Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 6, June : 2024

PROACTIVE RESOURCE MANAGEMENT FRAMEWORK USING ARIMA FOR WORKLOAD PREDICTION IN CLOUD COMPUTING SERVICES

Krishan Kumar[1], K. Gangadhara Rao[2], Suneetha Bulla[3]
[1]Research Scholar, Department of CSE, Acharya Nagarjuna University, Guntur, India, krishan0405@gmail.com
[2]Professor, Department of CSE, Acharya Nagarjuna University, Guntur, India. kancherla123@gmail.com
[3]Associate Professor, Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswram, Guntur, 522502, AP, India. suneethabulla@gmail.com

Abstract: Resource management is a strategy of the elastic cloud to provide availability of service for the end-users. It improves the runtime performance of services; two aspects of technical issues need to be addressed. The first one is the balancing of a large amount of data on existing resources and the second is resource provisioning which can adjust the number of resources optimally to adapt the time-varying workload. As the growth of data is increasing tremendously, efficient resource management is the need in cloud computing. This paper proposes a cloud framework Resource Provisioning framework through workload prediction to process data in automation with adaptive resource and workload management strategy. The proposed architecture is intended to provide resources dynamically and efficiently satisfying the demands of the user. To achieve this objective of efficient resource provisioning, algorithms are developed for workload prediction which helps in deciding optimum resource provisioning. Our system uses a proactive approach resource management and deployment of the adaptive cloud system. In traditional systems, resources are managed based on demand, availability and the strategy of scheduling which results in delayed response time at large. We configured the ARIMA model to predict the future workload for provisioning the resources dynamically and remove the problem of over-provisioning and under-provisioning in a cloud environment. Finally, results are drawn, and conclusions are presented.

1. Introduction

Cloud computing is a service that provides customers with access to resources such as CPU, software, hardware, information, and devices over the Internet. This technology employs utility mechanisms, such as autoscaling and on-demand service deployment, which are widely acknowledged by analysts [1]. Utilizing virtualization, the cloud computing system establishes an environment encompassing both homogeneous and heterogeneous operating systems [2-4].

Various cloud providers cater to the Quality of Service (QoS) needs of IT sector end-users, responding to their changing demands over time. However, this dynamic user behavior can lead to inefficient resource management, causing fluctuations in the number of users accessing the services. Consequently, users might experience periods of resource abundance and scarcity, resulting in subpar service quality and increased costs. Many researchers have proposed different strategies to address this challenge, aiming to optimize resource utilization while minimizing expenses.

This study introduces an innovative approach to building an adaptive cloud that addresses resource provisioning in cloud computing. The focus is on efficiently provisioning resources to end-users, enhancing performance, and ensuring user satisfaction through a QoS-based approach. To handle the

exponential growth of data, a cloud architecture with an adaptive resource management method is developed, enabling automated data processing.

The system design progresses through several stages, including data synchronization between cloud services, cloud creation for resource management, integration of user data, resource management, and service delivery with performance optimization. To anticipate resource demands effectively, a prediction model is proposed based on historical database observations. The Enhanced ARIMA for Resource Provisioning (E-ARIMA) model is employed for resource allocation to end-users according to their specific needs. A user demand-based framework and method for workload prediction in cloud computing are suggested and implemented. This includes constructing a prediction mechanism and applying it to calculate the workload for various time periods based on historical data, specifically employing the E-ARIMA model for assessment.

This paper proposes Resource Provisioning framework to predict workload for effective resource provisioning in an adaptive cloud. The prototype leverages a prediction model and integrates machine learning and cloud platform to efficiently provide resources. The utilization of a best-fit algorithm for prediction techniques in resource management stands out as a key finding in this study.

The rest of the paper is organized as follows. Section 2 describes the related work for the proposed approach. Section 3 explains prediction mechanism applied for proposed approach. In Sect. 4, experimental setup is described. Section 5 presents results with discussion. Finally, proposed work is concluded with future work.

2 Literature review

To ensure effective allocation of resources, the utilization of a predictive approach holds substantial importance [5]. As delineated in Figure 1, this predictive methodology can be segmented into two distinct parts: anticipation of application behavior and prediction of host load. Accurately predicting the load on hosts within cloud systems is a critical stride towards attaining service-level agreements. Employing a Bayesian model, a technique for forecasting precise host load over extended time frames has been identified [6]. This approach to host load prediction is subsequently categorized based on considerations of workload and performance.

The capability of a cloud workload analyzer extends to forecasting the load on hosts, drawing insights from a variety of factors including incoming requests, resource usage patterns, and resource requirements. The assessment of host load prediction effectiveness can be carried out by monitoring performance indicators such as response time, CPU usage, data throughput, and memory utilization. Another essential aspect of achieving resource efficiency in the cloud environment is the anticipation of application behavior to forecast upcoming resource needs. This application prediction process encompasses various facets, including performance expectations, quality of service attributes, workload projections, and SLA metrics. Within the context of performance prediction, specific parameters like response time, CPU utilization, data throughput, and memory usage are estimated subsequent to the allocation of resources [7].

