2024 Third International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)

Development of a Predictive VM Provisioning in a Cloud Environment

1st Sai Likith P
Electronics and Communication Engineering
Nitte Meenakshi Institute of Technology
Bengaluru, India
psailikith12@gmail.com

2nd Santhosh P
Electronics and Communication Engineering
Nitte Meenakshi Institute of Technology
Bengaluru, India
santhosh12345ind@gmail.com

3rd Sudeeksha K
Electronics and Communication Engineering
Nitte Meenakshi Institute of Technology
Bengaluru, India
sudeeksha24113@gmail.com

4th Karunakara Rai B
Electronics and Communication Engineering
Nitte Meenakshi Institute of Technology
Bengaluru, India
karunakara.rai@nmit.ac.in

5th Rajani N
Electronics and Communication Engineering
Nitte Meenakshi Institute of Technology
Bengaluru, India
Rajani.n@nmit.ac.in

2024 Third International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE) | 979-8-3503-1860-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICDCECE60827.2024.10549492

***Abstract*—In contemporary virtualized data center environments, efficient resource allocation in virtualized data center environments is crucial for preventing server downtimes. This paper introduces a method to mitigate resource exhaustion on ESXi servers using predictive analytics. Leveraging historical workload data, the proposed approach forecasts potential resource saturation events by analyzing past usage patterns. A proactive provisioning mechanism is outlined to deploy virtual machines (VMs) onto new ESXi servers during predicted high-load intervals. This preemptive strategy strategically distributes workloads, averting server downtime and ensuring optimal resource utilization. Experimental results validate the effectiveness of the proposed method in improving the dependability and security of ESXi servers in virtualized environments and out performs better than other state-of-arts. The proposed method is implemented using ESXi host client and the result shows effective performance and improved stability of the overall system.**

Keywords—VM, cloud computing, VM provisioning, workload pattern, host resource prediction, VM allocation.

I. INTRODUCTION

In recent years, the widespread adoption of cloud computing has transfigured how businesses manage their computing resources, offering on-demand access to scalable infrastructure. However, as the utilization of virtual machines (VMs) proliferates, effectively managing and allocating resources within a cloud environment becomes increasingly challenging. To address this, innovative technologies like virtualization have emerged, enabling businesses to optimize resource usage and enhance performance. Datacenter virtualization, a cornerstone of modern cloud infrastructure, involves the creation, deployment, and management of virtualized data centers. By virtualizing physical servers and leveraging advanced computing technologies, organizations can efficiently manage storage, networking, and other critical infrastructure components. Researchers therefore face difficulties when co-hosting many types of virtual machines (VMs) on a small number of servers due to resource conflict between co-hosted applications, which causes servers to be overutilized, which in turn degrades application performance. Furthermore, a lot of cloud services, such as interactive apps, have regularly shifting workload requirements, which lead to dynamic resource demand. If dynamic server consolidation is employed, this might cause SLA violations and performance degradation. This approach leverages historical data to forecast future resource requirements and allocate resources pre-emptively. By analysing past consumption trends, predictive VM provisioning assures that sufficient resources are available to meet anticipated demand, optimizing resource utilization and application performance. In sectors like banking, where server availability is paramount during peak periods of activity, such as high-volume transactions or system updates, cloud environments may experience sudden spikes in resource demand and there is high chances of server-failures. To address the aforementioned concerns, the hypervisor identifies suitable virtual machines or sets of virtual machines and relocates them from heavily utilized servers to underutilized ones, aiming to enhance overall performance. Predictive VM provisioning anticipates these spikes, checks and assures that adequate resources are allocated in advance, preventing service disruptions and maintaining uninterrupted operations.

To achieve this, effective VM allocation methods are essential. By employing statistical data based on observed job patterns and system metrics, organizations can anticipate future resource requirements and allocate VMs accordingly. This proactive approach minimizes the risk of Service Level Agreement (SLA) violations and optimizes resource utilization efficiency. Additionally, VM selection techniques play a vital role in resource management within cloud environments. By strategically choosing VMs for migration based on workload distribution and resource availability, organizations can rebalance the load across the system's active nodes without causing downtime or service interruptions. In [12] presents a survey of predictive VM provisioning strategies, encompassing existing prediction algorithms that aid and support in selecting virtual machines (VMs) to be supplied to other servers. In this study, we give a thorough assessment on cutting-edge predictive virtual machine provisioning strategies and overcome the shortcomings of previous surveys.

