Serverless Computing Cost Efficiency

Nelson R. Medero Núñez
Master in Engineering Management
Dr. Héctor J. Cruzado
Department of Civil & Environmental Engineering and Land Surveying
Polytechnic University of Puerto Rico

Abstract *— A software company in San Juan is currently modernizing the software product offering technologies for web apps. They wanted to understand how Serverless Computing can reduce costs. A proof of concept was done to refactor the web app and execute it into a Serverless architecture from an Infrastructure as a Service architecture. The Infrastructure as a Service architecture includes a Virtual Machine Web Server, while the Serverless architecture was done with the use of a Function App back end providing the web app front end from a storage static website. All these components were compared and appraised against the Virtual Machine Web Server costs. The Serverless Model is cost-effective and more convenient than the Infrastructure as a Service model. It is recommended that the software company in San Juan continue to pursue this option.*

Key Terms *— azure functions, cloud, IaaS, SaaS*

INTRODUCTION

On today's cloud computing there is a modernization of Software as a Service (SaaS) to provide solutions in the form of serverless computing such as Web App Services or Function as a Service (FaaS). In manufacturing, when producing any goods, the Cost of Goods Sold (COGS) is the term used for the cost of the different aspects of how much it costs producing that good. In software, part of the materials of provisioning this good is the hosting environment where the software resides. The paradigm of this provisioning has evolved from local servers in land and virtualization to cloud computing providing Infrastructure as a Service (IaaS), also known as Virtual Machines (VM) in the cloud. More recently, Containers came out as a new alternative, especially for web applications. About a year later Serverless Computing was available as an alternative with a new type of offering for web apps [1].

A software company in San Juan, Puerto Rico is working with the modernization of their SaaS offering. One of the options taken into consideration is to use the best fit technology to be able to provision the SaaS products as Serverless Computing. A decision needs to be made for whether the current infrastructure should be moved to Serverless Computing and how this affects the COGS in the server provision to host the software. Around this decision, other processes should be modified like, for example, the Software Development process and life cycle.

The project's objective is to reduce cost regarding the provisioning of resources for a web application. The idea is to be able to provide the same service at a lower cost by using a modern approach. Using the same cloud provider with Microsoft Azure the project will compare the cost between a Web Server, VM and IaaS approach versus a Serverless approach. The cost of converting or reengineering the web application will not be considered by assuming that the current application can be deployed in both environments.

Cloud Providers offer pricing with different combinations of servers, networking, and publishing. A similar IaaS configuration of a single web application of the software company in San Juan using IaaS was done. The new configuration was designed into the Serverless provisioning and compared to the existing IaaS configuration. A cost analysis was made to understand monthly costs against the new cost when providing the web app in a Serverless offering.

LITERATURE REVIEW

Before cloud computing, the term serverless referred to Peer-to-Peer (P2P) collaboration, which

means that a server was not involved. In cloud computing, there is server involved, but in a different way and was brought by AWS in 2014 [2]. Cloud Computing is "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models" [3]. Using this definition as a basis, Serverless Computers stress the concept of minimal management. In the cloud, IT and developers have been paying a lot attention to the benefits of a "NoOps" environment [4].

Serverless Computing can be described as an evolution in the last decade of managing different resources and differing to the cloud provider the server management. The industry went from land servers, to virtualization, then virtualization in the cloud and containers, to different kind of services like IaaS, PaaS, SaaS and FaaS/BaaS. These SaaS, FaaS and BaaS associated to Serverless Computing [1]. In Figure 1, the evolution and progress into serverless computing is shown.

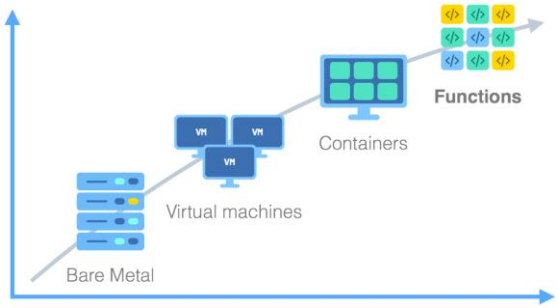

Figure 1
Serverless Computing Evolution [5]

One of the major impacts has been the economical part of this new cloud technology. To take the most advantage, there is going to be a technological transformation. This technological transformation is mostly overlooked given that if a company is looking to move everything to serverless an assessment needs to be done to measure and plan the effort of doing so. Cloud computing is price effective and can control more the overall resources [6].