Larsen et al. [26] have focused on resource prediction within cloud computing. This patent delineates methodologies for forecasting processing resources within a cloud computing module, rooted in

predefined tasks. The approach is employed to predict resource allocation by considering the dataset and input parameters. Similarly, Daniel et al. [27] have contributed a patent introducing a predictive auto-scaling engine that utilizes prediction models to scale distributed applications. This involves the analysis of historical performance data, which is then amalgamated through monitoring and the identification of scaling patterns for the application [28–30]. In comparison with the existing methodology, our approach has exhibited superior performance in terms of accurate prediction. The underlying architectural proposal is designed to dynamically and efficiently provision resources, catering to user demands. In pursuit of this objective of streamlined resource allocation, algorithms have been devised for workload prediction, thereby aiding in the identification of optimal resource provisioning strategies.

Our methodology embraces a proactive approach to resource management and the establishment of an adaptive cloud system. In contrast, conventional systems rely on demand, the availability of resources, and scheduling tactics, often leading to significant delays in response times [31]. In this regard, Kumar et al. [32] have introduced a tailored scheduling algorithm for workflows, aiming to enhance the utilization of resources and minimize the processing duration (make-span) within the cloud environment. Furthermore, Tyagi et al. [33] have outlined a scenario involving real-time soil monitoring through cloud computing with sensor-based technology. Their work introduces a hierarchical architectural framework that clusters both sensors and cloud resources, specifically designed for agricultural applications. These principles are visually elucidated in Figure 2, where considerations regarding prediction prerequisites, evaluation parameters, and the attributes of prediction are expounded within the context of application prediction strategies.

The evaluation of performance parameters for application prediction involves the assessment of estimates, success rates, error rates, and cost/profit considerations. The success rate delineates the precision of predicting future behavior using the employed prediction method. This metric can be calculated by comparing the number of accurate predictions to the total number of predictions made [34].

The assessment of predictive properties involves the scrutiny of accuracy, adaptability to resource needs, proactiveness, and the establishment of resource mapping within the cloud environment. The accuracy of a prediction model could potentially qualify it as a best-fit model based on its precision. To effectively cater to the evolving user requirements, the prediction model assumes a pivotal role in anticipating future resource demands. Resource adaptation functions to dynamically allocate resources in response to changing requirements. In contrast to reactive models, predictive models must possess a proactive nature to foresee future resource demands accurately. Prasad et al. [35] have elucidated the necessity of resource allocation in cloud computing, exploring diverse methodologies. They have also delved into various policies and scheduling algorithms within resource allocation models, discerning performance benchmarks for resource scheduling and allocation. To ensure the appropriate provisioning of resources, it is imperative to establish a connection between resource prediction and resource provisioning strategies through resource mapping.

3. System architecture

Utilizing a machine-learning technique, the prediction approach is notably more suitable for a proactive strategy than alternative methods such as control theory or queuing models. The objective is to establish an adaptive cloud system capable of autonomously supplying applications with necessary

resources, eliminating the need for manual intervention. The detailed architectural flow is depicted in Figure 3. Within this proposed framework, various phases are concurrently described, focusing on the processing of extensive data applications within the Hadoop ecosystem. The Hadoop ecosystem cluster is established within the cloud environment. For efficient resource provisioning and optimal resource utilization, the realm of cloud computing is of paramount significance. It orchestrates resource allocation to specific applications in accordance with their requirements [36, 37].

In this study, an ARIMA-based workload prediction is executed by assimilating workload data from both past and present time periods. The workload analyzer is responsible for promoting and storing updated information to facilitate the prediction model. During instances of commissioning or decommissioning virtual machines (VMs) as part of the service, the workload analyzer communicates with the VM manager to allocate or release resources from the resource pool, thereby mitigating instances of under or over-provisioning. Any unexecuted requests within the system architecture are buffered for consideration in subsequent iterations. The system architecture is delineated across four distinct phases, providing a comprehensive overview.

Phase 1: Data synchronization across cloud services
Phase 2: Formation of cloud for resource management
Phase 3: Integration of cloud with user data and resource management
Phase 4: Service provisioning with improvement of performance.

The above-mentioned phases are systematically depicted in Figure 1. In the initial phase (Phase 1), a continuous stream of user requests is generated. These requests are collected during the data acquisition stage and subsequently stored within the cloud database. Following each data acquisition stage, historical databases are updated for offline analysis. The stored database is then relayed to the data repository to be extracted in the required format. In the subsequent iteration, this data is transformed and distributed into a suitable format, enabling the generation of tasks intelligible to the task dispatcher. The task dispatcher stage oversees the distribution of tasks, forwarding them to the workload manager for workload details and subsequent arrangement in the scheduled queue. As depicted in Equation (1), criteria are captured to compare the allocated resources with the actual demand.

$$m(w) = r. \qquad (1)$$

In this equation, "m" represents the matching function, "w" signifies workload intensity, and "r" denotes the quantity of resources. The matching function yields the minimal amount of resources "r" corresponding to a specific resource type. For a given workload intensity "w," the workload units are quantified as the number of requests. The consumption of resources ("r") is gauged during fluctuations in workload intensity ("w"), which, in turn, directs the allocation or deallocation of resources. This is contingent upon changes in the workload intensity value, either an increase or decrease. The system requires an adequate time span to effectively allocate or deallocate resources for each workload intensity value.

