15-744 Computer Networks (Fall 2010)
Homework 3

Due: Nov. 8th, 2010, 3:00PM (in class)

Name:
Andrew ID:

November 16, 2010

A Using 'dig' to Understand DNS

1. In this question you will use the unix utility 'dig' to explore the contents of DNS messages. Please use dig on unix.andrew.cmu.edu.

   The format of a dig request issimple. Just type: *dig www.princeton.edu* to perform a look-up for that DNS name. As you now know, DNS requests can do more than just ask for the IP address corresponding to a single DNS name. Type *dig princeton.edu ANY* to see DNS records of all types that are associated with the domain 'princeton.edu'.

   (a) What IP address did the computer you are logged into contact to make the DNS request? Where do you think this server is located?

   (b) List all of the different types of records received as a result of your query. For each record, explain its purpose, using one of the entries provided in the reply as a concrete example.

   (c) Note that some of the names in the reply are not in the domain 'princeton.edu'. Use the DNS names and/or 'traceroute' to find the general location of one of these servers. Where is it? Given the type of record, why would Princeton do this?

   (d) Use dig to find the names of two non-local servers you *could* contact in the process of identifying the nameserver for the domain 'cnn.com' (assume no DNS information is cached anywhere).

   (e) Use dig to find the TTL for the DNS mappings of 'www.cnn.com' and 'www.cs.stanford.edu'. What are they? If your boss asks you to provide two positive and two negative effects of having a short DNS TTL for the company's e-commerce site, what would you say?

Solution:

Note: These answers are correct as of 9/22/06.

(a) The IP address is 128.2.1.13. This is the local DNS server for unix.andrew.cmu.edu and is located on CMU campus.

(b) The following records exist:
A : maps from a DNS name to an IPV4 address. In this example, ns1.fast.net is 209.92.1.12 .
NS : indicates that a DNS name is an authoritarian nameserver for the specified domain. In this example, dns.princeton.edu is a name-server for the princeton.edu domain.
MX : provides information about the SMTP servers accepting inbound mail for the domain.

emfw1.princeton.edu is such a server for princeton.edu .
AAAA : maps from a DNS name to an IPV6 address. ns3.nic.fr has address 2001:660:3006:1::1:1 .

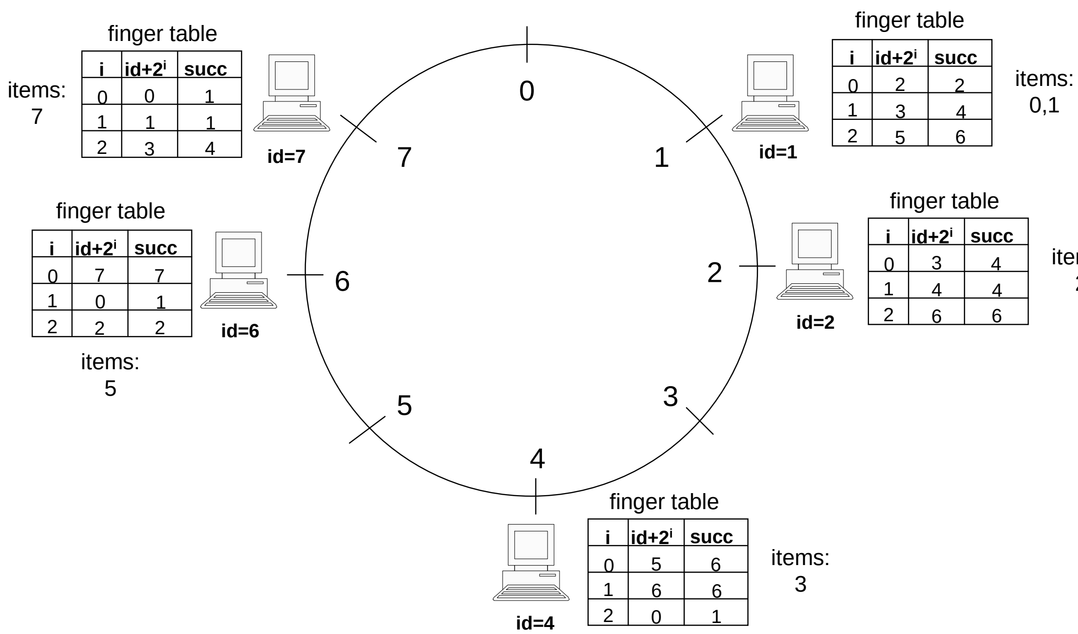
(c) Some of the name servers are in significantly different geographic locations. For example, ns1.ucsc.edu is in Santa Cruz, CA and ns3.nic.fr is somewhere in France. Princeton does this so that if something in the princeton area would happen to take out their resolvers, there would be back-ups in very diverse locations that would still be able to answer queries for *.princeton.edu .