The primary advancements outlined in this paper are as follows:

1. A thorough analysis of the literature on cutting-edge predictive virtual machine provisioning strategies, outlining their advantages, disadvantages, and important research gaps.
2. Discussions on different security risks, server failures and current mitigation techniques.
3. Identification of specific gaps and research challenges to reduce the downtime of servers.

The rest of this paper is organized as follows. Section II summarizes related works. Section III presents a proposed method, which provides a solution to prevent down time of servers. Section IV includes software requirements, host workload analysis, comparative analysis, key findings under results and discussion. We present our performance evaluation in Section IV, while the conclusion and future work are presented in Section V.

II. Related Work

In their seminal work, the authors of paper [1] addresses the challenge of optimizing the allocation of virtual machines (VMs) in cloud servers to enhance energy efficiency in cloud radio access networks. Employing Monte Carlo-based evolutionary algorithms such as particle swarm optimization, and genetic algorithms, the study aims to find the optimal number of VMs for energy-efficient operation. The model shows the impact of adding a greater number of VMs on server performance, particularly in processing resource blocks (RBs) per VM. Advantages of the model include its support for energy efficiency and improvement in quality-of-service metrics such as minimum user equipment data rate, allocated Resource blocks, and latency due to virtualization. However, shortcomings are noted in VM placement and power allocations. Additionally, the paper proposes a power model to assess power usage in virtualized server components, aiming to simplify evaluation processes. As elucidated by the authors in paper [2] they explored VM consolidation as a means to enhance resource utilization, utilizing a regression model to forecast future CPU and memory usage. Through analysis of real workload traces from Google cluster and Planet Lab, we implement the UP-VMC approach, aiming to minimize unnecessary VM migrations. The results demonstrate that VM consolidation can significantly reduce energy consumption, up to 71.6% compared to alternative methods, by consolidating VMs onto the most heavily loaded physical machines. However, challenges remain in terms of scalability and optimizing VM placement, particularly regarding network resource utilization and traffic management. Additionally, the authors introduce the UP-VMC strategy to optimize the energy efficiency of Cloud Radio Access Networks (C-RAN) by determining the optimal number of virtual machines.

In paper [3], a Multi objective genetic algorithm (GA) is employed to predictively estimate how resources are used and energy is consumed in cloud data centers in real time, considering CPU and memory utilization of both virtual machines (VMs) and physical machines (PMs). The study results in the development of an algorithm for placing virtual machines designed to enhance overall resource utilization and decrease energy consumption within the data center by leveraging prediction outcomes from genetic algorithms. The approach enhances PM utilization while decreasing energy consumption by minimizing the number of active PMs. However, limitations exist in the accuracy of the approach in realistic cloud environments, as it has not been tested with real-world data center traces such as those from Google. Additionally, the authors propose a novel GA-based prediction strategy to enhance forecast precision in cloud data centers and suggest a VM placement strategy based on GA predictions to optimize resource utilization and energy consumption. Expounding on the findings presented in paper [4] addresses the challenge of forecasting future cloud resource utilization by employing the Support Vector Regression Technique (SVRT) with a Radial Basis Function (RBF) kernel function. This method aims to accurately predict multi-attribute host resource utilization, particularly suited for nonlinear workload patterns. The Sequential Minimal Optimization Algorithm (SMOA) is applied for training and regression estimation to enhance prediction accuracy. By utilizing SVRT with the RBF kernel function, the research suggests a method less susceptible to fluctuations in resource utilization compared to other forecasting techniques. Through experimentation with eight datasets, the research investigates variations in workload demand and resource utilization within a cloud setting, emphasizing the opportunity to minimize resource waste and enhance resource efficiency in cloud data centers.

Embarking on their analysis in paper [5] the author develops a NICBLE-based system prototype for Xen virtualization to assess the impact of hypothetical changes in VM setups on central processor quantities. NICBLE predicts application execution based on simulated calculations. The Priority-Aware VM Allocation (PAVA) technique is proposed for VM allocation, leveraging network topology information to assign critical applications to closely connected hosts. The model further explores relationships between VMs based on ARIMA-predicted resource requirements, introducing an affinity model to evaluate resource utilization volatility when VMs are placed on the same host. The resultant algorithm for virtual machine placement, rooted in predicting affinity, consolidates VMs with strong connections onto single physical machines, maximizing resource usage while staying within the limits of physical machine capacities. Extensive simulation experiments validate the algorithm's effectiveness in reducing energy consumption, VM migrations, and SLA violations based on Planet lab and Google workload traces. However, further adjustments are needed to improve the accuracy of the prediction model. The authors in [6] have released the ground-breaking Forecast Empathy Based Virtual Machine Positioning algorithm. By using this technique, a single physical machine is created from the high-affinity virtual machines. Estimating a request within a range is pointless because the projection will continuously stray from the actual requests and serve as a guide.

