Big Data Analytics Using Serverless Computing - A Personalized Recommendation System Case Study

Md. Mijanur Rahman, Md. Hasibul Hasan

Abstract: The enterprises face challenges to process and get insight from big data. It is very costly to compute and manage massive amounts of data. So, Serverless is such a technology that helps to analyze big data with low cost and high performance. Serverless cloud providers manage the operating system, servers, hardware and execute codes. Developers focus only on writing code rather than managing infrastructure. Serverless development has drawn a lot of attention in the market because customers only charged for executing code. Still, all users are not getting benefit from the Serverless technology due to appropriate solution architecture. Mostly an expert architect can ensure scalability, security, cost efficiency, etc. So users often pay for the right architecture to compute and store data in the Serverless cloud platforms. An architecture has been proposed with reduce charge and improve the performance of big data processing using Serverless technology and it is made using Amazon Web Services (AWS). The proposed architecture is evaluated using a real-world use case. As a case study, Movielens data are used in our model for personalized recommendation using Amazon personalized Hierarchical Recurrent Neural Networks (HRNN) algorithm.

Index Terms: Big Data Processing Architecture, Hierarchical Recurrent Neural Network (HRNN), Amazon Personalization, Serverless cloud computing, Amazon Web Service, Movielens, Scalability.

1. INTRODUCTION

In the prevailing industry, information are critical for the enterprise organization. The Big Data initiatives [14] and technology are used to analyze this large quantity of records for getting insights that may assist in making strategic decisions [1]. When the data sets are very large, for instance, of a Petabyte scale, managing that big data becomes very complex. We are facing many challenges to organize, process and compute this data like, collecting, storage, clustering, server management, algorithm deployment, etc [2]. Online users are facing difficulties regarding information overloaded. It is particularly an issue when a user has to quickly and accurately identify a resource from an exponentially growing set of information and products. It is also complex and burdensome for the companies to collect, process and merge the data to forecast. There are few serverless providers available in the market. So, using serverless service, a user's can create and deploy code without the knowledge of operating and managing infrastructure. [3]. In cloud computing, a user has to maintain servers but in the serverless scenario, a client manages functions. In traditional system, developer or user's need to manage the whole infrastructure first, then setup the environment as per the requirements of the system. But, serverless in a opposite model where cloud provider already setup the infrastructure and related environment for the users.

It is executing all the codes of the user's and dynamically distribute all the resources. Serverless codes are run in stateless containers and it is divided into wide range of events like, file uploads, Http requests, scheduled events, database events, etc. The main purpose of serverless is to focus on the application rather than infrastructure [5]. As, serverless executes function so, sometimes serverless called as "FaaS" or "Functions as a Service" [6]. Our main goal is to make a data lake but AWS serverless does not have all capabilities to make the data lake. Still, it is very high-priced to process and analyze data in the cloud. In the traditional system, it is more complex to process data because data are brought from various sources and prepare the data before processing using AWS glue also costly. The current big data analytics techniques in the AWS are very expensive. We are going to propose a customized serverless architecture using the AWS component. Our architecture for big data analytics will be easier and it reduces cost also, we would show how to implement, deploy and maintain big data analytics applications on AWS (Amazon Web Service) based on serverless technology. In addition to it, we will demonstrate our proposed architecture using large data sets for a personalized recommendation system. A recommendation system is a type of information filter, which can learn users' interests and hobbies according to their profile or historical behaviors, and then predict their ratings or preferences for a given item. It changes the way businesses communicate with users and strengthens the interactivity between them. The recommendation system has to face critical challenges as the systems have to offer accurate and real-time recommendations. It also performs a responsible role in considering the respective trade-off between the accuracy and respective computation time. These problems can be minimized by our proposed architecture. In this research, we will show a personalized recommendation system using Amazon Personalization Hierarchical Recurrent Neural Network (HRNN) Algorithm on users and items using our proposed architecture. The top cloud provider likes AWS (Amazon Web Service), GCP (Google Cloud Platform), IBM, Microsoft Azure, etc. provides Serverless Service, using their facilities, and we proposed a Serverless Architecture for the recommendation system [7]. We will also implement a recommendation system using our proposed architecture in AWS. Personalized recommendation systems typically use two sources of information: collaborative filtering based on user-item interaction histories and content filtering based on user/item features. Collaborative filtering techniques are used for user-item interaction since histories are often limited by the model capacity. In our paper, we will use AWS-HRNN Recipe as an algorithm which is the part of collaborative filtering based recommendation.

