CS 594 IP Assignment 2

Submitted By : Shrikrishna Bhat

1) a) Ben's architecture 85899.346 seconds

Alyssa's architecture - 429496.73 seconds

Ben's company has a minimum file distribution time.

Shrikrishna Bhat

1.i) Let us consider

Server upload Bandwidth = us

Peer upload Bandwidth = up

Peer downlod Bandwidth = dp

File size = F

$F = 1GB = 1 \times (1024)^3 \times 8$ bits

$us = 100Gbps = 100 \times 10^9 = 10^{11}$ bps

$up = 100Kbps = 100 \times 10^3 = 10^5$ bps

$dp = 1Mbps = 10^6$ bps

Ben's Architecture

No of users = N = 10

Time taken to distribute files to user = tcs

$$tcs = MAX[NF/us, F/min(dp)]$$

$$= MAX[10^6 \times 1024^3 \times 8/10^{11}, (1024 \times 8)/10^6]$$

$$= MAX[85899.34592, 8589.934]$$

$$= \underline{\underline{85899.346 \text{ seconds}}}$$

Alyssa's Architecture

Total Users = $N = 10 \times 10^6 = 10^7$

Active Uploaders = $A = 10\%$ of $10^7 = 10^6$

Time to distribute files = t_{pf}

$$t_{pf} = \text{MAX}\left[F/u_s, F/\min(d_p), NF/(u_s + \Sigma u_p)\right]$$

$$= \text{MAX}\left[(1024^3 \times 8)/10^{11}, (1024^3 \times 8)/10^6, (10^7 \times 8 \times 1024^3)(10^{11} + (10^6 \times 10^5))\right]$$

$$= \text{MAX}\left[0.0859, 8589.934, 429496.73\right]$$

$$= \underline{\underline{429496.73s}}$$

$\therefore$ Ben's company has min. file distribution time with client-server architecture

1b) Min distribution time with no free riders - 78090.3144 seconds.

ii) In Alyssa's P-2-P, 10% of users are uploaders & others are free riders
If no free riders total uploaders are 100%

$N = A = 10^7$

$t_{pf} = \text{MAX}[F/u_s, F/\min(d_p), NF/(u_s + \Sigma(u_p))]$

$= \text{MAX}[F/u_s, F/\min(d_p), NF/(u_s + A \times u_p)]$

$= \text{MAX}[1024^3 \times 8/10^{11}, 1024^3 \times 8/10^6, (10^7 \times 8 \times 1024^3)/10^{11} + (10^7 \times 10^5)]$

$= \text{MAX}[0.0859, 8589.934, 78090.3144]$

$= \underline{\underline{78090.3144 \text{ seconds}}}$

2) Distributed Hash Table is a decentralized storage system that provides a lookup and storage scheme similar to a hash table storing the key and value pairs. The main advantage of DHT is that nodes can be added and removed efficiently by re-distributing the keys. DHT provides a database service where peers can query the value with a given key. The insertion and lookup process in DHT is done in constant time. Each Node in DHT is responsible for the keys and has associated values.
In Alyssa's case, she is using a centralized look-up server that contains all the data. The clients of her company connect to the same server and due to multiple requests by users at the same given time, the central server cannot handle large amounts of requests and will fail at some point in time.
The best way to handle this dilemma is using a DHT. In DHT data is decentralized.
In the decentralized server, each peer will store some data and these peers act as servers for other peers.
Here the network traffic is distributed and there is less chance of the server failing. As the number of users increases, they will act as peers and store some amount of information. One of the key aspects of this server is that, if there is a failure in any of the peers, it won't affect the whole system. Every data entered will have a hash value that corresponds to the key and the value is stored for each key entry. In this system, users are the keys and values are the information about the neighboring nodes

Each peer in the system will store information about its neighboring peers. The addition and deletion of peers are maintained by adding and deleting nodes in the DHT system. The major aspects of DHT are that it is a decentralized server, DHT can be scalable and it is fault tolerant. Similar to hash tables the average Lookup Time in DHT is O(logn). Since each peer has a key, finding the value corresponding to the key is not a tedious task.

