Problem Set 2

15-440/15-640 Distributed Systems Spring 2017

Assigned: Tuesday February 21, 2017

Due: 10:00 am on Tuesday February 28, 2017

Submission procedure:

- Create a .pdf of your answers and upload to Autolab**.**
- If you handwrite your answers, scan into .pdf before uploading. Please use a real scanner that outputs easily readable .pdf. Images taken with a camera (e.g., your smartphone or tablet) are not acceptable. Illegible submissions will receive a zero grade.

Question 1 (15 points)

Impressed by your performance in 15-440/15-640 at CMU, DropBox hires you to replace their poorly-performing full-replica design with a new product that uses a genuine caching mechanism. This new product is targeted at a group of companies in the movie animation industry whose workloads are video reviewing (read-only) and video editing (read-write). Video files range from 50 MB to 150MB in size, with a mean of 100 MB. About 90% of the opens are for video reviewing, and about 10% are for video editing. A typical review session takes 30 minutes, while a typical editing session only takes 10 minutes. The companies that will use this system are globally distributed across the Internet, with a mean end-to-end bandwidth of 10 Mbps and a mean RTT of 100 ms. They all use Linux clients.

For the new DropBox product, you architect a write-through whole-file caching system. On a close after a video editing session, the entire contents of the just-edited file are transmitted to the server. You use read and write leases on whole files as the cache consistency mechanism.

A. Suppose you would like most sessions to complete with a single lease request. In other words, you would like to reduce the number of lease renewal requests. How long should read leases and write leases be in order to achieve this goal? State and justify any assumptions that you make.

B. Your DropBox team has implemented and deployed this system, using the lease periods stated in your answer to A. Now consider a situation where a new lease request arrives at the server when there are 5 other requests already waiting ahead of it in the FIFO queue for a lease. What is the worst case waiting time for this new request, before it receives its lease?

C. For B, what is the best case waiting time?

Question 2 (15 points)

An Internet service makes available non-sensitive, anonymous user data that can be used for data mining purposes by customers. This data is stored in a read-only in-memory database consisting of one million equal-sized blocks of 32KB each. The blocks map to the following non-overlapping sets: 500 descriptor blocks, 1,000 tag blocks, and 5,000 index blocks, with everything else being data blocks. Database code executing on clients can directly access these blocks over the Internet.

A client implements a cache of size 10,000 blocks. Accessing a block takes 5 ms on a cache hit, and 200 ms on a cache miss. In a typical database operation at the client, 5% of the accesses are uniformly distributed over the descriptor blocks, 5% of the accesses are uniformly distributed over the tag blocks, 10% of the accesses are uniformly distributed over the index blocks and the remaining 80% of the accesses are uniformly distributed over the data blocks.

For the following questions, you may need to make simplifying assumptions. Be sure to clearly state them, and explain why they are reasonable assumptions.

A. How would you best use the available cache space?
(*Hint: You may want to combine static and dynamic policies.*)

B. The cache starts out completely cold. What is the point at which you would consider the cache fully warmed up?
(Hint: a precise qualitative answer is acceptable.)

C. Once the cache has been warmed up (as defined in your answer to B), what is the average time for a typical client access?

Question 3 (15 points)

The *Alternative Facts Journal (AFJ)* has become one of the nation's largest sources of news and current information. On a typical day, news stories (entirely new, or updated versions of existing stories) come out at an average rate of once every 15 minutes. AFJ's popularity, nationwide distribution span, and (mostly) static content make it a perfect use case for a CDN. Your CDN startup is trying to get AFJ's business.

A. You are trying to convince AFJ management of the value of using a CDN. What two advantages of using a CDN (from AFJ's viewpoint) would you stress? A skeptic of CDNs on the AFJ management wants to derail your efforts. What is one potential shortcoming of a CDN that he could point out?

B. Once you succeed in winning AFJ's business, you have to design a cache consistency mechanism for their expected workload. You have 40 CDN sites, each of which receives an average number of 300 requests per second to fetch some news article. You first consider using check-on-use as the cache consistency mechanism — in other words, before handing out a cache copy of an article to a user, the CDN site checks with AFJ to make sure that the cache copy is up to date. Suppose it costs the AFJ IT infrastructure 0.01 cent to handle such a check-on-use request. What is the minimum annual IT budget of AFJ?