2 RELATED WORK

The growth of cloud computing programs and an increasing need for always-on, connected experiences, drives highly advanced IT usage and a push for groundbreaking technology. Consequently, there has recently emerged a cloud computing derivative known as serverless computing. It stacks against another type of similar technology that has long been popular: containers or container-based programming. However, adopting something new is not always the best idea because it is trendy. It may operate better and be more effective, but it is not necessarily the best solution for a team or organization. Straight away, understanding the word "serverless" does not imply

- *Md. Mijanur Rahman is currently working as an Assistant Professor at Southeast University, Department of CSE , Bangladesh, PH- +880 55034135*
  E-mail: mijanur.rahman@seu.edu.bd
- *Md. Hasibul Hasan is currently pursuing BSc degree program in Computer Science and Engineering in Southeast University, Bangladesh, PH-01889177778.*
  E-mail: hasan33@uwindsor.ca

that there are no servers or remote portals required in computing [8]. In fact, the truth is the exact opposite, as the technology still depends on remote servers based on the cloud. Why then is it called serverless? This is because all IT operations and maintenance are addressed by a third-party service provider. Within a cloud operating system, the main platform still lives, but the code is written and implemented independently. This technology enables the programming team to manage, create and distribute the application itself while remotely handling the hardware and system infrastructure. It gets rid of the burden of providing, powering, and maintaining remote hardware from the programming team and enables them to concentrate on their product and software development specialty [9]. That is basically why, especially through Amazon's Lambda, many regard serverless computing as the perfect solution. Big Data is a broad word for large and complex datasets where conventional tools for data processing are insufficient. It's quite complicated to integrate these large data sets. During this integration, several difficulties can be faced such as collection, analysis, data curation, search, visualization, privacy, sharing, and storage. Compared to the traditional relational database, the key components of the big data platform are managing data in new ways. Accurate management of big data will result in a more confident decision-making process. Serverless computing is becoming the most preferred solution in developing a big data model within an organization. Setting up a dedicated server to support your custom needs is now history. Organizations often redesign their Information Technology (IT) infrastructure to cope with complex analytic functions. Besides, the "pay-as-you-go" phrase proves to be a key driver in accelerating serverless architecture adoption [10]. It is becoming mainstream due to a sudden increase in acceptance as more and more individuals are choosing to use it. Serverless is certainly quite beneficial for organizations worldwide. However, very few take advantage of it to boost their big data programs.

Much of the expense of setting up on-site infrastructure with inexpensive commodity hardware was incurred during the early days of Hadoop processing. A Petabyte Hadoop cluster needed about 125-250 nodes that cost about $1 million. Each node's price is about $4000. Therefore, it costs about $32-40 per hour for operation and hardware. This price is per hour regardless of whether it is running or not. As cloud programs became more fully-fledged, organizations started to exploit cloud programs for Big Data challenges. It was then no longer necessary to maintain commodity hardware-based infrastructure. In addition, the pay-as-you-go system provided more flexibility in getting rid of the hardware maintenance requirement [11]. In a cloud model, the cloud instances were closed when the solution was attained, further lowering the price. Using Amazon's static cluster for 100 Terabyte (TB) data usage cost around $78,000, while using Amazon's EMR with S3 for the same quantity of data costs around $28,000 [17].

3 METHODOLOGY

a) HRNN Recipe

The hierarchical recurrent neural network (HRNN) Recipe is introduced by Amazon [12]. It can model the changes in the user behavior. The process of temporal modeling is closely linked with recommendation systems, as the interest and the intent of the user has a possibility of drifting away or altering with time. This feature is difficult to model with traditional methods. As a common style of factorization of machines, we can consider the manual process of discounting service for respective distant interactions (the distance calculated at the time). They are human effort-intensive, and they are subjected to inaccuracy. The respective Amazon Personalize HRNNs model can obtain requested user histories and it can perform perfect inferences. There is a gating mechanism on HRNN for modeling the respective discount weights as a function which can be used to get knowledge of the particular items and the related timestamps. The accuracy is increased by developing the temporal model efficiency of the hierarchical component. Based on the perspectives of the usage, amazon Personalize has extracted characteristics for each user by considering datasets that have been provided. If we have used the integration of real-time data, the respective features update also updated in real-time, so the user has to provide only user-id at the interface. The system does not show any error if we enter an item-id. The system does not consider the values, so values have not an impact on the result.

