:

A secondary index is very beneficial for workloads that consist of looking up individual records in a large database, allowing the system to locate the target records much faster, compared to scanning the table.

But, an index must be updated on every change to the database, increasing the amount of work necessary during database updates. This performance penalty must be weighed against the benefit.

(c) Most file systems use write-back caching, which decouples application performance from disk access times for writes. At some point, however, modified data is written out to disk. Identify two reasons for such disk writes.

A:

(a) Write-back cache full. When the cache is full some modified data gets flushed to the disk.

(b) Bounding data vulnerability. If the write-back cache is not non-volatile, then a crash will cause any modified blocks in it to be lost. Many systems write out data from such caches within a certain amount of time (e.g., 30 seconds) to bound the amount of work potentially lost.

(c) fsync *calls: Applications such as filesystems require that modified data blocks get written immediately.*

Problem 4 : Instructor trivia. [up to 2 bonus points]

(a) Timmy will be dressing up as a football player for Halloween, wearing the winged helmut of daddy's favorite football team. Which team is it?

University of Michigan (the Michigan Wolverines... Go Blue!)

(b) Identify at least one "term" coined in 18-746 this term to refer to a desirable or undesirable system property. What have we used it to mean?

"Unfortunate", referring to poor performance (usually very poor)

(c) Which team won the Greece vs. USA semifinal match in the 2006 basketball world championship? Bonus on bonus: associate the instructor and TA with the appropriate country of origin, from among the teams involved. *GRE 101 , USA 95*

Stratos is from Greece; Greg is from U.S.A.

This page intentionally left blank in case you need scratch space.