During the cloud resource management phase, cloud resources are commissioned or decommissioned from the resource pool based on various resource features such as name, type, configuration, availability, utilization, usage, and cost. In order to facilitate the accurate provisioning or de-provisioning of resources based on workload information, a prediction model is implemented to predict

the precise resources necessary for identified workloads. The QoS (Quality of Service) metrics are evaluated by the CloudWatch monitoring event within the QoS manager stage. The QoS manager assesses and evaluates these metrics during the integration phase. Depending on the outcomes of resource management deployment and integration phases, services are provisioned to users in the form of responses tailored to their specific requirements.

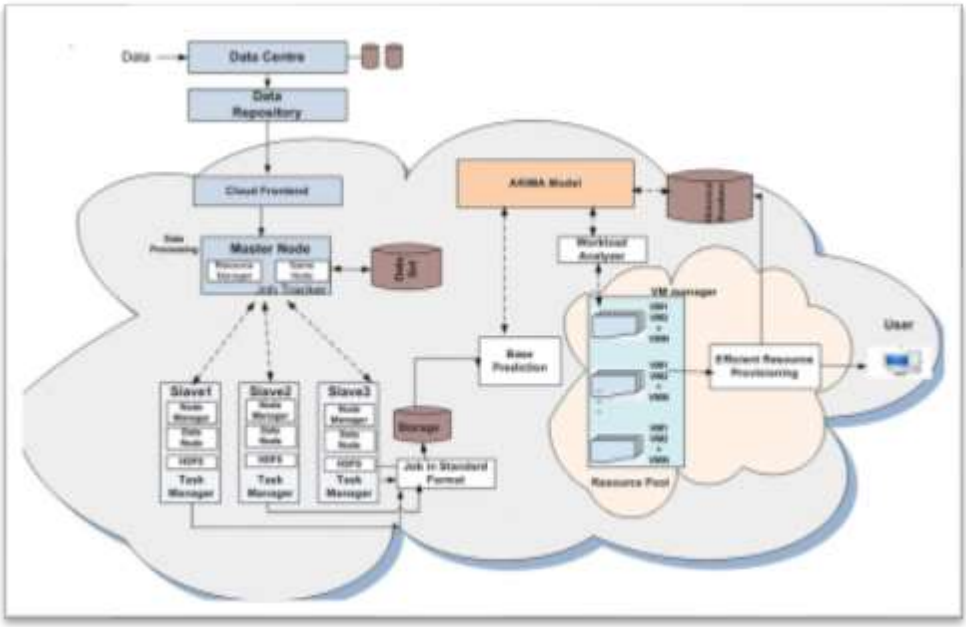

Fig: 1: E-ARIMA model

4. Prediction mechanism for resource management

The system involves sending web requests to cloud servers to verify and retrieve the necessary resources from specified regions. These continuous logs contain the count of HTTP requests for each resource within the time interval "t - 1" and "t". Leveraging the current and historical database, the workload for the time intervals "t - 1" and "t" is computed. To observe request patterns during these intervals, we analyzed the arrival of requests from the years 2003 to 2014. This span serves as the original time series data, utilized in the training phase to predict the subsequent intervals. To incorporate the requests, they are integrated with the time series values of ARIMA for parameters "p," "d," and "q."

The ARIMA (Autoregressive Integrated Moving Average) model is a commonly employed technique in time series analysis and forecasting. Typically denoted as ARIMA (p, d, q), where "p" represents the autoregressive components' order, "d" is the differencing duration, and "q" corresponds to the moving average term's order. Each of these components aims to minimize the residuals produced in the final step of ARIMA modeling. This entails passing time series data through these components sequentially to achieve the smallest residual. Caglan et al. [38] have patented a System for the allocation of network resources using an autoregressive integrated moving average method, predicting future network parameter values through resource allocation. For improved performance, the ARIMA model is adopted. The Mean Square Error (MSE) is used to quantify the disparity between actual and predicted values. David et al. [39] have patented a rapid and automated ARIMA model initialization

process. ARIMA is employed to generate time series data, involving the determination of the difference order for the time series data.

The class diagram depicting prediction-based resource management is showcased in Figure 5. The Resource Manager class contains resource attributes that provide insights into the status of virtual machine (VM) instances. The VM Info class offers information about available VMs to the Resource Manager class. VM Info class is linked to two subclasses: VM Provisioner and VM Destroyer. These subclasses handle creation and termination operations, respectively, guided by feedback from the Workload Analyzer class. The Data Acquisition class is responsible for storing the extracted input data. It forwards this input as a task to the Resource Manager class to obtain the requisite resources. The QoS Manager class oversees the workload status and utilizes it to predict future workload, facilitating advanced resource allocation. The IaaS Services class initiates AWS Cloud to provision or deprovision cloud services [40, 41]. To evaluate predictions, the best fit ARIMA model algorithm is outlined through the following steps;

