Carnegie Mellon University Qatar

Distributed Systems - Fall 2019

Problem Set (5)

Out: November 20, 2019

Due: December 5, 2019

15pts 1. The *Alternative Facts Journal (AFJ)* has become one of the international'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, international 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?

15pts 2. Impressed by your performance in 15-440 at CMUQ, 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) & video editing (read-write). Video files range from 50MB to 150MB in size, with a mean of 100MB. About 90% of the opens are for video reviewing, and about 10% are for video editing. A typical review session takes 30 min, while a typical editing session only takes 10 min. 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 100ms.

   For the new DropBox product, you architect a write-through whole-file caching system. In particular, on a close after a video editing session, the entire contents of the just-edited file are transmitted to the server. You use read & 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 & 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 a 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?

15pts 3. 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* does not 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	Topic 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?

Handout continues on the next page(s)

30pts 4. You have been nominated as the president and official web guru of `DundeeAndMe`.net, an unofficial fansite paying tribute to Paul Hogan and all things Crocodile Dundee. Due to the incredible growth of the site's popularity, you realized that having just a single web server can no longer handle the load. Hence, you decided to replicate the server, placing three replicas in different Australian cities (where the majority of the fan base resides), a single replica in southern Germany (where there is a small, but devoted cult), and a fifth replica in Boca Raton, Florida.

a. Assume you decided to use 2PC among the replicas for performing updates on site contents. List two advantages of using 2PC in this situation.

b. A freak tsunami destroyed the physical termination point of the main undersea cable connecting Australia and Asia. The Internet was effectively partitioned, separating Australia from the rest of the universe. Explain how this will affect users in Australia and in Germany, both when browsing the site as well as creating new blog entries.

c. Suppose you decided to use a Gifford-based voting scheme instead of 2PC to manage the replicas. In addition, assume a write quorum of 3 and an equal voting weight of 1 among all servers. Under the same partition conditions of part B, explain how users in Australia and in Germany will be affected, both when browsing the site as well as creating new blog entries.

d. Suppose you still wanted to employ Gifford voting with an equal weight of 1 across all servers, but now with a read quorum of 2. Again, under the same partition conditions of part B, explain how users in Australia and in Germany will be affected when browsing the site and creating new blog entries.

e. After some long daunting days, the undersea cable was finally restored, putting an end to this unpleasant network partition. During this time, however, the fortunate users who were able to continue using your site made several updates to the site's contents, which of course were not reflected on all replicas. Assuming a Gifford-based replica management scheme like the one discussed in part C, what recovery steps (if any) should you apply in order to ensure proper functioning of the replica set once network connectivity is restored?

25pts 5. CNP bank uses a group of servers to coordinate and synchronize critical banking transactions. In particular, a Paxos-style protocol is employed on all the servers so as to achieve consensus on each operation. For any submitted operation, let $I \rightarrow J$ denote a successfully delivered message from server I to sever J. Recall the 4 types of messages that are used in Paxos, which can be written as follows:

- `Prepare(n)`, where `n` is a unique sequence (or round) number
- `Promise(n, (`n_k`, `a_k`))`, where n_k and a_k are the last sequence number and value, respectively accepted by Acceptor `k` (if any) and stored at its stable storage
- `PleaseAccept(n, v)`, where `v = `a_k of highest n_k seen among the promise responses, or any value if no promise response contained a past accepted proposal
- `Accept_OK()`

a. Assume 3 servers, S1, S2, and S3 are involved, and they all start out with no past accepted proposals (i.e., no sequence numbers and values are stored at their stable storages). A server can act as a Proposer, an Acceptor, or both. Suppose also that servers S1 and S2 submit proposals to quorums of Acceptors as shown in the Paxos communication trace below. What are the quorum sizes of S1 and S2, assuming that both will not submit prepare messages other than what is shown in the trace? Also, which proposal (or proposals) will achieve consensus? Explain your reasoning.

S1 → S1:	`Prepare(101)`
S1 → S1:	`Promise(101, null)`
S1 → S2:	`Prepare(101)`
S1 → S3:	`Prepare(101)`
S2 → S1:	`Promise(101, null)`
S2 → S3:	`Prepare(102)`
S3 → S2:	`Promise(102, null)`
S2 → S1:	`Prepare(102)`
	...

Handout continues on the next page(s)

b. Assume now 5 servers, S1, S2, S3, S4, and S5 are involved, and also starting out with no past accepted proposals. As in part A, a server can act as a Proposer, an Acceptor, or both. Suppose also that servers S1 and S4 submit proposals to quorums of Acceptors as shown in the Paxos communication trace below. Which proposal (or proposals) will achieve a consensus in this case, assuming the [EVENT!] in the trace is S1 crashing? Explain your reasoning.

S1 → S1:	Prepare(101)
S1 → S1:	Promise(101, null)
S1 → S2:	Prepare(101)
S2 → S1:	Promise(101, null)
S4 → S4:	Prepare(104)
S4 → S5:	Prepare(104)
S4 → S4:	Promise(104, null)
S5 → S4:	Promise(104, null)
S1 → S3:	Prepare(101)
S3 → S1:	Promise(101, null)
S1 → S1:	PleaseAccept(101, X)
S1 → S1:	Accept_OK()
S4 → S1:	prepare(104)
S1 → S2:	PleaseAccept(101, X)
S1 → S3:	PleaseAccept(101, X)
	[EVENT!]
	...