15-440/640 Distributed Systems
Homework 4 (Optional)

Due: December 10, In Class

Name:
Andrew: ID

November 17, 2015

1. You've recently been hired as the distributed systems guru at a new game development company. The company's flagship game, *Luke Flukem Whoever*, is supposed to be a massively multiplayer first-person-shooter game. The online world is expected to consist of thousands of players distributed around the world, and because the company is eliminating dedicated servers to save money, you are responsible for designing the peer-to-peer system for online play.

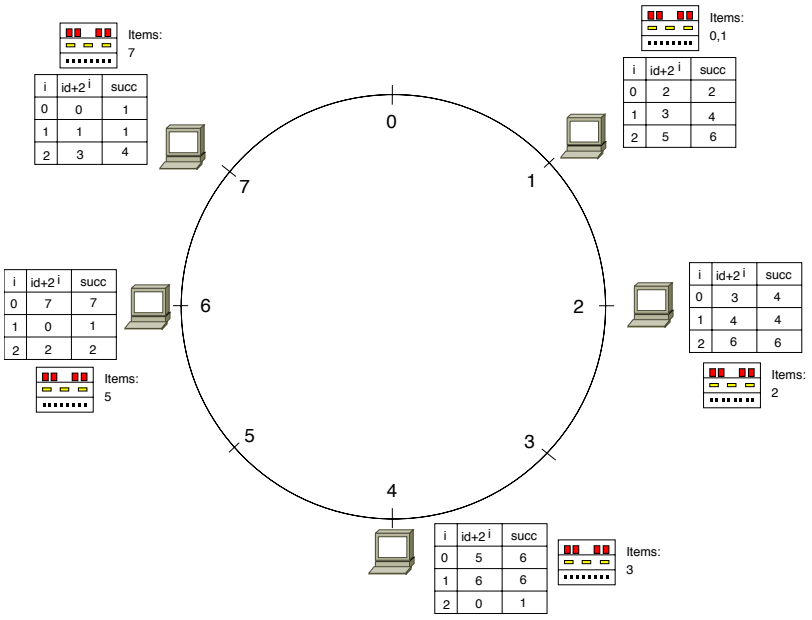
   (a) The game's initial design is to store data, such as other players and in-game objects, in a peer-to-peer system that a player will query dynamically. Your manager hears about a cool new idea called Chord, and suggests that you look into it, arguing that it provides a distributed hash lookup primitive and is robust to frequent node arrivals and departures. Your manager argues that you can store in-game objects on peers (with replication to handle node departures) using Chord for scalable lookup of these in-game objects.

   After giving it some thought, you decide that this is a poor solution for a first-person-shooter game where the peer-to-peer system may have thousands of geographically distant players in it. Explain briefly the rationale for your decision.

   (b) The publisher, Eccentric Arts, decides that the game release date needs to be pushed up by 3 months. Consequently, the game design slightly changes, and instead of all the players being in one very large world, players only connect with players in the same geographic city. You now decide that Chord is a reasonable design choice for storing certain static in-game objects on peers.

   You implement this system with ease and begin a small scale deployment consisting of only four peers. The peers contain the listed items (e.g., *Node 4* only has *Item 3*), and have successor tables as shown (the $id + 2^i$ column is there to remind you how the successor table is set up).

   List the nodes that will receive a query from *node 1* for *item 7*.



(c) List the nodes that will receive a query from *node 2* for *item 5*.

(d) During larger-scale testing, you notice that the popularity of objects in your game is skewed, and that a small set of peers get overloaded with requests for the most popular items. Like any good distributed systems student, you pull out a valuable tool from your distributed systems belt to reduce this load: caching.

Your manager likes this idea, and suggests that once an object is found, it should be propagated back down the path in the Chord ring taken to lookup the object on the forward path, with the item cached at each node along the way. Your manager argues that this is an effective choice of nodes to replicate on because the nodes caching the object in this path will never have seen this object before. Why is this true?

2. (a) Provide three reasons a company might prefer to pay Akamai to host their webpage instead of putting it onto a peer-to-peer network (such as Napster) for free.

(b) We saw no examples of of chunk-based peer-to-peer networks that use flooding. What would make such a network inefficient?

(c) Chord was designed to be a distributed, wide-area hashtable. Why does Chord use consistent hashing to map data items to nodes in the key-space?

(d) Name an advantage to using a centralized p2p network.

3. (a) Describe the difference between Type I (Hypervisor) and Type II (Hosted) VMM? Try to answer in 4-5 lines.