C. Terrified by the answer to B, AFJ's management begs you to reduce their IT expense. You therefore decide to use a callback-based mechanism in which a callback break message from AFJ pushes the bits of new articles (or the new version of an existing article). You negotiate a payment by AFJ of one cent per interaction with a CDN site. What is the new minimum annual IT budget of AFJ?

Question 4 (15 points)

A server stores key-value pairs, where the value is an integer and the key is a unique identifier provided by the client which supplies that value. There are two RPC operations:

- `write(key k, int v)` changes the data object identified by key *k* to have new value *v*. If key *k* doesn't already exist, it creates a new object with value *v*.
- `read(key k)` returns the value of the object identified by key *k* if it exits.

Each client can go through one of 3 caching proxies: A, B, and C. For example, calling `B.write(K1, 4)` means write via proxy B. Assume each proxy has a LRU cache that is able to fit exactly 3 entries. For the following time sequence of operations, answer each of the questions below. Show your working.

Time:	***Operation***
01:	`A.write(K1, 1)`
02:	`B.write(K2, 1)`
03:	`C.read(K1)`
04:	`B.write(K1, 0)`
05:	`A.write(K1, 1)`
06:	`A.write(K2, 4)`
07:	`A.write(K3, 2)`
08:	`C.write(K3, 7)`
09:	`B.read(K1)`
10:	`C.write(K4, 10)`
11:	`B.read(K3)`
12:	`C.write(K2, 0)`
13:	`A.read(K4)`
14:	`A.write(K5, 2)`
15:	`C.read(K2)`

A. Suppose a callback-with-new-value mechanism for cache consistency is used (i.e., invalidation plus new value is provided by a callback break). What does the cache of A, B and C look like after t=4? What about after t=12?

B. Instead of callback, suppose the cache uses “check on use”? What will the contents of A, B, and C’s cache be after t=4? What about after t=14?

C. Suppose faith-based caching is used, with cache entries being assumed valid for 10 time units without checking. When is the earliest that one-copy semantics is violated in the above operation sequence?

Question 5 (12 points)

Consider a Linux uniprocessor system that is managing an I/O buffer cache of 100 4KB blocks using the ARC (Adaptive Replacement Cache) algorithm. The total storage size is 4 GB, so there are 2^{20} total blocks (roughly one million) in the system. Answer the questions below by drawing the state of the ARC cache, using the notation from class. We expect to see lists T_1, B_1, T_2 and B_2 clearly identified in your illustrations, and their contents labeled if known.

(Hint: using the notation from class, the quantity c is 100 here.)

A. Initially, there is just a single process P_1 in the system. It executes in a tight loop whose working set is just 8 blocks. For simplicity, you can assume that these are blocks 0, 1, 2, … 7. Show the state of the ARC cache after one million iterations of the loop.

B. Suppose a second process P_2 is launched at the same time as P_1. Starting at block 100, this process sequentially scans all the blocks up to 100,000. The interleaving pattern of P_1 and P_2 will clearly influence the cache contents. Suppose Murphy influences the scheduling of P_1 and P_2 to make the cache state as bad as possible for P_1. Under these conditions, draw the state of the ARC cache after P_1 has completed 10 iterations.

C. For question B, suppose P_2 was launched after P_1 completes one million iterations. P_2’s access pattern is the same as before, and Murphy tries just as hard to make the cache state as bad as possible for P_1. Under these conditions, draw the state of the ARC cache after P_1 has completed 10 iterations beyond the launch of P_2.

Question 6 (20 points)

Consider a LRU page cache that receives the following reference stream (the page number of each reference is shown):

27, 12, 15, 15, 15, 27, 34, 15, 12, 27, 34, 15, 15, 15, 12, 8

Suppose the cache size is 3 pages, and the cache is initially empty.

A. How many misses does the cache experience for this reference stream? Show your working.

B. Which are the pages left in the cache at the end of this reference stream?

C. If an optimal replacement policy (OPT) is used instead of LRU, how many misses will there be and what will be the cache contents at the end? Explain your answer.

D. Give a recurring reference stream for which LRU is suboptimal with a cache size of 3 pages, but is optimal with a cache size of 4 pages.

Question 7 (8 points)

Consider a file server that implements lease-based cache consistency. When the server becomes heavily loaded, would you suggest decreasing, increasing, or keeping intact the length of new leases it hands out to clients? Explain your answer and discuss the effects of this strategy on the server and clients.