Web App Architecture

Eli the Computer Guy - November 20, 2024
Version 1

Speed and Latency

Speed is how fast an entire block of data can go from the server to the client.

Latency is how fast the packets will go between the two systems.

- A train can transport 100 people 30 miles in 60 minutes.
- A Ferrari can transport 2 people 30 miles in 10 minutes.

For sending large files such as an OS update the overall speed is most important. For VoIP latency is more important.

Latency should be measured for all hops in the communication.

DNS

Domain Name Service resolves domain names and records to IP Addresses. cnn.com -> 151.101.3.5

A records are for the full domain name.

CNAME Records are for www, mail, or such prefixes and point to an A record

MX Records are for email servers

Geo DNS/ Anycast DNS

Allows DNS to resolve based on the IP Address of the user.

https://www.cloudflare.com/learning/dns/what-is-anycast-dns/

Services / Protocols

Different services that your web app provide such as Real Time Communication may require specific TCP Ports to be open and available.

SSL and HTTPS

SSL Certificates allow for encryption in transit of data between a website and the web browser.

CloudFlare has SSL built into DNS

Lets encrypt is an open source SSL Certificate system with certificate rotation.

- https://letsencrypt.org/

Browsers

Different browsers offer different levels of compatibility for web applications. Everything from basic security features, to whether they support specific HTML 5 API's.

Also how apps are presented in different browsers may be different. Default font styles, and formatting can look markedly different between browsers.

CSS

CSS can do many of the things Javascript used to do and won't be blocked by Javascript blockers.

Javascript

JS is a client side language which means the server sends the code to the browser and then the browser turns the code into functionality.

Using JS reduces resource load on your servers.

JS is insecure for data because everything is sent as a text file

JS may not work on browsers due to Javascript blocking

Back End Languages

Backend languages such as Python or PHP run on the server itself. They only send data to the client that they are specifically coded to send and so are more secure for data.

Languages resource consumption can be dramatically different. Example: Ruby on Rails used more resources but was easier to use, while PHP was harder by ran leaner.

Frameworks

Frameworks allow you to extend the functionality of your programming language. They are generally designed for specific niches and you should choose the one that suits your needs.

Make sure you are able to install the Framework on the production server you will be using.

Servers

Servers are the "boxes" that will be delivering resources to your clients. Whether you run physical or virtual servers will be up to your requirements, and which OS you choose will depend on the stack your institution has the most knowledge of.

Hosting Types

Bare Metal

Bare Metal servers mean you are running an OS on a physical server. This may be a server in your house, your server room, a colocation facility, or MaaS (Metal as a Service). You get the best performance, but are responsible if the CPU fan fails

Shared Hosting/ PaaS

Shared Hosting is a service provided by companies like GoDaddy where you pay for a full platform and then just upload your code. This provides for lower labor resources and you do not have to buy hardware. The issue is that it's inflexible, and you do not control the hardware environment.

VPS - Virtual Private Servers

VPS offerings from Digital Ocean or Linode are much better than they used to be. You get full control of your OS environment, and performance is generally good.

You are still responsible for all aspects of maintenance of your VPS, and the actual abilities of the resources you have paid for may be hard to understand. (Are two "CPU Cores" good or bad?)

IaaS - Infrastructure as a Service

AWS or Azure are IaaS. You build your architecture based off numerous different components that you can connect together. This will give you the most performance and scalability, but can be very expensive and hard to understand.

Web Servers

Web Servers allow clients to access web sites. Apache, Nginx and IIS are standard web servers used for websites built with HTML, PHP and RUBY.

For Python there are web servers that include Gunicorn. When you deploy your Bottle or Django web application Gunicorn will be used to deliver your code as a web application. Gunicorn is multi threaded which means that it can handle multiple requests at the same time. The built in Python Web Server is single threaded which means it can only handle one request at a time which will cause a bottle neck on any site that has a number of people visiting it.

Reverse Proxy Servers

Reverse Proxy Servers are used for load balancing between inbound traffic and a number of web servers. This allows for fault tolerance and not over loading a single server. NGINX is used in our projects for this purpose.

Databases

Databases store data in a way that makes it easy to retrieve and interact with.

Relational Databases use a "schema" to determine the type and structure of data that will be used. These are generally considered reliable but can be hard to modify and scale.

NoSQL Databases do not have a schema and so that data that is input can change on the fly. These are generally considered easier to scale, but can lose data.

Different database servers offer different performance metrics. MySQL is faster at INSERTS, but Postgress is faster at SELECTS.

SQLite is easy, but has no security...

With new functions such as NoSQL and Vectors different Database Server software will offer different levels of compatibility.

Hosted Databases

Vendors such as Digital Ocean offer hosted databases. You pay a fee and then simply have access to database services. This means you don't have to worry about replication and maintenance of the servers and reduce security concerns.

Storage

Storage is used for the files that are required for your application. If you are using Apache this may simply be a folder on the server. Of if you use Django you may connect to a service that provides storage.

SSD vs Plater

Hard Drives use different media to store data. Platter drives have spinning platters like records. Cost per TB is lower, they are not consumable, but are slower.

Solid State Drives are faster, but are consumable. They have a usage life expectancy.

RAID

No...

Storage Arrays

Storage Arrays are generally servers designed specifically for data storage. A NAS is a single server, and a SAN is a cluster of Storage Servers.

Block Storage

Block Storage is like a hard drive volume that would be on a normal server. You can install software on the volume.

Object Storage

Object Storage is a more scalable from of storage that i good for web apps. You can store data and retrieve it, but does not have the performance of block storage.

Caching

Caching is when your system stores requested from data in a slow storage data to a high performance system.

This may be an SSD Accelerator, or even by sending data to clients to be used later.

Cache time outs are important so that old data is not presented after it has been updated.

Static Site Generator

SSG's write static pages from a Content Management System. This allows for dynamic page creation with the speed of static pages.

CDN's - Content Delivery Networks

CDN's are cache systems that move files physically closer to the end user to reduce latency.

Can be good, can also be expensive and buggy.

CloudFlare - DDoS Protection

DDoS protection protects your app from a DDoS attack. Services like CloudFlare act as a shield to your server. Clients are pointed to CloudFlare and the CloudFlare routes good traffic to your server.

Very useful if your app might be attacked, but can be buggy...