Step 1: Test dataset is iterated for each time interval.
Step 2: For each iteration, a new ARIMA model is trained for all historical data.
Step 3: Prediction is done for the next time interval.
Step 4: Predicted data is saved for next iteration to use as a historical data and steps 1–4 are repeated for next interval.
Step 5: Selection of best-fit model is done by calculating the root mean squared error (RMSE) and maximum likelihood. RMSE [42] measures the average magnitude of the error using quadratic scoring rule. It is the square root of the average of squared differences between prediction and actual observation as shown in the Eq. (1).

$$RSME = \sqrt[3]{\frac{1}{N}\sum_{t=1}^{n}(Y_t - Y1_t)^2} \qquad (1)$$

Where “N" represent the number of observations, "Y_t" signify the actual value, and "$Y1_t$" denote the predicted value. To anticipate the workload for a specified period, we seek to identify the best-fitting prediction model within the ARIMA framework through the utilization of the maximum likelihood function. To achieve this, we employed the maximum log likelihood method [43], which we will succinctly outline below to elucidate the most suitable prediction model. For expeditious likelihood estimation, the Kalman filter algorithm serves as a recursive procedure for computing one-step-ahead prediction errors and their respective variances. The computation of the likelihood function employs the Kalman filter algorithm [44] through three main steps: (1) forecasting future states based on current state information, (2) predicting new observations, and (3) updating state estimations when new observations are incorporated into the system. In this context, we have made an assumption regarding the series' stationarity, denoted as {W}, which guides the parameter estimation for an ARIMA model.

$$\mathfrak{F}(x, y) = \mathfrak{F}(x)\mathfrak{F}(y|x) \qquad (2)$$

By taking $x = \mathcal{W}_1 and\ y = \mathcal{W}_2, \ldots, \mathcal{W}_T$ Eq. (3) can be written as,

$$\mathfrak{F}(\mathcal{W}_T) = \mathfrak{F}(\mathcal{W}_1)\mathfrak{F}(\mathcal{W}_2, \ldots, \mathcal{W}_T|\mathcal{W}_1)$$

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 6, June : 2024

To minimize the prediction error , let ε_t as prediction error for $\mathcal{W}_T$ which is presented in Eq. (4)

$$\varepsilon_t = \mathcal{W}_T - \mathcal{W}_{T|T-1}$$

For AR(1), assume series of zero mean and size T by known parameter ∞ as defined in the Eq. (5),

$$\mathcal{W}_T = \infty \mathcal{W}_{T-1} + a_T$$

Where T=2……..n and a_T is the innovation model for step ahead prediction error as shown in the Eq. (6),

$$a_1 = \mathcal{W}_1 + \infty \mathcal{W}_{T-1}$$

In the one step ahead prediction error, innovation model for a_1 is defined as,

$$a_1 = \mathcal{W}_1 + \infty \mathcal{W}_0$$

So that for T=2……n the variance of the one step ahead prediction error is σ^2 as defined in the Eq. (7),

$$var(\mathcal{W}_T|\mathcal{W}_{T-1}, \ldots . \mathcal{W}_1) = \sigma^2 V_{T|T-1} \quad (7)$$

Where AR (1):

$V_{T|T-1}$=1 for T=2……n= $(1-\infty^2)^{-1}$ for T=1

Based on the prediction error decomposition joint density function for general ARIMA process [36] can be written as defined in the Eq. (8); To evaluate the likelihood function to reduce the problem of one step ahead.

$$\mathfrak{F}(\mathcal{W}_T) = \prod_{T=1}^{n} \sigma^{-1} V_{T|T-1}^{-1/2} (2\pi)^2 exp\left(\frac{1}{2\sigma^2} \sum_{T=1}^{n} \frac{(\mathcal{W}_T - \mathcal{W}_{T|T-1})^2}{v_{T|T-1}}\right) \quad (8)$$

In order to consider the provisioning/deprovisioning of resources, proactive prediction approach is the key feature. Prediction based proactive approach might work optimally to manage the resources in the cloud environment for varying workload [45]. Using the best-fit workload model, workload for next interval might be predicted accurately by getting minimum residual and maximum log likelihood value. Log likelihood function and root mean squared error is evaluated using the Eqs. (9) and (2). Used notations in the equations are described in the Table 1.

Table 1: Notations

Notations	Descriptions
y_t	Original time series
ε_t	Prediction error
β	Likelihood
$\mathcal{W}_T$	Stationary series
a_T	Innovation model

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 6, June : 2024

v_T	Variance
Ar	Autoregressive term
Ma	Moving average term
σ	Standard deviation
X_t	Partial autocorrelation coefficient (PACE)
P	Prediction function
N	Number of observation
T	Time
d	Square difference
R	Number of resources
W	Workload intensity
M	Matching Function

6 Results and discussion

The ARIMA(1, 1, 1) model has been identified as the most fitting model among various ARIMA models, determined based on considerations such as Akaike's Information Criterion (AIC), corrected AICC, and Bayesian Information Criterion (BIC) values. These values undergo continuous updates upon the arrival of new requests, while old values are relocated from the current database to a historical database for future training phases. Through training this database, the predicted future demand for the upcoming year is derived. The outcomes stemming from these projected values for diverse metrics are presented in Table 3. A forecast unit is established by employing the number of requests, with a total of 145 observations utilized to generate forecasts for different ARIMA models employing distinct series.

