

SAE Institute London

Student Full Name: **Marco Di Summa**

Student Number: **98272**

Course Code: **CMN4200**

Assessment Code: **CMN4200.1**

Date of Submission: **4 December 2020**

Word Count: **2311**

Declaration:

I hereby declare that this is my own work, and does not use any materials other than the cited sources and tools. All explanations that I copied directly or in essence are marked as such. This work has not been previously submitted.

3 Dec 2020, London, Marco Di Summa

A handwritten signature in black ink, appearing to read 'Marco Di Summa', written in a cursive style.

Comparing WordPress and Jamstack

<i>Introduction</i>	2
<i>System</i>	3
<i>Scale</i>	4
<i>Cost</i>	5
<i>Security</i>	6
<i>Development</i>	7
<i>Team</i>	8
<i>Deployment</i>	9
<i>Future scenario</i>	10
<i>Conclusion</i>	12
<i>Reference</i>	13

Comparing WordPress and Jamstack

In the early days, the web ran from static servers dealing with clients requests using HTTP protocol and HTML. As the web technology matured, the server-side implemented technologies aimed to support virtual machines with a stack of specialised software and components running a monolithic server structure, connected to a DNS (domain name server) by an IP address. Thanks to the use of cloud services and third-party vendors, today, it is possible to deploy applications without the needs of server infrastructures.

Introduction

I recently attended a debate at Jamstack Conf Virtual 2020 that included Netlify CEO Matt Biilman and WordPress founder Matt Mullenweg. From an online article written on Aug 31, 2020, by Richard MacManus (MacManus, 2020) Matt Mullenweg, founding developer of WordPress says “Jamstack is a regression for the vast majority of the people adopting it”. The replay arrived on Sep 15, 2020, by Matt Biilmann (Biilmann, 2019) On Mullenweg and the Jamstack - Regression or Future?, opened the debate in tech communities between Jamstack and WordPress.

WordPress is a dynamic CMS server based on the LAMP stack. LAMP stays for Linux, Apache, MySQL and PHP. This CMS is extensively used in web development and offers sets of application and plugin to control the creation, management, publishing, and archiving of a site’s content. WordPress is opensource and have broad community support.

Jamstack is a serverless philosophy that serves pre-rendered static files from a CDN. It takes advantage of the implementation of JavaScript and third-part API by processing dynamic applications directly from the client-side. JAM stays for JavaScript, API and Markup, and it is relatively young compare the server-side architecture. Still, it is very active on the tech panorama. This comparative discussion paper will first look at the differences between these two approaches when it comes to building and deploying a website. Their pros and cons, and finally conclude with what the possible future for the web will be.

System

We started to talk about the web running HTTP request from a server to a client, and this is still true. However, the server-side improve drastically the ability to process information and store data. We mentioned LAMP, but the stack can vary depending on the technology implemented. A dynamic server generally has an operating system, a web server and a programming language, meanwhile, a database store the file in folders waiting for the HTTP request. Monolithic based server architecture processing client request dynamically by following some logic step like giving priority to the script process and retrieving files from the database. After the end of the rendering stage, the server is ready to deliver the contents to the browser; however, update the website will require a new request to the server. CMS improved latency issue by the implementation of caching application. Jamstack architecture is a static based technology, serving pre-rendered file from a CDN to the browser, and using JavaScript to modifying the DOM. Jamstack is part of the serverless categories which use cloud services and third-party API to implement the functionality required to the application which can operate on request. This approach takes advantage of the modern browser by shifting the scripting process to the client-side, cutting down the loading time required from a server. Storing files on a CDN eliminate latency issues and also the needs of caching application. The trend of delegate specific task to third-party vendors and cloud-based solutions is broadly adopted, thanks to the support from a flourishing market.

Scale

The first consideration in setting up a project application needs to account provisions related to traffic and performance. Monolithic architecture requires to generate a web page for every visitor request, so the infrastructure tends to over-provisioning the amount of traffic generated and scale it to avoid the risk of downtime or crashing down the system. Between component used to create and manage the application, there is no logical separation and is not possible to scale pieces of the infrastructure without influence the entire architecture of the file system. To manage spike of traffic, servers rely on caching and CDN, but this is where Jamstack lay its fundamentals. In 2002, Aaron Swartz(Swartz, 2002) posted an article titled Bake, don't fry, which is making a useful analogy between baking the contents once and serving up again and again, rather than frying them up for the user every time they are requested. Baking is what Jamstack does by building pre-rendered Html page hosted on a CDN. One concern of the "baking" philosophy is related to the use of dependency, in keeping track of which page depends on which other when the project reaches a fair amount of complexity.

