

CSE 121: Homework 2

Due: Thursday, 2/12/2009, in class (hardcopy or email kaiseni@cs)

Note 1: If emailing, use PDF, not DOC/DOCX (Google for CutePDF if you need a PDF printer)

Note 2: Answers quoting passages (verbatim or close to verbatim) receive no credit. Summarize.

1. Soft Updates and Journaling

- (a) As we saw in class, file systems usually address the issues of performance and reliability. How do Soft Updates and Journaling deal with these issues?
- (b) Journaling requires write-ahead logging. That is, the log gets written to disk before any of the actual operations. Because disk operations are blocked by the log operation, Louis Reasoner believes log writes should always be flushed to disk immediately. Do you agree? Why?
- (c) As we saw in lecture, Soft Updates can create extra write operations in order to keep the file system consistent. However, the authors claim they can minimize the frequency of extra writes through careful buffer cache replacement. Give a disk flush order example where two different number of disk flushes can occur. [Hint: It can be done with 3 blocks]
- (d) Looking at figure 1, it appears that LFFS-wafs-2sync performs similarly to FFS until the file size becomes 32k. Then they perform comparably to Soft Updates. Why is this?

2. Rio

- (a) What does Rio assume about the buffer cache that previous file systems that we have read, did not assume?
- (b) The registry contains important data structures that allow Rio to make sense of the file cache. Thus its integrity is important. Ben Bitdiddle thinks he can achieve added reliability by flushing this registry to disk. Do you agree? Why?
- (c) Because file caches are now “non-volatile”, Rio must be careful with ordering when updating metadata in the buffer cache in order to maintain file system consistency. Describe a way to deal with this. (Hint: Think back to the previous papers we read.)

3. Caching/Prefetching

- (a) How does the controlled-aggressive prefetching policy answer the following questions: 1) When should prefetching be done? 2) What should be prefetched? and 3) What should be evicted?
- (b) Ben Bitdiddle wants to make a modification to LRU-SP that uses placeholders, but not swapping. What problem does this cause?
- (c) In LRU-SP, when an application chooses for eviction a page not at the end of the global LRU, a placeholder is created. These placeholders accumulate until a page is accessed.

However, because Alyssa Hacker was procrastinating on her project and rushing to finish before the deadline, her implementation only keeps the most recent placeholder (instead of all of them). Give an access pattern example of how a process could gain an extra block in the buffer cache. Assume the buffer cache is empty to begin.

- (d) In their implementation (ACFS), they used a proxy PCM and ACM component. Why might they want to do this rather than directly asking the process? [Hint: Think back to lecture]

4. Global Memory Service

- (a) In their design (not implementation) what happens when node A takes a page fault, but the page it wants is in global memory on node B?
- (b) When a page fault occurs on a node and the contents are on disk, it does not necessarily evict its own page, it uses LRU on the cluster. LRU is approximated in GMS. Describe at a high level how this is done (also again in its design, not implementation).