

Devclub Assignment

Saksham Dhull

March 9, 2019

1 Basic Networking Systems

1.1 Working of websites

Each website contains a set of web pages which are HTML pages and are interpreted by the web browser to view accordingly. Whenever a site loads, a request is sent to the server hosting the website, and the response is subsequently sent back as these web-pages. If for example, multiple source files are required for viewing by the browser, for example, CSS, JS, etc., various requests are sent to the web server via the internet. All the response files are downloaded as temporary files to your browser's memory. There exist two protocols for this information exchange (source files) namely, HTTP and HTTPS with HTTPS being more secure and thus, being used in websites that include secure exchanges, for example, passwords.

1.2 Local Server

A local server is a web server hosted on your computer on a port specified. All the requests made to your IP address on the specified port will be handled by this server which can be controlled using various techniques. For displaying a file to another user, you can accept a get request on your localhost on the port, and the person inside the intranet can access it as by sending the request. Now as the HTML files are not read as such and are executed, so as long as you can execute the HTML file, this method will work.

1.3 Global Connection to Local server

All the computers connected to a single router will be behind a single NAT which maps the various global ports to local ports. You will not be able to connect to the intranet server as the NAT will decline all your requests to the given IP. However, you can accomplish this by UDP hole punching or Relaying. Detail discussion about these methods are beyond the scope of this assignment.

1.4 NGINX/Apache

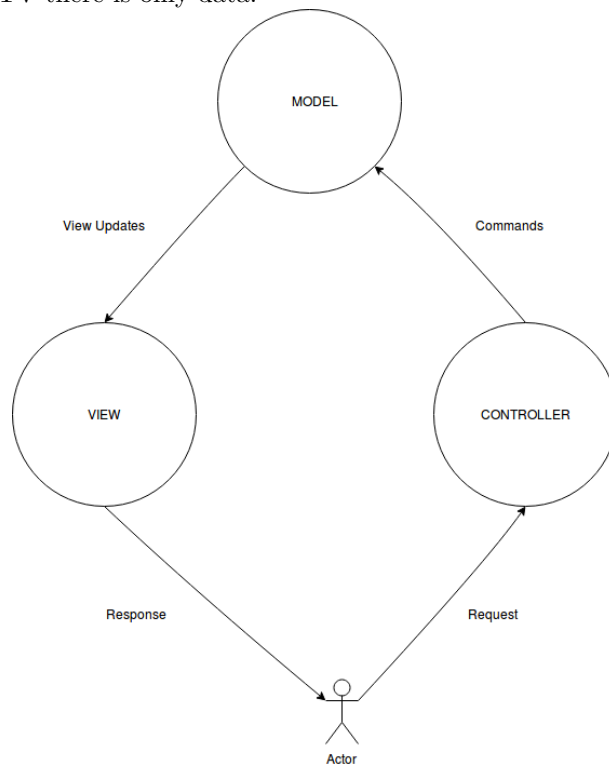
Nginx is a web server that can be deployed to serve dynamic HTTP content on the network using various methods and it can also serve as a software load balancer. Nginx uses an asynchronous event-driven approach, rather than threads, to handle requests. It is frequently placed between clients and a second web

server, to serve as an SSL/TLS terminator or a web accelerator as it efficiently handles tasks that might slow down your web server, such as negotiating SSL/TLS or compressing and caching content to improve performance. Apache is also a similar framework and provides competition to Nginx in terms of speed and efficiency.

2 Web App Architectures

2.1 MVC Architecture

The Controller in MVC architecture inputs the requests from user and converts them into commands to be processed by the Model module and send the appropriate viewing updates to the View module which presents the data to the user as a response. The MVC architecture uses state wise updating whereas in MTV there is only data.



2.2 Various Frameworks

	Ruby on Rails	Django
Architecture	RoR uses MVC architecture as described above	Django is based a MVC architecture but works as an MTV one, i.e., here View module provides what to present and not how to present which is decided in the template module.
Working	RoR follows Convention over configuration ,i.e., it as predefined layouts to use	User is able to define his own layout and configuration settings
DataFlow	In RoR, the whole architecture works on state changes, i.e. upon any change, it goes from one state to another/makes changes in original one.	In Django, there is only data, no states are defined as such and is thus said to break easily.
How to Choose	RoR is less customisable but much easier to implement and get up running within minutes	Django requires a lot more backend but is highly customizable and provides a stronger data processing background.

2.3 MVC *vs* MTV

MVC and MTV are both server side frameworks used to handle requests and provide adequate responses. MVC operates on the principal of states machines, i.e., it has states which are updated as the requests are processed checking all the headers and input and output statuses, whereas MTV is more like data management, it looks at the data only and how will it process it, ignoring the headers and HTTP statuses. In MTV, Views module is used as Controller as it controls the data that will be presented rather than how it will be presented, which is handled by the Template module.

3 Newer Technologies

3.1 AMP

- AMP makes everything that comes from external resources asynchronous, so nothing in the page can block anything from rendering. i.e., all external resources are downloaded asynchronously without hampering the load performance. This also optimizes the font triggers, as the font files are pretty large and hamper performance when requested synchronously.
- Other performance techniques include the sand-boxing of all iframes, the disabling of slow CSS selectors.
- Pre-calculation of the layout of each and every element rather than fixing them dynamically. This is known as static layouting. So only one HTTP

request is required to render the whole page.

- Only a single style sheet is allowed and it should be inline with the whole page and are thought of less than $50kB$ which makes them size bound. Although, very complicated pages can be loaded within $50kB$ but it still might cause a restriction in future.
- All the animations/videos in AMP are GPU rendered. Now AMP can only support css items that can be rendered using GPU making it a restriction for the framework.
- AMP prevents reloading and recalculation of layouts when some element is changed. It rather calculates them first and fits in the new objects.
- AMP prioritizes how the resources will be loaded based upon how likely they are to be seen by the user, for example, the images at the bottom are requested later than those above it.
- AMP utilizes Google AMP cache. It is a proxy-based content delivery network which fetches all valid AMP HTML pages, caches them, and improves page performance automatically. The cache also comes with a built-in validation system which confirms that the page is guaranteed to work, and that it doesn't depend on external resources. Another version of the validator comes bundled with every AMP page. This version can log validation errors directly to the browser's console when the page is rendered
- All the integrations are done on server side rather than client side to improve the performance.

3.2 AMP vs PWA

	AMP	PWA
Introduction	AMP allows developers to build web pages that load near-instantly with a unique coding language. These pages are hosted on a CDN, which delivers a cached version of the page when it's visited by a user.	Progressive web apps are mobile website experiences designed to look, feel, and work the way native mobile apps do. Users encounter them in a browser as they would any mobile website. After engaging with that site, the user is prompted to install the web app on their device. Should they choose to install, the app will download to the device in the way a native app would.
UI and Speed	AMP prohibits a lot of scripts and functions of the pages and the AMP pages are relatively simpler	PWA allows everything and anything you can do with any normal web browser, that's why although it's User Interface is much better, it lacks speed.

References

- [1] Peer-to-Peer Communication Across Network Address Translators
Bryan Ford, Pyda Srisuresh, Dan Kegel