Cost

There are different associated costs with the designing applications stack we choose. Also, we mentioned how to estimate the traffic level is fundamental when it's come to plan for the hosting infrastructure required, and often the costs are equally proportionated to the maximum capacity estimated. The final architecture usually results in multiple database servers, application servers, caching servers, load balancers, which have related cost for every piece of the infrastructure, eventually duplicated to provide staging, testing and development environments. Jamstack sites based are far simpler architecture which relies on the CDN able to satisfy large peaks of traffic, without the needs of the traditional server infrastructure. In term of hosting costs, Jamstack and serverless architecture in general, are cheaper, but those applications generally require additional service. However, according to an article by Chris Coyier (Coyier, 2020) compare the two approaches underlining how both are free at the very starting point, and getting more pricy at an enterprise-level. Jamstack evaluates their enterprise pricing starts at \$3,000/month, while WordPress is around \$2.000/month. Still, the modularity of Jamstack aloud solution to best suit an eventually start-up. Another alternative from the serverless market is to pay as much service has been consumed.

Security

Serverless architecture opens new scenarios in term of security. Because of the multitude of theme and plugin combinations available, servers environment are subject to attack attempting to penetrating the infrastructure by forcing some logic. According to The Hacked Website report 2017 (sucuri.net, 2017), WordPress infections rose from 74% in 2016 to 83% in 2017. To mitigate poor system administration, credentials management, and lack of security knowledge, the popular CMS published an article to improve best practice, Hardening WordPress (wordpress.org, 2020). The reality is that WordPress is a secure server; whitehouse.gov is on WordPress. Still, it needs to deal with a broader kind of opportunity to hack the system, so it needs more maintenance in term of security than a serverless infrastructure. The static hosting site reduces consistently the area of the application exposed to attack by using files which are read-only based, shifting part of the security management to vendors which are strongly motivated in be performing in security implementation. Eventually, attacks unrelated to the hosting account will influence just the service involved, reducing the risk of contamination for the hosting environment. By reducing the area of logical computations in the building environment, Jamstack produces assets which are static document dislocated strategically by the CDN for easy client access, enhancing dynamic experience directly from the browser.

Development

Development experience is essential when it comes to design an application; poor development experience penalised the ability for developers to finalise the designing strategy. CMS are more likely to support an unskilled user, by sets of frameworks, configurations and restricted design styles, ensuring a smooth path for first experience designer. It is possible to configure well-organised permission lists, to support the web application when it grows in complexity. However, this virtual machine employs a series of tasks necessary for mitigating disruption in the workflow. We mentioned already how maintenance related to security are fundamental to the health of the environment, as any piece of the server is related to each other, a project based on CMS needs to consider the amount of work required for the development and time invested for maintenance and implementation. WordPress for almost two decades support and implement a monolithic approach to web development. Serverless applications take advantage of the most recent technology developed on the client-side. Static-site generator provides faster building cycle, and produce an impressive amount of pages in a few seconds, the tools available to set a performing pre-builds environment are growing exponentially, along with emerging techniques to splitting task in well-defined purpose. Even though serverless application offer grate modularity to minimise unnecessary complexity. It would be not easy to manage dependency and API without a clear architecture strategy, but it will always offer a content-first policy thanks to the pre-rendering process.

Team

Developing and maintaining an application require different working profiles with determinate specialisation related to the components of the server. Because of their complexity, traditional architectures tend to relay on bigger teams. Complex DevOps are leaving the space for simpler environments to favourite development workloads, delegating less task in term of maintenance. The reduction of specialistic skills, the smaller maintenance surface, shifting monolithic ecosystems to modular architecture implemented by third-party vendors, contributing to lower the boundaries for innovation and iteration. Immediate improvement relay on the implementation of VCN that allowed to work on branches, step back to previous versions and merge code sourced from others developed version.

Deployment

Deploying websites commonly rely on FTP, moving files and folder from a server to another, without the chance to revert the process once it's started. WordPress, along with technology implementations, developed a series of plugin and software that make the migration from the local environment to the live server easier for beginners. However, a first set of the CMS it is always necessary. Skilled programmers are more likely to set procedures by defining custom configuration strategies to ensure some deployment automation. Even though this kind of processes are PHP based and needs to be computed by the server, still they had essential improvement in reducing latencies. Gilbert Pellegroni (Pellegroni, 2018) published an article on May 08, 2018, listing four different approaches to design a deployment strategy supported by automation aims to speed up the deployment process. The list includes VCS based support and atomic deployment as alternatives to improve WordPress workflow. VCS are a fundamental asset in the Jamstack philosophy. Because of the complexity of new applications, the ability to revert the building stage to previous versions is an essential aspect to consider to mitigate mutable deployments, easily subjected to human errors and faulty deployment propagation. Immutable deployments are a self-contained set of assets which contain in a generated repository sitting inside a VCN generally connected to the CDN. By default, the modified

code committed to VCN will keep track of changes and merge them instead of replacing the entire document. Commitments will be listed and organised, allowing accessing older versions of the application and eventually stage it back if necessary. The VCN tracking system creates a better environment for testing and experimenting with new functionality.

