CS 240 - Final Exam

Stanford University
Computer Science Department

June 3, 2014

!!!!! SKIP 20 POINTS WORTH OF QUESTIONS. !!!!!

This is an open-book (but closed-laptop) exam. You have 75 minutes. Cross out the questions you skip. Write all of your answers directly on the paper. Make your answers as concise as possible. Sentence fragments ok.

NOTE: We will take off points if a correct answer also includes incorrect or irrelevant information. (I.e., don't put in everything you know in hopes of saying the correct buzzword.)

Question	Score
1-10 (50 points)	
11-15 (45 points)	
Total (max 75 points):	

Stanford University Honor Code

In Accordance with both the letter and the spirit of the Honor Code, I did not cheat on this exam, nor will I assist someone else cheating.

Name and Stanford ID number:

Signature:

1. *(5 points)* The LFS paper does not precisely state what can happen if a crash occurs when a file is being written to disk. Give a simple, efficient approach that uses features of LFS's existing infrastructure to make writing a file atomic on close (i.e., the same guarantee that the LBFS file system tries to provide with only partial success). Hint: do not use rename and you may ignore partial modifications.

2. *(5 points)* Figure 8 in the LFS paper compares LFS's performance to FFS. You re-run this experiment on top of two NFS file systems, one running on top of LFS and the other on top of FFS (SunOS). Roughly speaking what do you expect to happen to the relative performance difference between NFS-LFS and NFS-FFS?

3. *(5 points)* xsyncfs:

   (a) Figure 3: if we disable the drive's write cache how does this table change?

   (b) Why is ext3-sync's performance so much closer to xsyncfs in Figure 5 versus Figure 4?

4. *(5 points)* Leases: Redraw Figure 2 assuming a single client fails.

5. *(5 points)* Since LBFS is inherently a request / response protocol, you decide to speed it up by switching from TCP to UDP. Explain concretely what modifications you would need to make to the messages sent in Figures 2 and 3. Hint: Just as messages can be lost, machines can fail.

6. *(5 points)*

   (a) "LBFS has a minimum chunk size." Give the graph in the paper that appears to contradict this statement and why the data looks the way it does.

   (b) "File systems can also better tolerate network latency than remote login sessions." Give the graph in the LBFS paper that shows this most directly and state why.

7. *(5 points)* Map/reduce: "We rely on atomic commits of map and reduce task outputs..." What do they mean by atomic? How do they make map and reduce output commits atomic?

8. *(5 points)* Give two places in Haystack where you could use a simple whole file hash (similar to LBFS) to potentially significantly save time or bandwidth. Please provide an intuition for why you would expect these hacks to at least potentially matter in practice.

9. *(5 points)* Haystack:

   (a) Figure 8: what would this look like if they were doing badly?

   (b) Figure 10: why does the number of reads increase over time for a write enabled machine?

10. *(5 points)* There are a variety of procedures you cannot call from a Linux kernel interrupt handler. Assuming you have a list of interrupt handlers (and by construction, non-interrupt handlers)— explain how to statistically infer probable candidates.

11. *(5 points)* You have a pre-existing checker that flags when code uses network data without first sanitizing it. The checker would work great, except that in the code you want to check it is often unclear whether a given piece of data comes from the network or from trusted sources. Fortunately, you notice frequent uses of `ntohl` and `ntohs` which convert 32-bit and 16-bit integers from "network byte order" to "host order." Give:

    (a) A belief-style argument for how the checker can use these.

    (b) The checker in the Deviant paper that uses belief analysis to solve the most similar problem to this one.

Note: This question and the rest are worth 10 points!

12. *(10 points)* Carl states that since ESX controls all disk and output devices it's possible to implement a version of xsyncfs inside of ESX at the level of the disk drive. Explain how to do so. Roughly speaking, how do you expect the performance gain of this approach to compare to original xsyncfs paper?

13. *(10 points)* List four places in NFS where techniques / ideas from other papers can be adapted to *non-trivially* improve performance or consistency. Be very concrete about which data structures or operations they are applied to and when.

14. *(10 points)* List four techniques / ideas used in LFS that also appear in Haystack.

15. *(10 points)* Give examples of four ideas / suggestions from the "hints" paper that are used by map/reduce.