A Study on Current Scenario of Serverless Computing

Aparajita Gogoi[1], Mr. Vignesh S.[2]
[1]*MCA Scholar, School of CS & IT, Dept of MCA, Jain (Deemed-to-be) University, Bangalore, India*
[2]*Assistant Professor, School of CS & IT, Dept of MCA, Jain (Deemed-to-be) University, Bangalore, India*

Abstract **- Serverless computing has gained prominence in being the most compelling archetype in the deployment of various services and applications. Of late there has been a tremendous change in cloud technologies such as programming models, services and abstractions which is brought about through implementing serverless computing architectures in different products by cloud practitioners such as software application architects, or developers and end users. Since it is a relatively new technology and also due to its high architectural effect, its acquisition is lacking quite behind, and it is not being utilized to its maximum capacity as it should have been. And as such most developers or engineers might be aware that now cloud computing is not synonymous to only backend developments and devOps but due to rise in backend-as-a-service, the barrier to entry into cloud computing has been significantly lowered for frontend developers. In this project we will try bridging the gap between frontend and backend development by leveraging a new generation of services and tools provided by the AWS Amplify Framework, we will verify how such a serverless application can be adopted in running a business like an "hotel app" rather than using traditional web technologies. This serverless technology gives us a new opportunity to govern the reduction of some amount of operational costs by efficient optimization and management of cloud resources, or managing scalable cloud applications or entire development stack, one such ecosystem is that of React and Aws. We shall be using React to develop the front-end part of the website while the backend will be taken care of by using AWS Amplify services and other Aws services like DynamoDB or Lambda to maintain the database or some API's.**

I.INTRODUCTION

Serverless computing is an execution model for the cloud in which a cloud provider dynamically allocates resources wherein the end user is only billed for the resources or services used during the period of time. Serverless computing permits us to fabricate and run various types of applications and services without even having to contemplate about servers. In serverless computing, servers are still present, but the key difference is that AWS will handle all the server management.

AWS Lambda has acquired a ton of fame all throughout the most recent couple of years, with numerous architects/developers embracing it to function microservices and APIs for a negligible part of the expenses of traditionally facilitated hosted systems with predominant scaling abilities. Nowadays we can run fully functional Node.js web applications with React frontends on AWS Lambda without the requirement for any dedicated EC2 instances. With Lambda, architects are utilizing serverless infrastructure with Amazon functioning as a delegate between the client and code execution in a software containerization. environment, AWS Lambda does the following: Gives rules architects to be followed when submitting the code and how to enter the code into containers by means of automated measures, thus the whole backend development process becomes a well-managed service for the end users.

One of the popular front-end frameworks is React which can be implemented to create single-page applications (SPAs). React has recently emerged as one of the indispensable technologies upon which the building of the new and modern web development is being done on. In this project, we have tried to configure how a complete a web development environment in association with React is utilized to create an ecosystem which enables us in building applications. We went through all the vital and essential steps in getting our React app already which included deploying, connecting the frontend and the backend, and finally the react app being supported. Next, we went through as to how we can create certain React components to categorize and show our application. After that we configured how to add cloud services, for example like a database using JSON data and performed several checks in order to that ensure

our code works on the cloud platform that we have selected. AWS and DynamoDB are interacted through the command line for example to create and use a DynamoDB table. Performing a lot of checks like how to create, load, and test a new role that can execute Lambda functions and interact with DynamoDB was part of this project. At last, we created and deployed an API gateway and concluded the project by configuring how to connect our React code to an API Gateway endpoint.

Software has become part of our daily lives now. Sadly, most organizations actually cannot deliver software effectively in the way that they are supposed to, significantly less do so as at the speed they are expected to remain serious. For the individuals who wish to keep up, not to mention lead, software delivery and its functionality should be drastically improved.

This is when serverless Architecture comes into the picture, it is based on cutting edge public cloud benefits that auto-scale and charge just when it is utilized. At the point when it scales, capacity planning, expense management are automatically done, the outcome is software that is simpler to set up, keep up, and mostly up to 99% less expensive.

However, Serverless Architectures are new and hence require a change by the way we recently pondered about architectures and work processes. Our objective was to build and work on a serverless application, in one basic, incredible and exquisite experience.

Organizing Principles:

II.THEORITICAL/CONCEPTUAL FRAMEWORK

1. A Survey on Serverless Computing and its Implications for JointCloud Computing
2. Serverless Architecture - A Revolution in Cloud Computing
3. Serverless Computing: Current Trends and Open Problems
4. Potential Bottleneck and Measuring Performance of Serverless Computing: A Literature Study
5. Serverless Is More: From PaaS to Present Cloud Computing

III.RESEARCH QUESTION(S)/HYPOTHESIS

1. What applications are suitable for serverless?, What performance problems does serverless computing introduce? , What security issues does serverless computing raise?, What happens when jointcloud computing meets serverless?
2. What is the functioning standard of serverless figuring reference model
3. adjusted by AWS Lambda? [3] What are distinctive existing serverless stages from industry, the scholarly community, and open-source activities, and distinguishing proof of key attributes and use cases, and depict specialized difficulties and open issues.
4. This paper reviews the past and recent work in the serverless computing to identify possible bottlenecks and the scope of measuring performance of serverless computing, also how machine learning can be used to increase some performance features.
5. What are the computer technologies underlying serverless?, What is the status of the current serverless technology? , What can we expect from the field of the serverless computing in the foreseeable future?

