Serverless Computing : Economic impact on Digital Enterprise operations & Implementation of Azure function

Neeraj Dhiman

Student, Dept. of CSE , Baba Banda Singh Bahadur Engineering College, Fatehgarh Sahib, Punjab, India

---

Abstract - *Cloud computing can be think of as the evolution of on-premise data centers.Cloud computing comes with an idea of abstracting the on-premise servers into the cloud.Over the time new technology and enhancements were made into cloud computing techniques. In early 2014's the idea of Serverless Architecture came which means you write highly compact individual functions that do one thing - and run in the cloud.The name serverless may be confusing to someone but it doesn't mean you don't need servers, you need servers to run your code, So term serverless is just a misnomer.*

This paper presents different techniques which can be used to deploy an application , economical difference between them, how an appropriate deployment technique can affect organizations operations.

Key Words: Cloud Computing, Serverless Computing, Serverless Architecture, Deployment Cost Analysis, Deployment Time Analysis, Azure function.

1. INTRODUCTION

1.1 Serverless Computing

It is a cloud computing execution model where cloud provider dynamically manages the allocation of resources. The pricing is based on the amount of resources used or consumed by an application, rather than on pre-purchased units of capacity [1]. It is a form of utility computing.

When someone say Serverless computing it doesn't means this technology will not use servers, it still requires servers, hence it's a misnomer. The name "serverless computing" is used because the infrastructure management and capacity planning decisions are not known to the developers they are completely hidden from them. The code used in Serverless can be used in partnership with the code deployed in traditional styles, such as microservices [2].

2. Literature review

The first official Serverless platform named 'Lambda' are disclosed in late 2014 by Amazon Web Services [3].

Since then, every the major cloud service provider is working on some kind of Serverless platform some them have also released initial version of their platforms where rather than deploying and running monolithic services, or dedicated virtual machines, users are able to deploy individual functions, and pay only for the time that their code is actually executing.

These technologies are gathered together under the marketing term 'serverless' and the providers suggest that they have the potential to significantly change how client/server applications are designed, developed and operated.

For service or platform to be considered serverless, it should provide the following capabilities:

- No server management – You don't have to plan or maintain any servers.
- Flexible scaling – Application scalability can be managed automatically by you or by adjusting its capacity through toggling the units of consumption rather than units of individual servers.
- High availability – Serverless applications have built-in availability and fault tolerance.
- No idle capacity – The most important service they provide is you pay for how much you have used.You will never for the time for which your application is idle in other words you don't pay for the idle capacity. There is no need to pre-plan or over-plan capacity for things like compute and storage.

Various Deployment Methods and Techniques

- On premise deployment
- Cloud Deployment

Cloud Deployment Techniques

- Dedicated virtual servers
- Containers with Orchestrator
- Serverless Computing

2.1 PROBLEM FORMULATION

The main problem when comes to deployment of any software you can either choose on-premise, private cloud solutions or public cloud solutions.

But initially its very difficult to identify what will be the best solution for you in terms cost and efforts. So it's important to compare all these approaches and explain the clear difference of cost and efforts required.

Microsoft Azure function is another serverless approach similar to AWS. Azure function is in its early stage. Its version v1 is released and work on v2 is in progress.

There were some issues in azure version v1 which will be fixed in v2. One of the issue in Azure version v1 is "handling of Multipart-data".

E.g., Suppose you have written a Azure function of HTTP Trigger type which accepts textual data as well as Multipart data (images, files etc.).

You can easily read textual data from incoming POST request but when you will try to read a POST request which includes Multipart data then Azure function throws 500 internal server error .

Objectives of my research are as follows :

1. To study various deployment techniques.
2. To study Economics of serverless computing (Cost Analysis)
3. To propose solution for handling Multipart-data in Azure function version 1 (v1)
4. To compare proposed deployment technique with the techniques studied in literature.

2.2 Design methodology

To implement serverless function one needs a Cloud service provider account subscription that can be of Azure OR AWS accounts subscription so that he/she can deploy his/her code there and compare different aspects of deployments.

First two objectives can be achieved by deploying applications directly on AWS or Azure.

For third objective one need to have an azure function of HTTP trigger type. Below are the steps how to create an azure function from azure portal.

- Start by opening Azure portal and go to the Azure Functions option or service and follow the instructions.
- It will ask you for scenario which you'd like to use or address.