Future scenario

Jamstack is become a popular name in the business, offering a general design process with the implementation of modern service aims to shift the work to focus on applications development. It is performing when it comes to working on continuous delivery services. In conjunction with CDN and VCS, it is handy when a fast and easy deployment is a fundamental requirement of the application. However, to serve heavy and dynamic applications, Jamstack architecture may require a significant amount of API and dependencies, which could be challenging to manage. WordPress, along with the LAMP stack, has been around for years, and the technology adopted is considered tested and secure. It comes with a Linux based service, a wide range of libraries, frameworks, and a local database to enable dynamic computing.

However, the monolithic structures of WordPress require extra workloads in term of maintenance and scalability, and security matters accounted to the server environment. A growing number of applications shifted to Jamstack, one million on Netlify (Biilmann, M. & Bach, C. (2020), and a large community of developers believe that static server improves security performance, scaling costs, and developer experience. The backbone of Jamstack is adopting git-centric workflow, serving application by the CDN. Also, the Jamstack mass-adoption supported by active communities working on deployment optimisation and its building tools. WordPress widely adopted to created any sort of website, it is a product optimised mainly for blogging. The popular CMS got a bad reputation on relying on dozens of plugins, but because is well established in web applications development with a strong community of contributors it is likely to be around still for a while. Ohad Eder-Pressman(Pressman, 2020), Stackbit CEO, on Sep 04, 2020, describe how the abstraction of computing introduced by AWS in 2006 creates the base for disruptive innovation as cloud servers, which are today adopted from applications with specific goals. The achievements of WordPress in term of market share are remarkable. Still, the new technology trends marking a diversification of service based on custom solutions, web developer abilities require new skills to comply with the variety of approaches. Decoupling the component of a web application would be an opportunity to rethink web application whit a modular system adaptable to specific business models.

Conclusion

The web along the years adapted to new technology heading for new solutions to enhance web capabilities. Many failed to run long, and others contribute to support new implementation. The debate between web communities should be always always encouraged, along with networking and collaborations. The web environment is in constant evolution, but the final user should always come first. As WordPress has been around for almost two decades, Jamstak has been absorbed from the environment and growing in popularity. The implementation of HTML5 left behind plugin like Flash, CSS3 and javascript empowered graphic and interactivity. While older technology adapting to the last implementation, new approach shaping application design around these new capabilities. Still, the best choice is the one that better responds to the needs of the final user, so I would like to consider diversification as an extra tool to achieve the final goal.

Reference

Biilmann, M. (2019), *Modern Web Development on JAMstack*. O'Reilly Media, Inc [Online].

Available at: <https://www.netlify.com/oreilly-jamstack/> (Accessed: 28 Nov ,2020).

Biilmann, M. (2019), *On Mullenweg and the Jamstack - Regression or Future?*. Opinon Inside.[Online].

Available at: <https://www.netlify.com/blog/2020/09/15/on-mullenweg-and-the-jamstackregression-or-future/> (Accessed: 28 Nov ,2020).

Biilmann, M, & Bach, C. (2020) *One million on Netlify*, News & Announcements. [Online]

Available at: <https://www.netlify.com/blog/2020/08/03/celebrating-1-million-developers-whatsnext-for-netlify-and-the-jamstack/> (Accessed: 28 Nov ,2020).

Coyer, C. (2020). *WordPress and Jamstack*. [Online].

Available at:<https://css-tricks.com/wordpress-and-jamstack/> (Accessed: 28 Nov ,2020).

Eder-Pressman, O. (2020) *An Open Letter to Matt Mullenweg: What folks often get wrong about Jamstack*. [Online]

Available at: <https://www.stackbit.com/blog/open-letter-to-matt-mullenweg-what-folks-oftenget-wrong-about-jamstack/> (Accessed: 28 Nov ,2020).

Hacked Website Report, (2017)[Online]

Available at: <https://sucuri.net/reports/2017-hacked-website-report/> (Accessed: 28 Nov ,2020).

MacManus, R. (2020). *WordPress Co-Founder Matt Mullenweg Is Not a Fan of JAMstack*. [Online]

Available at: <https://thenewstack.io/wordpress-co-founder-matt-mullenweg-is-not-a-fan-ofjamstack/> (Accessed: 28 Nov ,2020).

Pellegrom, G. (2018) *WordPress Deployment Part 1: Preparing WordPress*[Online]

Available at: <https://spinupwp.com/wordpress-deployment-workflow-preparing/> (Accessed: 28 Nov ,2020).

Swartz, A. (2002). *Bake, Don't Fry*. [Online].

Available at:<http://www.aaronsw.com/weblog/000404> (Accessed: 28 Nov ,2020).

Wordpress.org, (2020) *Hardening WordPress/support/article* [Online]

Available at: <https://wordpress.org/support/article/hardening-wordpress/> (Accessed: 28 Nov ,2020).