IV.METHODOLOGIES/ALGORITHMS

1. For checking performance: Fast boot (Container cache, pre-warming, Container optimization, Looking for other abstractions) ; Communications (Optimizing the storage server, Optimizing the communication path), Security issues: abstraction for function.
2. Configuring AWS Lambda for responding the notifications from the Auto Scaling Group: AWS Lambda function was configured in a way in which snapshot was automatically taken and then a new AWS EC2 instance was attached that was launched by the auto scaling group.
3. Survey of serverless commercial platforms like AWS Lambda, Alpha release, Google Cloud Functions, Microsoft Azure Functions, IBM OpenWhisk on the basis of different characteristics like Cost, performance, limits, security, deployment, accounting , programming languages , monitoring and debugging. Also benefits and drawbacks, workloads, frameworks etc.
4. Bottlenecks and performance were determined on the basis of the following:
   - Serverless Computing
   - Performance Testing

- Machine Learning and leveraging it for performance Validation.

5. An application called ExCamera, which uses cloud computing in addition with various workflows to alter, change, and encode recordings with low latency and cost was utilized to respond to the inquiries posed by conveying ExCamera utilizing the IaaS model. They inferred that serverless registering model is on the ascent and is giving customary cloud foundation a run for their cash by dividing applications into short-running, progressively adaptable, and stateless capacities. However, serverless likewise presents new issues in both performance and security regions.

V.ANALYSIS AND RESULTS

1. By using serverless computing to implement the jointcloud model, a user can deploy functions at different clouds to enjoy the lowest prices and best performance simultaneously, high availability. The traditional VM abstraction used in today's clouds might not be the preferred method to implement jointcloud computing.
2. Two snapshots were created by Lambda function. This proves the successful execution of Lambda function of the auto scaling group. Otherwise, the Lambda function was either a failure or was not triggered
3. Is tooling for serverless generally unique in relation to existing arrangements?, the question to be asked is serverless platform can handle legacy code? , Is serverless on a very basic stage right now? , or if there shall be patterns dedicated for building serverless solutions? , Does serverless reach out traditional cloud service providers ?
4. There is a ton to be done in the comprehension of serverless of serverless platform performance in the area of clocking and network latencies to check the quality measures, language runtime, worker utilization, function code size, system performance of serverless platforms, event types, CPU allocation scaling. etc.
5. The fine granularity for expressing computation adds significant overhead to the resource management and scheduling layers. To overcome this, significant research efforts invested in co scheduling and orchestration for workflows of functions should be done. It allows enhanced access control, function-level auditing, and provenance for seamless, efficient GDPR compliance. The performance issues can be solved by the users by being able to more accurately monitor, profile, and schedule these fine-grained services.

VI.CONCLUSION

1. They concluded that serverless computing model is on the rise and is giving traditional cloud infrastructure a run for their money by splitting applications into short-running, dynamically scalable, and stateless functions. But serverless also introduces new problems in both performance and security areas.
2. The paper stated that serverless architectures can be considered as a new era of computation which can be adapted for use in much broader sense and there are possibilities of exploring more research avenues for the academic and research community in the arena of serverless computing.
3. The Function-as-a-Service (FaaS) model lends itself well to a number of common distributed application patterns, including compute-intensive there a wide variety of problems in this new technology ranging from infrastructure issues such as optimizations to the cold start problem to the design of a composable programming model.
4. The solution to improve performance-based issues/ bottlenecks and also to be able to measure is to use different kinds of artificial intelligence and machine learning for performance engineering.
5. Fine billing granularity, control and insight , the ability to run very fine arbitrary functions on request are offered by the current serverless technologies to its users however the drawback is that it works just on a few chosen applications.

VII.IMPLICATIONS FOR FUTURE RESEARCH

1. Will focus on unresolved problems in serverless computing and extend the serverless model to the jointcloud infrastructure.
2. There are many areas where this technology has it Challenges like hard-ware level, Management-level, business-level , developer level.

3. In what ways will serverless extend the original intent of cloud computing of making hardware fungible and shifting cost of computing from capital to operational expenses.
4. The solution to improve Performance based issues/ bottlenecks and also to be able to measure is to use different kinds of artificial intelligence and machine learning for performance engineering.
5. To make serverless computing available for everyone by removing the underlying drawbacks of this technology such as physical containerization.

REFERENCE

[1] Mingyu Wu, Zeyu Mi, and Yubin Xia (2020),” A Survey on Serverless Computing and its Implications for JointCloud Computing”, 2020 IEEE International Conference on Joint Cloud Computing (JCC)
[2] Dr. R. Arokia Paul Rajan, 2018, “Serverless Architecture - A Revolution in Cloud Computing“
[3] Ioana Baldini, Paul Castro, Kerry Chang, Perry Cheng, Stephen Fink, Vatche Ishakian, Nick Mitchell, Vinod Muthusamy, Rodric Rabbah, Aleksander Slominski, Philippe Suter, (2017), “Serverless Computing: Current Trends and Open Problems”, arXiv:1706.03178v1 [cs.DC] 10 Jun 2017
[4] Deepak Khatri, Sunil Kumar Khatri, Deepti Mishra, June, 2020, “Potential Bottleneck and Measuring Performance of Serverless Computing: A Literature Study “, 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO) Amity University, Noida, India. June 4-5, 2020
[5] Van Eyk, E., Toader, L., Talluri, S., Versluis, L., Uta, A., & Iosup, A. (2018), “Serverless Is More: From PaaS to Present Cloud Computing”, Delft University of Technology and Vrije Universiteit Amsterdam
[6] Comparison of Serverless architectures, Accessed on 2nd August 2018,https://dzone.com/articles/4-use-cases-of-serverless-architecture