(d) To get the DNS name for 'www.cnn.com' with no previous caching, you would have to contact a root DNS server and a .com DNS DNS server. To find a root server type *dig .* and to find a com TLD server, type *dig com.* . Examples are A.ROOT-SERVERS.NET and a.gtld-servers.net respectively.

(e) TTL answers will vary depending on how caching. Possible answers are: 300 seconds for www.cnn.com, and for www.cs.stanford.edu .
Reasons for using a short TTL: Allows faster fail-over if a particular machine goes down. Also give more flexibility in load-balancing.
Reasons against using a short TTL: More load on your authoritative name servers and higher perceived latency by clients performing look-ups, both do to reduced caching.

B DHTs

2. Dave, in fear that the RIAA will shut down his centralized P2P server (like Napster), sets up a Chord DHT for lookups and routing in his peer to peer network. Unfortunately (or fortunately, for you), Dave's P2P network is not very popular and only consists of five peers at the moment with finger tables and items illustrated below. For example, *node 4* has *item 3*.



(a) List the nodes that will receive a query from *node 2* for *item 0*.

Solution: 6 points. $2 \to 6 \to 7 \to 1$. Node 6 can not contact node 1 directly (even though it has a finger to node 1) because $1 \notin (6, 0]$. Chord finds the *predecessor* of 0 first and then finds the item on the immediate successor of the predecessor. This is to ensure that we don't "skip over" the successor when our finger table is stale.

(b) Suppose node 4 crashes. *node 7* queries for *item 5*. List the nodes that will receive this query, assuming the the tables have had time to converge after noticing that node 4 has left.

Solution: 6 points. $7 \rightarrow 1 \rightarrow 2 \rightarrow 6$

C DNS Redirection

Harry Bovik is working on a web site that has multiple replicated servers located throughout the Internet. He plans on using DNS to help direct clients to their nearest server replica. He comes up with a hierarchical scheme. Harry has divided his server replicas into three groups (east, west and central) based on their physical location. A typical query occurs as follows:

- When a client makes a query for `www.distributed.hb.com`, the root and `.com` name servers are contacted first. It returns the name server (NS) record for `ns1.hb.com`. The TTL of this record is set to 1 day.
- The `ns1.hb.com` name server is then queried for the address. It examines the source of the name query and returns a NS record for one of {`east-ns, central-ns, west-ns`}`.distributed.com`. The choice of which name server is based on where `ns1` thinks the query came from.
- Finally, one of {`east-ns, central-ns, west-ns`}`.distributed.com`. is contacted and it returns an address (A) record for the most lightly loaded server in its region.

Answer the following 3 questions based on this design.

3. Harry's name server software has only two choices for TTL settings for A and NS records - 1 day and 1 minute. Harry chooses the following TTLs for each record below:

   1. NS record for {`east-ns, central-ns, west-ns`}`.distributed.com` - 1 day TTL.
   2. A record for {`east-ns, central-ns, west-ns`}`.distributed.com` - 1 day TTL.
   3. A record returned for the actual Web server - 1 minute TTL.

   Briefly explain why Harry's choices are reasonable, or why you would have made different choices.