The evaluation summary of the forecast models encompasses a total of 36 forecasts, each associated with varying metric values aligned with their configured estimates. The forecast model summaries for ARIMA (1, 1, 1), ARIMA (1, 0, 1), ARIMA (5, 0, 2), and ARIMA (5, 0, 0) are defined and subsequently juxtaposed in Table 3 in relation to error metrics. Extending the database's training to the upcoming period of 2015–2017, predictions for various standard errors are documented in Table 4. Upon analyzing the prediction accuracy calculations, it becomes evident that the proposed prediction approach yields more precise results compared to conventional methods.

Table 2: Estimation summery of ARIMA model

Metrics	**Estimation ARIMA (1,1,1)**	**Estimation ARIMA (1, 0,1)**	**Estimation ARIMA (5, 0, 2)**	**Estimation ARIMA (5,0,0)**
ME	0.01166989	0.01173753	-0.01631024	0.00019176
RMSE	0.07450627	0.07476611	0.07465462	0.110378
MAE	0.05526755	0.05564004	0.05641946	0.08351552
MPE	0.4306297	0.4306297	2.351555	50.55343
MAPE	-54.28284	2.351555	104.1008	144.0441
MASE	0.6783021	0.6109884	0.3806085	144.0441
ACFI	0.003732132	0.004404706	0.02803195	0.005862
σ^2	-0.1121943	0.005669	0.005869	0.01263

Log likelihood	167.42	166.42	163.42	108.66
AIC	328.83	-328.83	-310.84	205.32
AICc	328.66	-328.66	309.75	204.7
BIC	319.94	-319.94	287.19	187.63

Table 3: Standard error for ARIMA model (prediction)

Year	ARIMA (1,1,1)	ARIMA (1,0,1)	ARIMA (5,0,0)	ARIMA (5,0,2)
2015	0.084267686	0.088860657	0.24622095	0.144455569
2016	0.094540256	0.0901684	0.264308967	0.151261095
2017	0.103416887	0.0901684	0.2644302	0.1512611
Average of SE	0.094074943	0.089732489	0.258320039	0.148992588