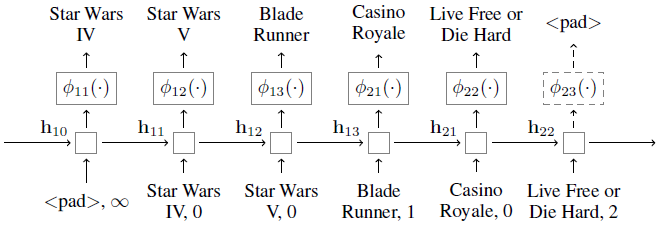


Figure 1 : *HRNN sequence model*

HRNN sequence model to predict the next item in a recurrent fashion. The second input position encodes a categorical hierarchy control signal which can be inferred from time-deltas [13].

b) Flow of the Complete Application

100K data has been loaded from external sources Movielens, and those data were loaded into S3 (Simple Storage Service) [15] bucket. Since data are brought from an external source so that those were in different formats. As data

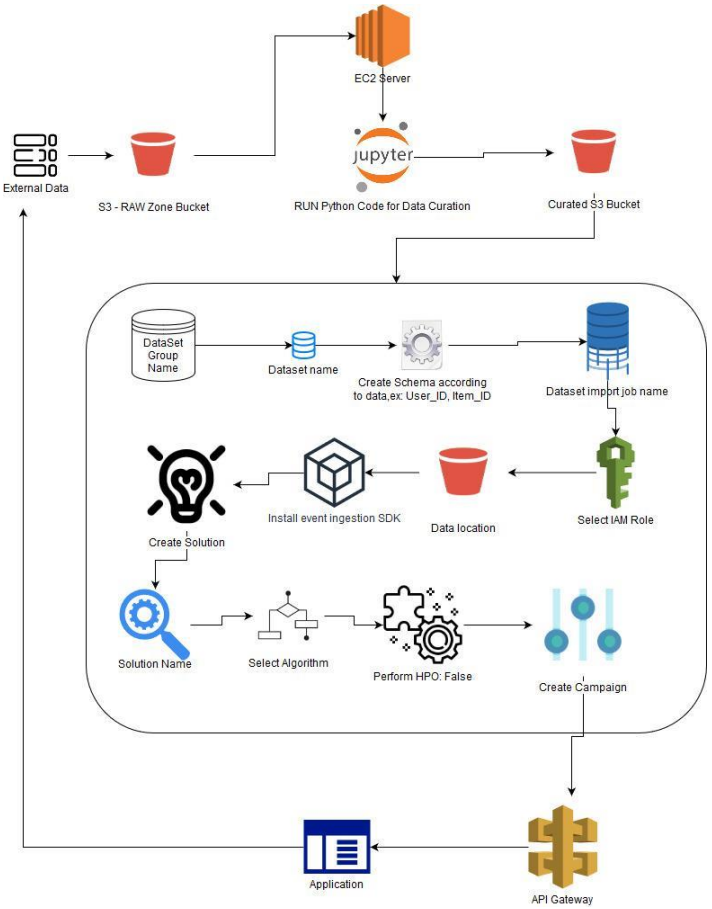


***Figure 2** : Application Flow*

in different formats, so It needs to be cleaned which is called data curation. For data curation, EC2 (Elastic Compute Cloud) Server is deployed [13] where Jupyter Notebook and Boto3 SDK are installed to run Python script [16]. The data source can be selected using Boto3 SDK from S3 and process that data in EC2 Server. After data curation, again it is stored those data in another S3 bucket. Now all of our data is in the same format, so we use cured data for the recommendation. For the recommendation, AWS(Amazon Web Service) provides a service called Amazon Personalize. There are six algorithms available for the recommendation system in Amazon Personalize. AWS-HRNN (hierarchical recurrent neural network) is used here which is able to model the changes in user behavior. Finally, a campaign is launched that publishes the trained algorithm which can be called REST API to get the recommendation on demand

4 PROPOSED ARCHITECTURE

This below section explains how the AWS Personalize architecture needs to be implemented.