Below shown Flowchart explains how one can create a Simple Azure Function :

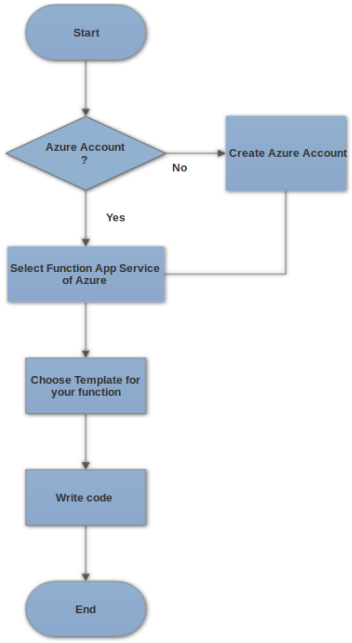

Fig -1: Flowchart can create a Simple Azure Function

3 Implementation and results

3.1 Economic Comparison of Deployment Techniques

Here economical aspects of Deployment techniques will be compared. The main factors on the basis of which this comparison is based are Cost and Time.

To setup a server one at least need

- CPU
- RAM
- Hard Disk
- Network Switches & Routers

In On Premise deployment Organisation needs all above material of their own but In cloud Deployment Developers do not have to worry about Hardware all necessary things handled via Cloud service Provider.

Example:

Let's take an example of a live Software named `Veris` which is Cloud based Veris Management System. Let's compare the Cost of Hardware if Veris is Deployed either On-Premise or On Cloud.

Parameters :

- Can Handle upto 10k users traffic per day
- High Availability

3.1.1 On Premise

- As discussed above On-premises softwares are deployed, managed and runs on computers on the premises (in the building) of the person or organization using the software, rather than at a remote facility such as a server farm or cloud.
- Efforts Required To Setup an On-Premise server is shown in figure below

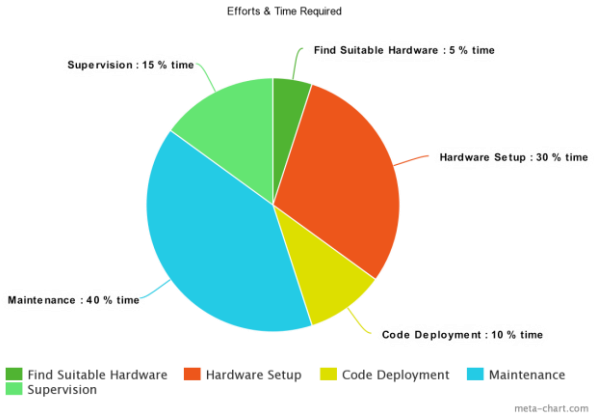

Fig -2 : On-Premise Server Setup Efforts

- To handle 10k Users Traffic per day one need at least
  - 16 GB RAM
  - Intel Xeon with 3.3 GHz speed Processor
  - Cisco SG300-28 28-Port Gigabit Managed Switch
  - 1 TB Storage
  - Dedicated internet connectivity with at least 5mbps speed (Recurring Cost)
  - Backup Hardware for High Availability