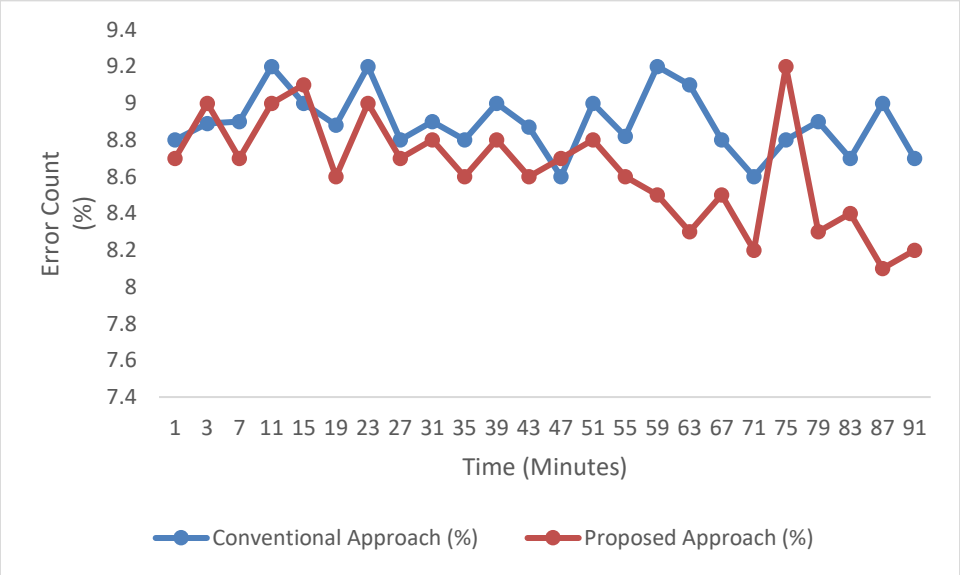

Fig 3: Prediction Error Count

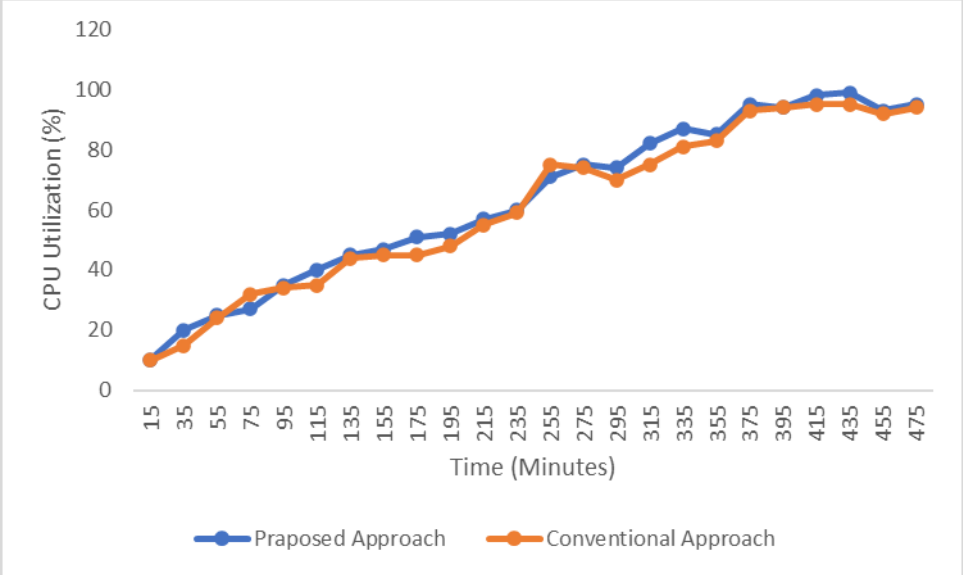

Fig 4: CPU Utilization

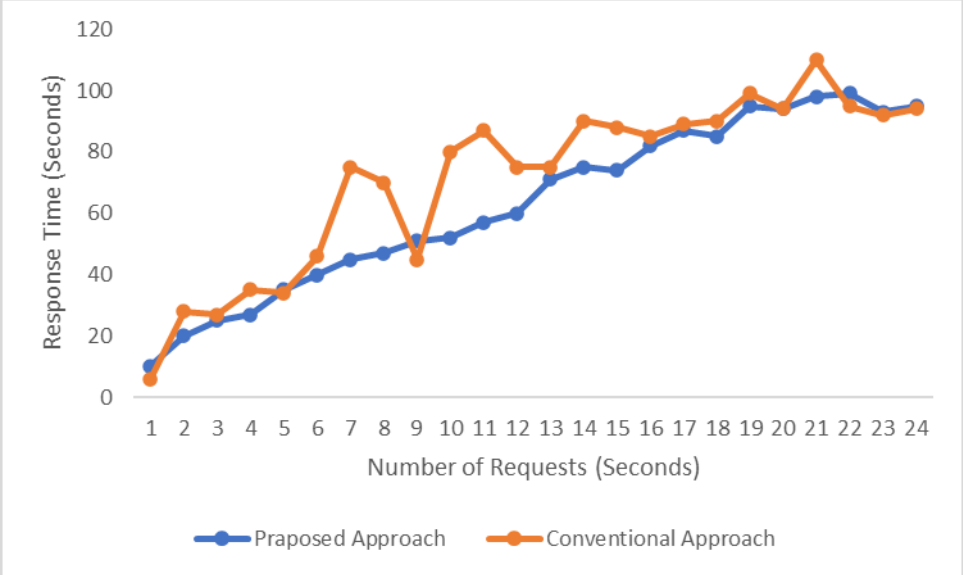

Fig 5: Response Time

As depicted in Figure 3, a comparison is drawn between the prediction error count for the quantity of resources per user request in both the conventional approach and the proposed approach. The heightened accuracy in prediction yields a reduction in the rejection rate of user requests. This, in turn, enhances the processing volume of requests and streamlines the execution time of processes. Notably, a decrease in the waiting time for new resource allocations contributes to the decline in rejected requests, subsequently fostering maximum resource utilization efficiency.

To gauge the capacity of serving user requests per minute, the CPU utilization metric is employed. The CPU utilization observed during the proposed approach proves notably more efficient than that of the conventional approach, as evidenced in Figure 4. Our evaluation underscores that our approach

attains a remarkable 91.11% accuracy in prediction, aligning well with the goal of efficient resource utilization to meet demand workloads.

When considering the execution time for processing requests and the subsequent receipt of acknowledgment in the form of responses to users, a crucial metric is the number of requests handled per second. The proposed approach emerges as superior, processing a larger volume of requests in comparison to the conventional approach. This distinction is visually portrayed in Figure 5 through the response time representation for the number of requests per second.

7 Conclusion

In this work, comprehensively addresses diverse facets of efficient resource provisioning within the domain of cloud computing. As a novel contribution, we have introduced an integrated methodology aimed at constructing an adaptive cloud system. This adaptive cloud system optimally allocates resources to end-users, culminating in enhanced performance. The architectural framework delineates various phases, encompassing data synchronization across cloud services, the establishment of a resource management cloud, integration of the cloud with user data, resource management, and service provisioning, all underscored by performance enhancement considerations.

Our proposed approach involves the application of a prediction model, strategically forecasting resource demands in advance to facilitate efficient resource provisioning based on historical and observed databases. We have formulated the ARIMA Workload Prediction for Efficient Resource Provisioning (E-ARIMA) model, tailored to efficiently provision resources to end-users, catering to their specific requirements via accurate prediction strategies. Calculations concerning prediction accuracy underscore the superiority of our proposed prediction approach in comparison to conventional methods. The count of prediction errors for multiple resources per user request validates the efficacy of our proposed strategy.

Future endeavors will delve into more in-depth mechanisms, exploring new proposals that incorporate emerging concepts such as the Internet of Things with cloud computing. Our trajectory will entail the development of a more adaptive system for end-users, potentially exploring alternative proactive machine learning approaches in subsequent studies.