1. Build a foundational data lake: Firstly data is loaded which is important for customer profile and details and external to the AWS environment needs to be loaded into S3 bucket. There can be multiple options to do so:
    a. Load Relational Data: Relational data that are stored in RDBMS etc. can be ingested using AWS DMS service into S3 partitioned by Date and Time.
    b. Load Large Scale Batch Data: Data which are as files - CSV, TSV, JSON, and XML can be loaded using script-based solution into S3 AWS Glue jobs is loaded to moderate workloads, use EMR for loading large scale data or use AWS Lambda to load small files.
    c. Load Data as Events: Event based data or messages can be loaded using a combination of AWS Kinesis and Lambda into S3 bucket.
2. Collect all data into S3 RAW Bucket: The above steps to ensure all the data from different systems are collected and saved into S3 RAW bucket that becomes the central repository of all external data.
3. Transform and Standardize: Step 3 is to standardize all data. XML, JSON data needs to be flattened; CSV, TSV data needs to be validated and enriched and RDBMS can be filtered to ensure correct format and structure of data is saved into S3 bucket. This steps is a custom step that can be achieved using AWS Glue or EMR based on the volume of data.
4. Build Curated Zone: All the standardized datasets are saved into S3 Curated zone bucket after applying previous step transformations and techniques. Again using PySpark or Python on AWS Glue or EMR seems a good recommendation.

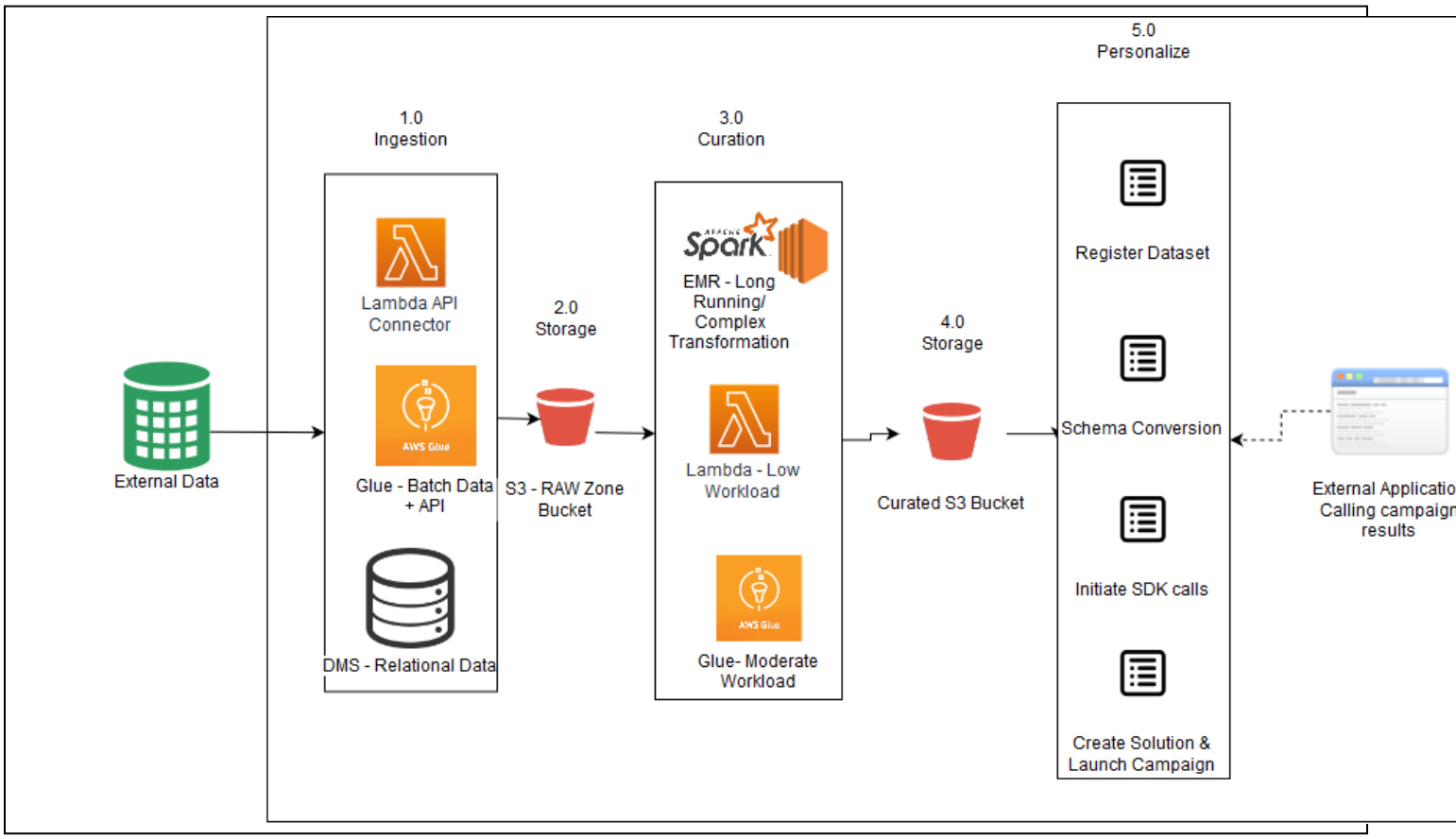


Figure 3: *Recommendation Architecture using AWS Personalize*

5. **Personalize: This step is the actual step to build the personalized algorithm and publish the same.**