(b) Explain the difference between process VMs, Hosted hypervisors (XEN PV) and bare metal hypervisors.

4. (a) How does Big table and Spanner ensure consistency?

(b) What things among CAP (Consistency, Availability, Partition Tolerance) does each of Big Table and Spanner support? Also explain in brief how they do so.

5. A professor in some other class wants the students in her class to collaborate on solving an exam.

Some of the students in the class are faulty: They either do not understand the material, or will deliberately lie to other students to try to make the others fail.

The professor asks the students to form a practical Byzantine fault tolerant system to solve the exam. The professor ("the client") sends the exam out to all of the students. The students have designated one student to be the organizer who then directs the other students as per the BFT protocol (Assume that this student does properly forward the request.)

(a) If all 82 students are participating, what is the maximum number of students who can be wrong or malicious?

(b) The professor/client is watching the exams being handed in (replies to the client). After how many students' matching committed replies will she have to observe before knowing that any other non-malicious student will hand in the same answers?

(c) Why couldn't a Paxos approach help here? (1–2 sentences).

6. The Byzantine empire has come up with a clever way for its generals to decide on the same plan of action (to *attack* or *retreat*) when they reach an enemy city. Before making the decision, each general must determine the total number of friendly troops that the participating Byzantine generals have in their armies. Specifically, the protocol they use to come to a decision is as follows: each general tells the other generals their troop count, and if the total number of friendly troops is $\geq$ some threshold T, the general attacks; otherwise, the general retreats. The goal is to make sure that all generals take the same action.

   You can assume communication is synchronous and unicast, and the delay is bounded. Every message that is sent is delivered correctly and the receiver knows who sent it.

   (a) In one particular scenario, three Byzantine generals, A, B and C, arrive at a city. General B is malicious (in byzantine fault tolerant way), and the other two are honest. Describe a scenario where the honest generals do not decide on the same plan.

   (b) What changes are necessary to the above scenario and protocol to deal with the single malicious general? Name two changes. Assume that the generals have no way to "sign" messages.

7. Srini and Yuvraj need to communicate to decide which TA is going to grade the next homework. They have a shared secret key, K_{Profs} that allows them to create unforgettable message authentication codes (MAC) so that Srini can verify that Yuvraj did in fact create any message that is received.

   Srini and Yuvraj have a simple protocol: Srini sends a "*Who grades HWX?*" message to Yuvraj in plain text, and Yuvraj replies with one of two messages: $M1 = MAC_{K_{Profs}}$("Arjun"), or $M2 = MAC_{K_{Profs}}$("Varun"). When Srini receives either M1 or M2, he verifies the MAC using K_{Profs} and knows who will grade the next homework.

   (a) This protocol is insecure. A malicious TA on a router between Srini and Yuvraj might be able to avoid ever having to grade a homework! In one sentence, describe the attack.

   (b) What simple change to the above protocol could defend against this attack?

8. Srini and Yuvraj are arguing over a telephone about who will teach the next 15-440 lecture. They decide to settle this fairly: by "flipping a coin" (i.e., choosing a random bit) over the phone. Yuvraj suggests that he can flip a coin and just tell Srini the answer, but Srini does not trust Yuvraj to flip the coin and reveal the result honestly. In this problem you will develop a protocol that uses the MD5 cryptographic hash function to prevent cheating.

   (a) Yuvraj suggests the following protocol: Srini and Yuvraj each choose their random bits x_c, x_d (x_c and x_d are each 1 bit), and Srini tells x_c to Yuvraj. Yuvraj then tells Srini x_d, and they both compute $x_c \oplus x_d$ to determine the answer. Argue that this protocol is correct if Yuvraj sends a truly random x_d, but that Yuvraj can cheat to manipulate the outcome.

   (b) Devious Yuvraj tries again: Srini and Yuvraj choose x_c and x_d as before, and Srini uses the MD5 cryptographic hash function to compute $h_c = \text{MD5}(x_c)$, which he sends to Yuvraj. Yuvraj then sends $h_d = \text{MD5}(x_d)$ to Srini. They now both reveal x_c, x_d (in any order) and again compute $x_c \oplus x_d$. Srini can use h_d to confirm that Yuvraj did not change x_d after sending h_d, and Yuvraj

can similarly confirm that Srini did not change x_c after sending h_c. Explain how Yuvraj can still cheat to manipulate the outcome.

(c) Improve the protocol from (b) to prevent either Yuvraj or Srini from cheating. Your protocol should use only MD5, as well as the ability to send messages between Yuvraj and Srini (and vice-versa) and the ability to generate random numbers.