Reference

1. Ji, C., Li, Y., Qiu, W., Awada, U., Li, K.: Big data processing in cloud computing environments. In: 12th International Symposium on Pervasive Systems, Algorithms and Networks, pp. 17–23, San Marcos (2012)
2. Li, C., Zhuang, H., Lu, K., Sun, M., Zhou, J., Dai, D., Zhou, X.: An adaptive auto-confguration tool for Hadoop. In: 19th International Conference on Engineering of Complex Computer Systems, Washington, pp. 69–72 (2014)
3. Ficco, M.: Security event correlation approach for cloud computing. Int. J. High Perform. Comput. Netw. 7, 173–185 (2013)

4. Banu, M., Aranganathan, A.: Study of load optimization and performance issues in cloud. Ind. J. Electr. Eng. Comput. Sci. 11(3), 1035–1041 (2018)
5. Amiri, M., Mohammad-Khanli, L.: Survey on prediction models of applications for resources provisioning in cloud. J. Netw. Comput. Appl. 82, 93–113 (2017)
6. Di, S., Kondo, D., Cirne, W.: Host load prediction in a Google compute cloud with a Bayesian model. In: SC '12: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, pp. 1–11, Salt Lake City (2012)
7. Cuomo, A., Rak, M.: Villano, U, Performance prediction of cloud applications through benchmarking and simulation. Int. J. Comput. Sci. Eng. 11(1), 46–55 (2015)
8. Song, W., Xiao, Z., Chen, Q., Luo, H.: Adaptive resource provisioning for the cloud using online bin packing. IEEE Trans. Comput. 63(11), 2647–2660 (2014)
9. Singh, S., Chana, I.: Resource provisioning and scheduling in clouds: QoS perspective. J. Supercomput. 72(3), 926–960 (2016)
10. 10. Liang, Q., Zhang, J., Zhang, Y.H., Liang, J.M.: The placement method of resources and applications based on request prediction in cloud data center. Inf. Sci. 279, 735–745 (2014)
11. 11. Yang, J., Liu, C., Shang, Y., Chen, B., Mao, Z., Liu, C., Niu, L., Chen, J.: A cost- aware auto-scaling approach using the workload prediction in service clouds. Inf. Syst. Front. 16(1), 7–18 (2014)
12. Chang, Y.C., Chang, R.S., Chuang, F.W.: A predictive method for workload forecasting in the cloud environment. In: Lecture Notes in Electrical Engineering, vol. 260. Springer, New York, pp. 577–585 (2014)
13. Jiang, Y., Perng, C.-S., Li, T., Chang, R.N.: Cloud analytics for capacity planning and instant VM provisioning. IEEE Trans. Netw. Serv. Manag. 10(3), 312–325 (2013)
14. Alasaad, A., Shafee, K., Behairy, H.M., Leung, V.C.M.: Innovative schemes for resource allocation in the cloud for media streaming applications. IEEE Trans. Parallel Distrib. Syst. 26(4), 1021–1033 (2015)
15. Amiri, M., Derakhshi, F., Reza, M., Khanli, L.: IDS ftted Q improvement using fuzzy approach for resource provisioning in cloud. J. Intell. Fuzzy Syst. 32, 1–12 (2016)
16. Garg, S.K., Toosi, A.N., Gopalaiyengar, S.K., Buyya, R.: SLA-based virtual machine management for heterogeneous workloads in a cloud data center. J. Netw. Comput. Appl. 45, 108–120 (2014)
17. Jheng, J.-J., Tseng, F.-H., Chao, H.-C., Chou, L.-D.: A novel VM workload prediction using Grey forecasting model in cloud data center. In: International Conference on Information Networking, pp. 40–45, Phuket (2014)
18. Yin, J., Lu, X., Chen, H., Zhao, X., Xiong, N.N.: System resource utilization analysis and prediction for cloud based applications under bursty workloads. Inf. Sci. 279, 338–357 (2014)
19. Lu, C.T., Chang, C.W., Chang, J.S.: VM scaling based on Hurst exponent and Markov transition with empirical cloud data. J. Syst. Softw. 99, 199–207 (2015)
20. Sheng, D., Cho Li, W., Cappello, F.: Adaptive algorithm for minimizing cloud task length with prediction errors. IEEE Trans. Cloud Comput. 2(2), 194–207 (2014)
21. Hu, Y., Deng, B., Peng, F., Wang, D.: Workload prediction for cloud computing elasticity mechanism. In: 2016 IEEE International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), Chengdu, pp. 244–249 (2016)
22. Akindele, A.B., Samuel, A.A.: Predicting cloud resource provisioning using machine learning techniques. In: 2013 Proceedings of the 26th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), pp. 1–4, Vancouver (2013)