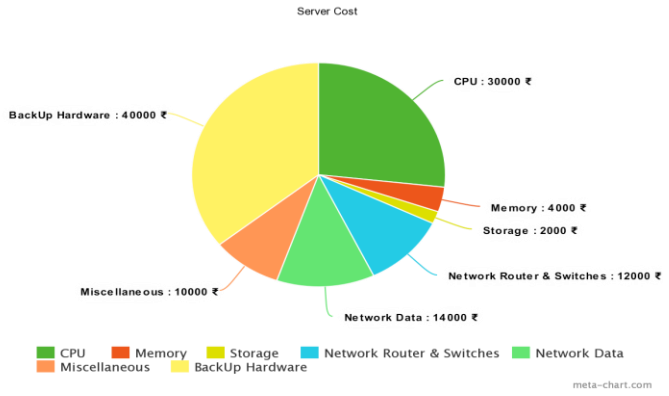

Fig -3: Cost of Server Setup

3.1.2 Cloud or Virtual Machine

1. Efforts Required To Setup an On-Premise server is shown in figure below

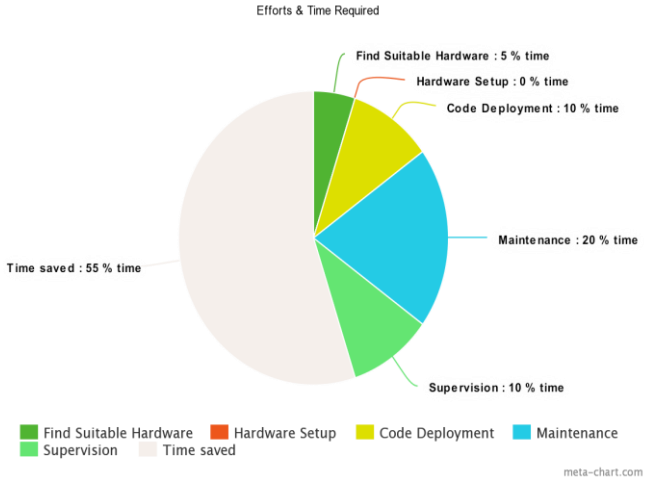

Fig -4 : Cloud Server Setup Efforts

2. Similar Configuration as mentioned in On-Premise, An Azure Standard D4s ve Machine costs around 10k per year with HA (High Availability) whose configuration is
   - 4 vcpus
   - 16 GB RAM

A General Comparison of ON-Premise and Cloud :

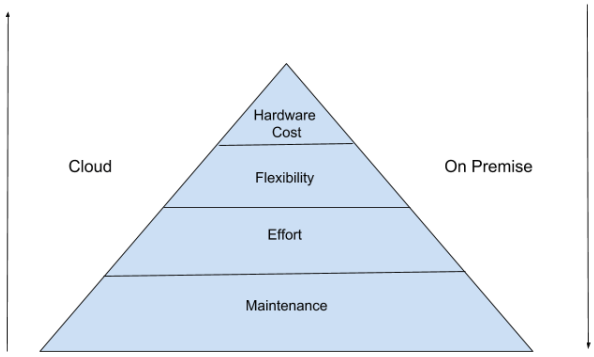

Fig - 5 : General Comparison of On-Premise and Cloud Deployment

From above figure anyone can see Both Cloud and On-Premise have their own pros and cons like :

- In Cloud Deployment developer face less Maintenance but in On Premise Maintenance Cost as well as efforts are very high

- Also the flexibility to update our server like if developers think they need to add more processing , Memory or any upgradation, then in On premise the flexibility to do the same is very less and time consuming but on the Other hand In Cloud since all above handles via Cloud service provider so developers just need to subscribe the better service and all migration is handled by Service Provider.

- However In cloud since Subscriber are paying in Recurring mode and not a one time cost so Over the time you feel the amount you are paying for Hardware is more but on the other hand in On-Premise you have to pay only once (replace any when any failure occured) you pay less.

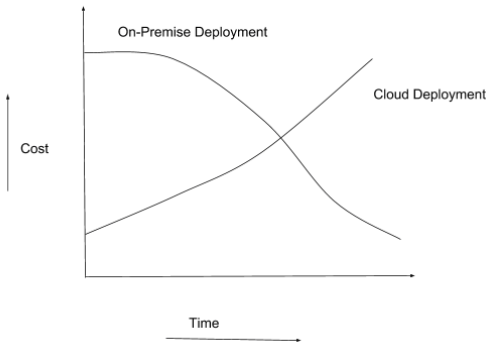

Fig -6 : Cost and time Comparison of On-Premise vs Cloud

Form above figure one can easily understand that initial cost of setting up a cloud server is very less as compared to ON-Premise Deployment but as the time progress since you don't need to buy new hardware until unless there are any hardware failure the Cost of Deployment dec as compared to Cloud Deployment.