Within the context of paper [7], the researchers utilize the Support Vector Relapse Strategy (SVRT), a managed factual learning technique, to foresee how the multi-property have asset will be utilized from here on out. Utilizing cloud resources to manage a non-linear workload is a perfect application for this strategy. We pick Spiral Premise Capability as the bit capability of SVRT and utilize Consecutive Negligible Streamlining Calculation (SMOA) for the preparation and relapse assessment of the expectation strategy to build the expectation precision of SVRT. Addressing the subject matter in paper [8] they proposed the use of a Markov prediction model to forecast the future load state of hosts in cloud environments. We introduce a host load detection algorithm to identify over or under-utilized hosts, aiming to prevent immediate VM migration. A VM algorithm for placement is then employed to select candidate hosts for migrated VMs, evaluated using cloud Sim simulation. Our approach relies on dynamic utilization thresholds to enhance the VM placement process and predict results to avoid host overload shortly after migration. Additionally, the Median Absolute Deviation Markov Chain Hot Detection technique (MadMCHD) to improve host detection performance during live migration by determining future overutilized and

underutilized states is used. By incorporating the RobustSLR prediction algorithm into the PABFD algorithm, authors altered both the existing VM placement method and the new VM algorithm for placement [9].

In the situation of a 140% system load, the suggested method can handle up to 12% more user requests and generate up to 8% more system rewards [10]. A comprehensive analysis by [11] introduces a predictive auto-scaling mechanism for elastic cloud services, integrating time series forecasting using machine learning methods with queuing theory to enhance service response times and reduce unnecessary resource allocation. This approach utilizes SVM regression to anticipate the workload of web servers by analyzing past data. Results indicate that SVM-based forecasting models perform better than basic models in terms of reducing over-provisioned resources. However, some SVM-based models exhibit poorer performance regarding SLA violations and unserved requests. Advantages include better prediction accuracy compared to simpler methods, as demonstrated by MAE and RMSE error measurements. Nonetheless, challenges remain in adapting predicting and performance models to the diverse components and functionalities of other cluster architectures such as MapReduce, HDFS, and Spark components. According to the research presented in [12] the authors employed a hybrid ARIMA–ANN model to predict future CPU and memory utilization in cloud environments, leveraging both linear and nonlinear components in the data. Analysis and experiments are conducted using workload traces from Google trace and Bit Brain compute clusters, focusing on CPU and memory usage. While the ARIMA model detects linear patterns, the ANN is effective in predicting nonlinear patterns by leveraging residuals from the ARIMA model. Advantages include improved prediction accuracy for nonlinear patterns with ANN, although reliance on training data may affect performance. To address this, the proposed hybrid model combines the strengths of both ARIMA and ANN. However, challenges remain in dynamically adjusting the sliding window size and assigning weights to recent historical data points to further enhance prediction accuracy.

The endeavor to optimize server allocation in the face of potential failures has prompted the development of advanced algorithms aimed at minimizing delays and maximizing system efficiency. In this context of [13], recent research has proposed polynomial-time approximation algorithms leveraging Particle Swarm Optimization (PSO) to tackle the distributed server assignment problem in the event of single server failures. The NP-completeness of the problem has been established, underscoring the complexity of the task at hand. The proposed algorithms demonstrate promising approximation performances, offering significant speed enhancements compared to traditional Integer Linear Programming (ILP) approaches. Numerical analyses indicate substantial efficiency gains, with the proposed algorithms outperforming ILP by factors ranging from 3.0×10^3 to 1.9×10^7 while slightly increasing the largest maximum delay by a factor of 1.033 on average. These findings shed light on novel strategies for efficiently addressing server allocation challenges in distributed systems, paving the way for enhanced reliability and performance in cloud computing environments. The scholarly contribution of [14] the authors delve into the intricacies of ensuring high redundancy and almost zero downtime for enterprise applications within cloud environments. They highlight the significant challenge of establishing surplus and robust architectures, both at the application and database layers. Emphasizing the importance of automated failover mechanisms, particularly for applications expected to operate around the clock, the authors propose a cloud-native template architecture for enhancing availability. Through their evaluation of automatic database failover techniques, they aim to minimize disruptions during planned maintenance activities and outages. While acknowledging the variability in achieving uninterrupted service based on factors like application size and data volume, the authors provide a foundational framework for building resilient applications. They stress the need for customization of the recommended architecture and failover strategies to align with each application's unique requirements, considering factors such as cost and specific business objectives.