23. Kousiouris, G., Menychtas, A., Kyriazis, D., Gogouvitis, S., Varvarigou, T.: Dynamic, behavioralbased estimation of resource provisioning based on high-level application terms in cloud platforms. Future Gener. Comput. Syst. 32, 27–40 (2014)
24. Manvi, S.S., Krishna Shyam, G.: Resource management for Infrastructure as a Service (IaaS) in cloud computing: a survey. J. Netw. Comput. Appl. 41, 424–440 (2014)
25. Lee, S., Meredith, J.S., Vetter, J.S., COMPASS: a framework for automated performance modeling and prediction. In: Proceedings of the 29th ACM on International Conference on Supercomputing, ICS'15, Newport Beach/Irvine, pp. 405–414 (2015)
26. Larsson, T., Svensson, M.: Resource Prediction for Cloud Computing. U.S. Patent 20160380908, 29 Dec 2016 (2016)
27. Jacobson, D.I., Altos, L., Joshi, N., Jose, S., Oberai, P., Carlos, S., Yuan, Y., Fremont, Tufs, P.S.: Pacifc grove, predictive auto scaling engine. U.S. Patent 14057898, 23 Apr 2015 (2015)
28. Kumar, A., Sangwan, S. R., Nayyar, A.: Multimedia social big data: mining. In: Multimedia Big Data Computing for IoT Applications. Springer, Singapore, pp. 289–321 (2020)
29. Kaur, A., Gupta, P., Singh, M., Nayyar, A.: Data placement in era of cloud computing: a survey, taxonomy and open research issues. Scalable Comput. Pract. Exp. 20(2), 377–398 (2019)
30. Singh, P., Gupta, P., Jyoti, K., Nayyar, A.: Research on auto-scaling of web applications in cloud: survey, trends and future directions. Scalable Comput. Pract. Exp. 20(2), 399–432 (2019)
31. Calheiros, R.N., Masoumi, E., Ranjan, R., Buyya, R.: Workload prediction using ARIMA model and its impact on cloud applications' QoS. IEEE Trans. Cloud Comput. 3(4), 449–458 (2015)
32. Kumar, R., Kalra, M., Tanwar, S., Tyagi, S., Kumar, N.: Min-parent: an efective approach to enhance resource utilization in cloud environment. In: 2016 International Conference on Advances in Computing, Communication, & Automation (ICACCA). Springer, Dehradun, pp. 1–6 (2016)
33. Tyagi, S., Obaidat, M.S., Tanwar, S., Kumar, N., Lal, M.: Sensor cloud based measurement to management system for precise irrigation. In: GLOBECOM 2017—2017 IEEE Global Communications Conference, Singapore, pp. 1–6 (2017)
34. Zhao, Shenghui., Chen, Haibao., Zhao, Ruibin., Zhao, Yuyan., Chen, Guilin.: A big data processingoriented prediction method of cloud computing service request. J. Appl. Sci. Eng. 19(4), 497–504 (2016)
35. Prasad, V., Nair, A., Tanwar, S.: Resource Allocation in Cloud Computing, Instant Guide to Cloud Computing. BPB Publications, New Delhi, pp. 343–376 (2019)
36. Singh, S., Chana, I.: A survey on resource scheduling in cloud computing: issues and challenges. J Grid Comput. 14(2), 217–264 (2016)
37. Serrano, D., Bouchenak, S., Kouki, Y., Alvares de Oliveira Jr., F., Ledoux, T., Lejeune, J., Sopena, J., Arantes, L., Sens, P.: SLA guarantees for cloud services. Future Gener. Comput. Syst. 54, 233–246 (2016)
38. Aras, C.M., Miller, J.D., Scott, R.K.: System for allocation of network resources using an autoregressive integrated moving average method. U.S. Patent 5884037, 16 Mar 1999 (1999)
39. Wood, D.A., Zafer, M., Zerfos, P.: Fast and automated ARIMA model initialization. U.S. Patent 14163418, 24 Jan 2014 (2014)
40. Singh, S.P., Nayyar, A., Kumar, R., Sharma, A.: Fog computing: from architecture to edge computing and big data processing. J. Supercomput. 75(4), 2070–2105 (2019)
41. Nayyar, A.: Private virtual infrastructure (PVI) model for cloud computing. Int. J. Softw. Eng. Res. Pract. 1(1), 10–14 (2011)

42. Wang, W., Yanmin, L.: Analysis of the mean absolute error (MAE) and the root mean square error (RMSE) in assessing rounding model. IOP Conf. Ser. Mater. Sci. Eng. 324, 1–11 (2018)
43. In Jae Myung: Tutorial on maximum likelihood estimation. J. Math. Psychol. 47(1), 90–100 (2003)
44. Kalyvianaki, E., Charalambous, T., Hand, S.: Self-adaptive and self-confgured CPU resource provisioning for virtualized servers using Kalman flters. In: Proceedings of the 6th International Conference on Autonomic Computing (ICAC ‘09). ACM, New York, pp. 117–126 (2009)
45. Chen, J., Wang, Y.: A hybrid method for short-term host utilization prediction in cloud computing. J. Electr. Comput. Eng. 1–14 (2019)
46. https://dumps.wikimedia.org/other/pagecounts-raw. Accessed 22 Aug 2016
47. Kumari, A., Tanwar, S., Tyagi, S., Kumar, N., Parizi, R.M., Choo, K.R.: Fog data analytics: a taxonomy and process model. J. Netw. Comput. Appl. 128, 90–104 (2019)