3.1.3 Serverless Function Setup Efforts & Time

1. Below figure shows the Cost and time required to setup and Serverless function.

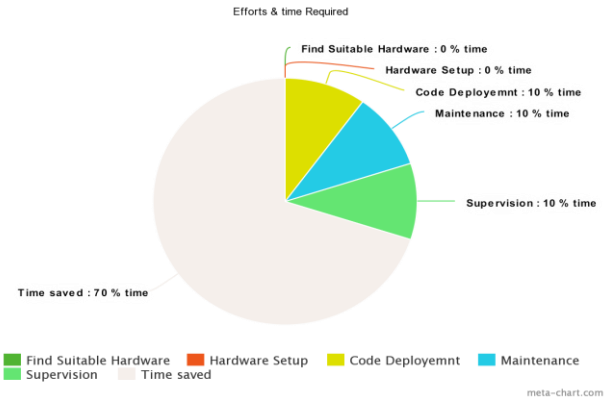

Fig -7: Serverless Function Setup Efforts & Time

2. Below figure shows the usage of an Azure function

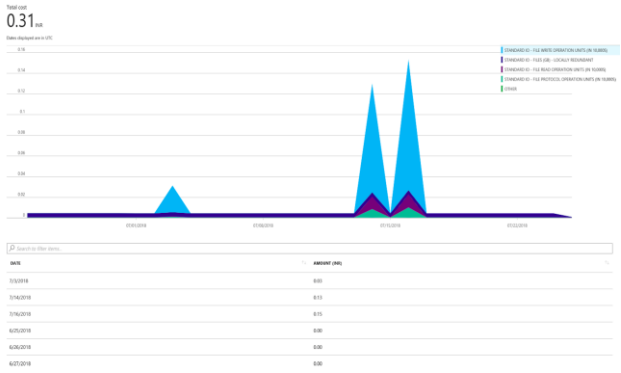

Fig -8: Azure function Usage

3.1.4 Cloud vs On-Premise Cost Comparison

Table -1: Sample Table format

Deployment Service	Memory In GB	CPU	CPU Speed	Storage	High Availability	Maintenance Effort	Cost
On-Premise	16	4	3.3 GHz	1 TB	No	YES	50k
Public Cloud	16	4	2.3 GHz	32GB	Yes	No	10k/year
Serverless	N/A	N/A	N/A	N/A	Yes	No	As per usage

3.2 Risks and Limitations

1. **Loss of control over infrastructure:** All the infrastructure controlled and managed by the cloud service provider, so one cannot customize/optimize the infrastructure according to his/her requirements or needs.
2. **Vendor Lock-In:** If you wants to switch from one vendor's serverless service to another's then it may require momentous time and efforts.
3. **Compliance concerns:** Since all the infrastructure are managed by cloud service providers then only they are responsible for doing vulnerability scanning and penetration tests on infrastructure. But as a consumer of serverless offerings, you cannot do these tests.

4 Conclusion and future scope

4.1 Origin and Expansion

In Early 2000's cloud computing vendors started providing open-source software and infrastructure setup to IT organizations, which can be assumed as the origin of Serverless.

To standardized the services provided by various cloud service provider cloud-computing models came, namely,

- Infrastructure as a Service (IaaS),
- Platform as a Service (PaaS), and
- Software as a Service (SaaS).

After some time Cloud providers also started providing hardware and software assets as per-use service-based models, resulting in significant reductions in costs when compared to existing expenditures of the IT organizations.

In 2014, Amazon Web Services released a groundbreaking service called AWS Lambda that offered a new way to deploy web, mobile or IoT application's code to the cloud. Instead of deploying the entire codebase of the application all at once, Lambda allows to deploy application's functions individually, to their own containers.

Microsoft announced its answer to AWS Lambda by launching "Azure Function" at its Build 2016 conference.

Google released "Cloud Function" as its own Serverless service in February 2016.

Serverless, originally a market limited to AWS's Lambda service, has expanded dramatically in interest because of competitive and upcoming services from Microsoft to Google to IBM [4].

4.2 Better optimization

Yes, it could be argued that the original infrastructure stack for the web application could be further optimized from the cost perspective, and opting for reserved instances could reduce the cost by about 75 percent. Or perhaps weaker compute instances could have been selected. The bottom line is this: Any such cost reduction would have resulted in either:

- Making a long-term commitment to keep the solution online, or
- Introducing potential capacity problems if the infrastructure is not fit for purpose.

In Conclusion, Cloud Computing Platforms today are very useful where efforts, initial cost and time are important especially in Small Digital Enterprises. The Economics of hosting a project in shorter span of time without worrying about Security and Maintenance cost is very affordable.

ACKNOWLEDGEMENT

1. I would like to thank Digicred Technologies (Veris) for sharing Deployment technique they used over the time and how it affected their operation and cost.

2. I would like to thank Ankit Popli for valuable assistance in the data collection and useful discussions.

REFERENCES

[1] Miller, Ron (24 Nov 2015). "AWS Lambda Makes Serverless Applications A Reality"

[2] MSV, Janakiram (16 July 2015). "PaaS Vendors, Watch Out! Amazon Is All Set To Disrupt the Market"

[3] Miller, Ron (13 Nov 2014). "Amazon Launches Lambda, An Event-Driven Compute Service"

[4] Novet, Jordan (9 February 2016). "Google has quietly launched its answer to AWS Lambda"