Dev Club Recruitment

Kshitij Alwadhi

March 2020

This pdf contains my answers to the Recruitment Assignment of Dev Club IITD - 2020

1 Brain Warm Up

We can set up a threshold limit in our file system so that we will always have sufficient space to store the zip file of the maximum possible course.zip. Moreover, we can make a script such that, whenever we receive a zip request, that zip file is stored in the server and when the file server storage is reaching that threshold limit **we delete the zip file which is already there on the server but has been requested for the least often.**

This way we will have the best of both worlds where we won't have to buy a lot of storage to save every zip file and also the zip files in demand will always be present in the server so that it can be served ASAP without any delay. We can also set some time limit apart from frequency such that if a file has gotten this old, we delete it.

Moreover to account for new material being added, we can set up a script so that, if the zip file of that particular course is already present, we delete the old one and save a new zip containing the updated material so that it can be served ASAP as well.

2 Development Assignment - Research

2.1 Networking

- SSH is simply a tunnel through which we can access a remote machine or server securely. But problem occurs when we try to connect to a server that is behind a firewall and the firewall rejects any incoming connection request. Let's say I am at home and I want to access my linux instance at IITD. I can't directly ssh into it as the server would straightaway deny the request. This is where reverse SSH tunnelling comes into play. With reverse port forwarding, I can forward a port from my remote machine (the IITD one) to the local machine. This works by assigning a socket to listen to the port on the remote side and whenever a connection is made to this port, the connection is forwarded over the secure channel and a connection is made between the remote and local machine.

- In computer networking, a port is a communication endpoint. When we locally host a website, the address in the address bar eg. 127.0.0.1: 8000, this 8000 is the port number. A port number is always associated with an IP address. Packets are routed through specific ports for specific purposes.

- Process followed when we type in a URL of a website:

  1. Firstly the address is typed into the address bar of any browser.
  2. The browser will then proceed to check if there is a DNS record of that address in various layers of caches:

     (a) The browser cache

     (b) OS cache

     (c) Router cache

     (d) ISP cache

     Basically what's going on here is that, for accessing a website, what we typically need is an IP address but we can't really remember those numbers for every website we usually visit, instead we remember this string which is then resolved to an IP address by a DNS query. If that query is not resolved in the various caches, the query is then sent to a DNS server of our choice / automatically determined by ISP. The DNS servers usually used are Cloudflaire, Google DNS etc. Once this query is resolved, we now have an IP address.
  3. Now that the browser has the IP address, it will try to establish a TCP connection with the server. This is established using a TCP/IP three way handshake.

     (a) The client sends a SYN packet to the server to ask if it's open for new connections.

     (b) The server acknowledges by sending back a SYN/ACK packet.

     (c) The client will receive the packet and then acknowledge it by sending back an ACK packet.

     Now a TCP connection is established between the server and the client.
  4. The browser will then send a GET request to the server asking for its webpage. The request will also contain additional information such as browser identification etc.
  5. The server contains a webserver like Apache which will receive that request and pass it on to a request handler to generate a response. The request handler reads the requests to see what's being asked for and also updates the information in the database if it were a POST request. Then it will assemble a response in a particular format be it a JSON or an HTML file.
  6. The server response will contain the web page, error codes if any etc.
  7. The browser will first display the barebones HTML file and then send out another GET request asking for additional static files such as images, CSS, JS files and these static files will be cached by the browser so that it can be served immediately the next time one visits the website.

- Python standard library provides a module called SimpleHTTPServer which can be used to set up a server on your local device.

  This will fire up a local server on your machine which can be accessed using 0.0.0.0 or 127.0.0.1 and by default, the port number is 8000. This is a localhost. It will look up for index.html file in the current working path directory from where the server is fired up.

  If we go to 127.0.0.1:8000/ from our browser, it will show us the index.html page.

  But to access this local server from other devices connected on the same network, we need the IP address of the machine on which the local server is set up.

  After we have that, suppose it's 192.168.1.8, we can easily access this page from other devices such by going to the page 192.168.1.8:8000 from other devices (the required permissions for networking and sharing should be activated first).

- NGINX and Apache are two webservers. They control how web users access the hosted files. They receive the requests from a user which is sent by the browser and send back the required files if present otherwise throw a 404 error. They first send back the HTML file followed up by the CSS and JS files.

2.2 Database

Database systems are used for storing data.
For storing the data of this model of Professors, I would prefer an SQL type database as we already know the structure of our database (schema). We would have 2 tables. One containing the list of Professors. And another table with One to Many mapping, containing the projects undertaken by the professor and their respective details. Since we already know the fields we are going to add, an SQL type database is better as compared to NoSQL. Moreover, an SQL type database is vertically scalable, so we can add more RAM, SSD to store more data, whereas, NoSQL type database is horizontally scalable, so we would need more servers to handle extra data.

2.3 Timer Functions in JavaScript

setTimeout() and setTimer() are not a good choice for all the applications. This is because, of the single threaded nature of JavaScript that only one timeout can fire its callback at any given moment. If another javascript is running its course when a timer was scheduled to run, the timers callback function would be added in queue to run AFTER the interpreter is done with the script its already running. This can lead to delays. This is known as Timer Congestion. When multiple scripts are running, this can lead to the UI becoming sluggish, it is possible to set so many timers expiring at about the same time that they just keep queuing up one after another; never giving the browser's UI a moment to update.

This can be solved by keeping track of how long the loop has been running. Before the loop has been running for long enough to affect the UI from being updated, you pause the loop, execute window.setTimeout to update the UI, and then pick the loop back up from we left it. This way the browser gets a chance to update the UI.

This way, the tasks will be pushed back a little bit to account for the UI being updated. In cases of extreme congession, this can lead to a task never being executed at all but this is any day better than an unresponsive browser.

To solve this problem, there is an implementation:

`https://github.com/fitzgen/chronos`

which implements the above idea with very little change in the already written code.

In attacks like meltdown and Spectre, the hacker can access the protected info based on subtle differences in hardware behaviour. It takes into account the time difference between the times taken to extract data that has been cached versus times taken to retrieve from memory. A precise measurement of this time is required for the hack to be successful. To prevent this, browsers are intentionally reducing the JS timer resolution such that these attacks can be stopped and it will still have almost no effect on normal activities.