   Solution: 4 points. The name server for a client is based on the region and hence probably does not change very often. Therefore, Harry sets the NS and A records for the name server address to 1 day.

   Harry wants the name server to direct clients to lightly loaded web servers. To do this, Harry must be able to control which web servers each client goes to. Therefore, Harry sets the TTL of the web server A record to 1 minute, so that clients won't cache the record for very long and will ask the name server which web server to use for subsequent requests. If the TTL was 1 day, each client would cache the first A record it got and then continue using the same web server for the entire day even if it becomes overloaded.

4. In general, name resolution systems map names based on the name and context. In this particular case, what are ***TWO*** items of context that the name resolution uses?

> **Solution: 5pts**
>
> 1) the IP address of the local name server
>
> 2) the load on the servers in the region

5. Harry's Web site is especially popular among CMU students. The CMU network administrator estimates that there is one access from CMU every 3 minutes. Each access results in the application resolving the name `www.distributed.hb.com`. Assume the following:

   - No other DNS queries are made in CMU
   - All CMU clients use the same local name server.
   - This local name server is mapped to the `east-ns` region.
   - Web browsers do not do any caching on their own.

   How many accesses per hour will be made to the following name servers to resolve these CMU queries? Explain your calculation.

   1. The Root Servers
   2. `ns1.hb.com`
   3. `east-ns.distributed.com`

> **Solution: 5 points.**
>
> 1. The Root Servers - 1/24 requests/hour
> 2. `ns1.hb.com` - 1/24 requests/hour
> 3. `east-ns.distributed.com` - 20 requests/hour

D Sensor Networks

You and a friend have a startup that designs and sells wireless sensor networks. The job of the sensors is to collect raw information (temperature, etc) and record **events** that happened. One example of such event is elephant sighting. External users are only interested in these events, but not the raw information. There is one access point through which the external users can issue queries to collect all events.

Since you attended 15-744, so you know how the original Directed Diffusion works:

- When a sensor S detects an event, it stores the event locally.
- When the external user issued a query through access point, the query is flooded to all sensors. At the same time, gradients are set up pointing back towards the access point.
- For each sensor node, when receiving the query, it sends the recorded events to the access point through the gradients set up during the flooding stage.

Now being a smart CMU student, you may think of an alternative design, called Data-Centric Storage (DCS), which works as follows:

- For any particular event e, one single sensor (S_e) is responsible for storing all happenings.
- Thus when an event e is detected by a sensor S, it sends this event to S_e through geographic routing.
- When the external user issued a query, it uses geographic routing to access all the happenings of event e from S_e.

For this problem, we assume there are n sensors equipped to detect various types of events. There are a total of D events happened and Q queries issued. Assume in the network, both gradient routing and geographic routing incur a cost, i.e., number of messages, of $\sqrt{n}$ on average (and there is no extra cost for geographic routing set up).

Also, you do not need to fully understand the geographic routing for this problem, but it works as follows:

- Imagine the whole sensor network as a DHT.
- For sensor networks, it does not make sense to use consistent hashing because it does not preserve locality. For example, Chord routing goes through $O(\log n)$ intermediate nodes, which can be located at various different places in the sensor network, which can cause the message to be sent back and forth.
- Thus instead of using consistent hashing, sensor networks can use geographic hash tables (GHT) that map each sensor to an ID according to its geographic location.
- Then routing is performed on the ID space as in DHTs, which in the case of GHT is essentially geographic routing, which takes $O(\sqrt{n})$ hops.

6. What is the approximate number of messages for the two approaches, i.e., Directed Diffusion and DCS? You can assume that each event happening requires a message (no aggregation).

Solution:

For Directed Diffusion with local storage: $Q * n + D * \sqrt{n}$.

For DCS: $Q * \sqrt{n} + D * \sqrt{n} + D * \sqrt{n}$.

7. Explain when you should choose one over another approach?

Solution:

DCS is preferrable when n is large, and D is small compared with Q (events happen very often but queries are infrequent). Otherwise local storage is preferrable.