Register the datasets: Every personalize use case starts with a business case, identify the business case and select the dataset from curated zone to be registered for the personalization process. This step can be achieved using Glue/EMR and the idea is to join all datasets to build a format that is similar to userID, ItemID, Rating. Schema Conversion: This step is the actual step where large datasets are collated and joined and eventually transferred into a schema that has values of userID, ItemID, and Rating

a. Thirdly, initiate boto3 sdk for personalization. Use the data sets and select the personalization algorithm from the SDK to train and test the personalized algorithm

6. **Finally, launch a campaign that publishes the trained algorithm which can be called using REST API to get the recommendation on demand.**

5 ANALYSIS AND RESULTS

We took one dataset from MovieLens website of 100837 users with userId, movieId, ratings, and timestamp. There are many files inside the dataset. So, we used some data that we need for our experiment. A data sample is given in the following after processing.

	userId	movieId	rating	timestamp
1	1	1	4	9.6E+08
2	1	3	4	9.6E+08
3	1	6	4	9.6E+08
4	1	47	5	9.6E+08
……	……	……	……	……
100834	610	168248	5	1.5E+09
100835	610	168250	5	1.5E+09
100836	610	168252	5	1.5E+09
100837	610	170875	3	1.5E+09

Table 1: *Dataset sample*

Result & Performance Analysis:

This is the user ID of the user you want to see campaign results for, which needs to be obtained from your user-interactions or user dataset. Here a random user id has been selected to its recommended items based on his/her purchase behaviors.

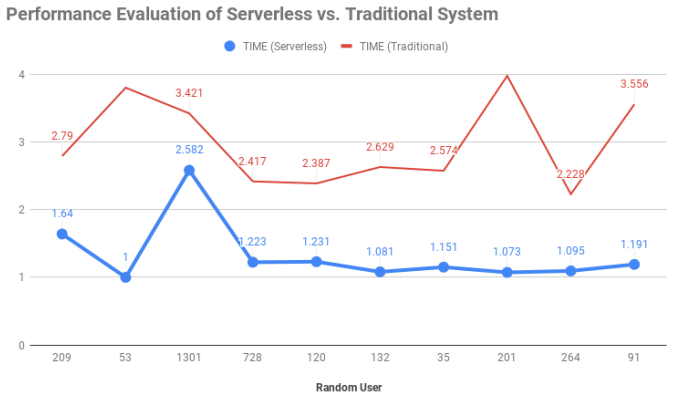
User ID:

53

Recommended items for user 10: 69122, 45517, 54001, 8528, 5630, 5481, 4025, …..

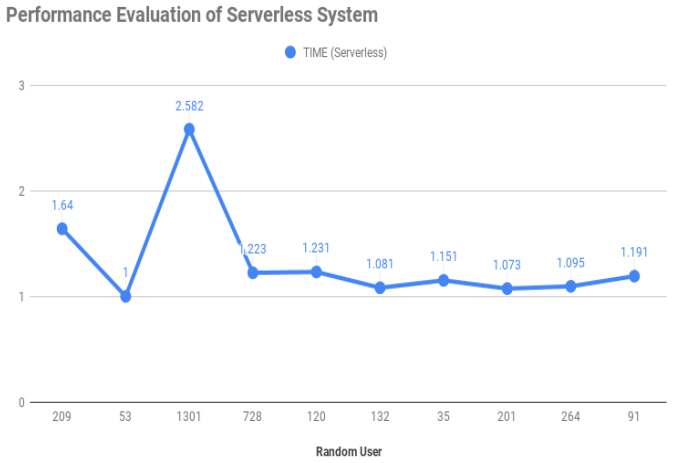
This model is evaluated for a few users who have been selected randomly among total users. Though it does not only depend on the

current platforms other factors are also involved. However, performance is measured which is shown in the below graph. it shows the time is taken to recommend items to a particular user in respect of number movie ratings only.



Graph 1: *Performance Evaluation of Serverless System*

In the Graph 1, X axis defines userID and in the Y defined recommended movie with times for execution.



Graph 2: *Performance Evaluation of Serverless System vs. Traditional System*

In the Graph 2, we generate performance evaluation of Serverless System and Traditional system. For example : userID : 91, recommendation execution time is 1.191, 3.556 on Serverless System and Traditional System respectively.