There are different costs when discussing IaaS and Serverless. In the IaaS model, the user pays for the creation, storing and usage of the infrastructure. Also, the user has to take care of any maintenance. In Serverless, it should be down to the transactional usage. In other words, all the request calls of the resources provided for the web app [2], [7], [8].

Price models will depend between cloud providers. Using Microsoft Azure models, there are three main models [9]:

- Pay as You Go: in general terms means that is a per second base, based on usage and there is control for when to start and stop.
- Reserved Instance: Fixed price for one or two years, unlimited usage and can be always-on when needed.
- Executions: Transaction based and with executions counts in a period of time.

DISCUSSION OF THE PROJECT

The different architectures provided that delivers the web app show the conditions and give more insight for which scenario can be more cost effective. A database SQL VM is shown in the diagrams. This database VM will be constant for the project because is a shared database between other applications. These other dependent applications can be moved to other technologies or even Serverless. If this happens, the storage approach can be redesigned. For this project, this does not take any effect in cost or any other analysis.

System architecture

The software company in San Juan is currently using an IaaS approach to provide the web app. Figure 2 shows the architecture is composed of a web server VM which contains all resources in a single box machine.

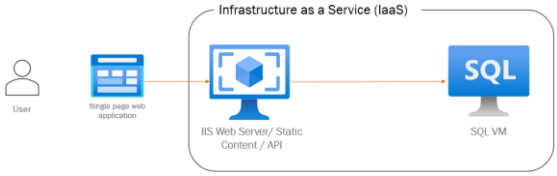

Figure 2
Current IaaS Configuration

The VM uses Windows Server and publishes the static content web app using Internet Information Services (IIS). The same IIS is used for the back end to execute Node.JS, which is a platform to develop web apps and runs JavaScript outside a web browser [10]. Node.JS provides the backend management to execute business logic and access storage, which in this case is the database in the SQL Server.

In Figure 3, the new serverless design is shown. It's composed of a managed storage account with a content delivery network (CDN) to provide faster delivery. This two first pieces contain the static content web app. The backend, also executing Node.JS, is managed by the Azure Functions App. As explained before, the same SQL Server database is used.

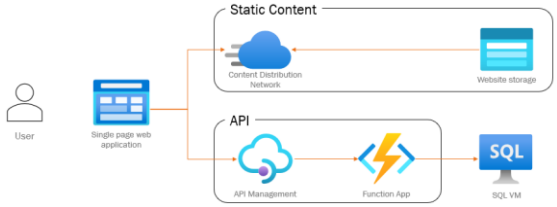

Figure 3
Serverless Design

Proof of concept process

A simple proof of concept (PoC) was done to validate that the web app was able to be used in this new architecture. To narrow the scope, the PoC focused on loading the login page. The login page uses all layers of the application to load and was able to communicate all the way to the database. The steps done to shift to a Serverless provisioning were:

- Create the SQL VM in a test environment.
- Prepare and obfuscate the database.
- Create an Azure Functions application.
- Understand architectural differences between platforms.
- Refactor the backend code running on IIS to be able to be compatible with Azure Functions.
- Communicate the new Azure Function with the database.
- Create a storage account and configure it for a static website container.
- Compile the web app static content to connect to the new back end.

Analysis and Results

The software company in San Juan provided the current cost of the IaaS configuration. This will be the baseline for comparison of the cost analysis, not taking into consideration the database because the cost will be neglected. Table 1 shows the VM server costs and the price model in the Microsoft Azure cloud. The project focuses in the web application given that the SQL Server VM will still be needed since is shared between other applications in used.

Table 1
IaaS Server Cost

Server (VM)	Cost (Montlhy)	Price Model
Web Server	$210	1-Year Reservation
SQL Server	$3000	1-Year Reservation

Comparing cost for both designs (IaaS and Serverless)

The web server cost comes from a combination of storage devices and operating system (OS). The software company in San Juan moved to a 1-year Reserved Model because this server will be always On, making this a more attractive deal.

The Serverless model is more similar to a Pay-As-You-Go model where you pay for what you use. The difference between this in a VM is that if you are in a Pay-As-You-Go, while the VM is On, even if it is not in use, the user will be charged. In Serverless there are various plans. The one use for this project is the Consumption model. This model uses two-man factor for billing. First the resources used to execute your Function App and second the number of executions done against any end point of the function App [9].

Using the price of the resources created for the PoC in the Microsoft Subscription and the Microsoft Azure pricing tool, in Table 2 the monthly price to provision the Serverless web app is shown.

Both the static website content and the Function App source code needs to reside somewhere.