Bootstrapping is a process of information sharing to the newly joined nodes such that they are successfully connected to the existing network. A new node can learn about other DHT-capable peers through its interaction with them. A peer's support for DHT is advertised in its Handshake. Once a client discovers at least one good, well-connected DHT peer, it can navigate the DHT to find more and closer DHT peers. It will remember these peers, called nodes in DHT-speak.

3) There are two types of DNS servers. According to Alyssa, DNS works by recursive calls and Ben says DNS has the understanding of the Authoritative DNS server.

A Recursive DNS server is where one DNS server communicates with several other DNS servers to get the IP address and return it to the client. This is in contrast to an iterative DNS query, where the client communicates directly with each DNS server involved in the lookup. When a client enters a web address, it is first sent to a recursive DNS server. Many recursive servers are managed by the ISPs and they know where to look for the requested information. The query is forwarded by calling multiple servers until the server which the client requests is found.
In comparison to this, Authoritative DNS servers just that they have the authority over DNS records for a domain. While the recursive DNS servers are for discovering the information for the domain, authoritative DNS servers have the actual information, as stated they maintain the records for a particular domain. When a client enters an IP address, it is sent to the local ISP which has a recursive server that might contain the information cached in the memory. When data gets outdated, a recursive server needs to find the IP elsewhere, it will then go to other recursive servers but if it still doesn't find it goes to an authoritative server. The data it receives in response is usually cached.
The advantages of Recursive servers are: It resolves the domain name to an IP address that handles network traffic. Since it is defined in local domain space, these are flexible and scalable. Local caching mechanisms are provided by them.
Disadvantages of Recursive servers are: If a cache or authoritative server sends a wrong result, it will take some time for it to recover because it has to make new queries. If many users are querying at different websites, there can be a large number of queries that can be difficult to handle. If an authoritative server is offline, there is a chance that a timeout error is received.
Advantages of Authoritative servers: It is very fast in responding to queries for a particular zone. It will not respond to recursive queries.
Disadvantages of Authoritative: It can be prone to hacking and other vulnerabilities.

Both Authoritative and Recursive servers have crucial functions and they depend on each other to fulfill their purposes. Authoritative DNS servers store the most recent and accurate information and can provide a final answer to a user's DNS queries. On the contrary, Recursive servers only keep a copy of DNS information for a particular amount of time which is their Time To Live (TTL) and they have to obtain information from other servers.
Since servers are chosen and built according to the requirements of the end users. Both servers have pros and cons and they can be chosen for a particular purpose. Recursive DNS is used by most end users since that is what is provided by the ISP. Authoritative DNS is used by small and Enterprise Level Businesses and people who have their domain. Since both have pros and cons, the servers can be chosen according to the needs of Ben or Alyssa for a particular purpose.

4a) Total Average Response time - 2.01lll seconds

4a) Access Link = $R = 10$ Mbps $= 10^7$ bps

Average object size $= L = 10^5$ bits

Average request = 10 request/s

Average Internet delay = 2s

Time to transmit $= L/R$

$$\text{Average time} = \frac{\text{Avg. object size}}{R}$$

$$\text{Average time} = \Delta = \frac{10^5}{10^7} = 10^{-2}$$

Traffic Intensity $= \Delta\beta = 10 \times 10^{-2} = 0.1$

Average Access delay $= \Delta/(1-\Delta\beta)$

$= 0.01/(1-0.1)$

$= 0.01111$

Total Average Response time $= 0.01111 + 2$

$= \underline{\underline{2.01111}}\,s$

4b) Total average Response time - 1.20625 seconds

4b) Hit rate = 0.4

Avg access delay = $\Delta / (1 - \Delta\beta)$

$= 0.01 / 1 - (0.4)(0.1)$

$= 0.01 \times 0.96$

$= 0.01041$

Cache miss rate = $1 - 0.4 = 0.6$

Avg misstime when cache misses

$= 0.01041 + 2 = 2.01041$

Avg response time when cache hits

$= 0$

Total Avg response time

$= (0.4 \times 0) + (0.6 \times 2.01041)$

$= \underline{\underline{1.20625}}$ s

5) Wireshark labs are in separate files which are attached.