6 CONCLUSION

Serverless is an emerging technology to process a massive amount of data with low cost and high performance. It is only possible to apply appropriate solution architecture for computing data in the cloud platform. In this paper, architecture has been proposed to process big data in Serverless cloud computing. It has been implemented on Amazon Web Services using a real-world dataset for personalized recommendation system as a case study. Where, HRNN algorithm is used to recommend personalized items. The proposed architecture is made based on several services of AWS which may help for big data analytics. The result of our architecture performance is encouraging and it may help for continued study and implementation. It may be evaluated with different domain's dataset. Still, it is difficult to compare with state of the art result, which has not been found also the rapid change of the cloud infrastructure. In the future, It will be evaluated with more AWS services to show more comparative results with cost metrics.

ACKNOWLEDGEMENTS

This paper has been supported by Institute of Research & Training, Southeast University, Dhaka, Bangladesh.

REFERENCES

[1] "Why is Big Data Analytics So Important? - Whizlabs Blog." [Online]. Available: https://www.whizlabs.com/blog/big-data-analytics-importance/. [Accessed: 12-Dec-2019].W.-K. Chen, Linear Networks and Systems. Belmont, Calif.: Wadsworth, pp. 123-135, 1993. (Book style)

[2] "Top Big Data Challenges." [Online]. Available: https://www.datamation.com/big-data/big-data-challenges.html. [Accessed: 12-Dec-2019].K. Elissa, "An Overview of Decision Theory," unpublished. (Unplublished manuscript)

[3] "What Is Serverless Computing? | Serverless Definition | Cloudflare."[Online].Available: https://www.cloudflare.com/learning/serverless/what-is-serverless/. [Accessed: 12-Dec-2019].

[4] "What is Serverless Architecture? - Twilio." [Online].Available: https://www.twilio.com/docs/glossary/what-is-serverless-architecture. [Accessed: 12-Dec-2019].

[5] "What is Serverless Architecture? What are its Pros and Cons? - By Faizan Bashir." [Online]. Available: https://hackernoon.com/what-is-serverless-architecture-what-are-its-pros-and-cons-cc4b804022e9. [Accessed: 12-Dec-2019]

[6] "What is Serverless? | Serverless Stack." [Online]. Available: https://serverless-stack.com/chapters/what-is-serverless.html [Accessed: 12-Dec-2019]

[7] G. McGrath and P. R. Brenner, "Serverless Computing: Design, Implementation, and Performance," Proc. - IEEE 37th Int. Conf. Distrib. Comput. Syst. Work. ICDCSW 2017, pp. 405–410, 2017.

[8] G. Adzic and R. Chatley, "Serverless computing: economic and architectural impact," pp. 884–889, 2017.

[9] S. Chaudhary, G. Somani, and R. Buyya, "Research Advances in Cloud Computing," Res. Adv. Cloud Comput., pp. 1–465, 2017.

[10] W. Lloyd, S. Ramesh, S. Chinthalapati, L. Ly, and S. Pallickara, "Serverless computing: An investigation of factors influencing microservice performance," Proc. - 2018 IEEE Int. Conf. Cloud Eng. IC2E 2018, pp. 159–169, 2018.

[11] "HRNN Recipe - Amazon Personalize." [Online]. Available: https://docs.aws.amazon.com/personalize/latest/dg/native-recipe-hrnn.html. [Accessed: 12-Dec-2019].

[12] Y. Ma, "Hierarchical Temporal-Contextual Recommenders," no. 2014, 2018.

[13] "Amazon EC2." [Online]. Available: https://aws.amazon.com/ec2/. [Accessed: 19-Jan-2020].

[14] N. Elgendy and A. Elragal, "Big Data Analytics : A Literature Review Paper Big Data Analytics : A Literature Review Paper," no. September 2014, pp. 214–227, 2018.

[15] "Cloud Object Storage | Store & Retrieve Data Anywhere | Amazon Simple Storage Service (S3)." [Online]. Available: https://aws.amazon.com/s3/. [Accessed: 19-Jan-2020].
[16] "AWS SDK for Python." [Online]. Available: https://aws.amazon.com/sdk-for-python/. [Accessed: 19-Jan-2020].
[17] "Can Serverless computing reshape big data and data science?" [Online]. Available: https://dashbird.io/blog/serverless-computing-reshape-big-data-data-science/. [Accessed: 19-Jan-2020].