Table 2
Serverless Cost

Web App Resources	Monthly
Static Website Storage Account (Front End)	$1.04
CDN Static Zone 1: North America, Europe	$0.08
Functions App (Back End) (54M Executions)	$58.20
Functions App Storage Account	$1.02
Total	$60.34

Both storage accounts are in place and the cost include read traffic and storage of the files. The CDN was configured to use the static website storage to be able to deliver globally and based on the static web. The site is 10MB in size and can use about 14MB in resources. A 1GB minimum was used for the site pricing calculation. Since the Functions App is a consumption model a scenario was done to provide a baseline for that cost. The scenario taken to provide an estimate in cost was using the Chrome debugger, a typical navigation was done with an average of 120 requests. With a potential of about 15k users and a potential use of 30 days in a month:

$$15k * 120 * 30 = 54{,}000{,}000\ executions$$

This number represent a worst-case scenario where every potential user uses the system all the days of the month. As mentioned before, Microsoft Azure provides a toll where you can plug-in the number of executions, the execution time for each request and the resources used. For this scenario, using the Chrome debugger, a 128MB resources were used and an average execution time of 500ms. This results in the cost of $58.20. The equation provided by Microsoft Azure is the following [9]:

$$Price = (NE * ED * (RC - MFRC) * \$0.000016/\text{GBs}) + ((NE - MFNE) * \$0.20)$$

where:

NE = Number of Executions

ED = Execution Duration

RC = Resource consumption (GBs)

MFRC = Monthly Free Resource Consumption

MFNE = Monthly Free Number of Executions

The result is a saving of $149.66 monthly. That is 72% saving of producing the same web app in this model.

To understand the limit until the model stops being cost effective is obtaining the Max Number of Executions. Using the same formula and setting the price to $210.00, the NEmax = 216,000,000 executions in a single month. At this point, the model should be reevaluated.

As mentioned, the focus of this project is to evaluate cost of provisioning the web app. Something to consider is the initial investment in the conversion of the web app from one technology to another. That can be done as a future work and be able to estimate that investment. A valuation exercise can be done, like for example a Net Present Value analysis to understand the entire picture from changing the app to release.

CONCLUSION

Throughout the project, it was illustrated what is serverless computing and its benefits to the software lifecycle. In addition, how modernization has taken place in the last decade to be able to provide this new way of delivering software. The software company in San Juan can continue with its effort of converting and go forward with the steps used to get the conversion of the web app done. Moving to Serverless is the best option given the cost reduction the Software in San Juan will gain. In other words, it will produce the same software at a lower cost. A good recommendation for the web app conversion is to estimate the effort to be done and be able to do an analysis and valuation. Also, if the user base is going to grow, understand how this will affect the Number of Executions. Finally, plan to migrate the storage from a SQL VM to another model, so that

Software Company in San Juan can also reduce cost in that area too.

REFERENCES

[1] J. Ben-David, "Turbonomic Blog," 22 January 2020. [Online]. Available: https://blog.turbonomic.com/cloud-computing-two-decades-in-review.

[2] P. Castro, V. Ishakian, V. Muthusamy and A. Slominski, "The Rise of Serverless Computing," *Communications of the ACM*, vol. 62, pp. 44-54, December 2019.

[3] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," *National Institute of Standards and Technology*, p. 2, 2011.

[4] C. Boulton, "CIO," 13 February 2019. [Online]. Available: https://www.cio.com/article/3244644/serverless-the-future-of-cloud-computing.html.

[5] G. C. Fox, V. Ishakian, V. Muthusamy and A. Slominski, "Report from workshop and panel on the Status of Serverless Computing and Function-as-a-Service (FaaS) in Industry and Research," in *First International Workshop on Serverless Computing (WoSC)*, 2017.

[6] R. Makhlouf, "Cloudy transaction costs: a dive into cloud computing economics," *Journal of Cloud Computing: Advances, Systems and Applications*, pp. 1-11, 2020.

[7] A. Alda-Rodríguez, G. Diaz-Lopez, M. Evgeniev, F. Alvarez and P. Horrillo, "BBVA Innovation," 09 December 2020. [Online]. Available: https://www.bbva.com/en/economics-of-serverless/.

[8] M. Evgeniev, C. Gallego-Rodriguez and F. Alvarez, "BBVA Innovation," 11 April 2019. [Online]. Available: https://www.bbva.com/en/serverless/.

[9] Microsoft, "Azure pricing," Microsoft Azure, 2021. [Online]. Available: https://azure.microsoft.com/en-us/pricing/. [Accessed May 2021].

[10] Joyent, "Node JS," Joyent, 2021. [Online]. Available: https://nodejs.org/en/about/. [Accessed May 2021].