An in-depth exploration of cloud computing, beginning with a comprehensive background to setup a foundational grasp of the subject. It delves into fault-tolerance components and system-level metrics, emphasizing their significance and applications within cloud computing environments. The authors meticulously examine both proactive and reactive approaches to fault-tolerance in cloud computing, showcasing the state-of-the-art techniques and frameworks. By organizing and discussing current research endeavors in fault-tolerance architectures, the paper offers valuable insights into the evolving landscape of cloud computing resilience. It concludes by outlining key future research directions, underscoring the importance of continued development in fault-tolerance strategies specific to cloud computing [15]. The authors in [16], presents a comprehensive data-centric analysis correlating DRAM errors with server outages, aiming to predict server outages based on DRAM error patterns. Leveraging an extensive eight-month dataset from Alibaba's production data centers, comprising over three million memory modules, we identify that correctable DRAM errors often precede server failures, highlighting the importance of regular and frequent server failure prediction intervals for accurate forecasting. Our investigation also explores various factors influencing instances of server malfunctions, encompassing component breakdowns within the memory subsystem, variations in DRAM setups, and categories of fixable DRAM inconsistencies. Notably, we extend prior work by considering multiple types of server malfunctions in the prediction of server failures, achieving significant reductions in server downtime. Our findings reveal that UE-driven failures pose challenges in prediction due to the limited occurrence of correctable errors before these failures, whereas CE-driven and miscellaneous failures exhibit higher predictability. By employing all feature groups, we enhance prediction accuracy, with tree-based prediction models demonstrating superior performance, underscoring the importance of short prediction intervals for timely failure prediction.

Delving into the intricate the correlation between input/output (I/O) workload characteristics and disk reliability, aiming to uncover factors influencing disk lifespan and identify detrimental I/O workloads. By proposing an innovative measure, AISR (Average I/O Service Rate), we shed light on "dangerous" I/O workloads posing significant risks to disk health. Our research represents a pioneering effort in comprehensively analyzing how input/output (I/O) workload influences disk dependability, providing valuable perspectives to improve I/O scheduling strategies within data centers. While our work marks an initial exploration in this

domain, we anticipate that our findings will stimulate further research in the community, prompting a reevaluation of disk I/O workload assignments. Future endeavors will focus on incorporating additional workload metrics to extract actionable insights and implement them effectively in data center environments [17]. As documented in [18] the research introduces a strategy for determining the sequence of migrations, utilizing network traffic data between virtual machines to reduce downtime by 50% during cloud-to-cloud migration. Employing Prim's algorithm, they identify the Minimum Spanning Tree (MST) for a weighted undirected graph, facilitating efficient migration. The controller nodes, equipped with Intel Xeon W35653.25GHz CPU, 48GB RAM, and 1TB SSD, contribute to achieving shorter service downtime during migration. The strategy accounts for migrating multiple VMs, addressing communication dependencies among them. While the approach yields shorter service downtime, it may sometimes necessitate higher security measures, potentially leading to increased service downtime.

Through their meticulous study in [19], unveils an analysis among the services that are online and batch jobs, which are collocated within a production cluster in Alibaba Cloud. By clustering servers based on CPU and memory utilization correlations, it identifies opportunities for job co-allocation and resource estimation. Through examination of mean time between failures (MTBF) and completion times, it sheds light on failure distribution and workload assignment disparities. The insights gleaned from this analysis can empower data center operators to optimize resource utilization and enhance failure recovery mechanisms, ultimately fostering a deeper understanding of workload characteristics and operational efficiencies within cloud environments. In accordance with [20] the author evaluates the effectiveness of various disk-failure prediction methods using metrics such as timeliness and convergence. By comparing classification tree (CT), a neural network with recurrent connections and a regression tree model boosted using gradient descent models, the research highlights the nuanced performance differences across prediction experiments. While RNN exhibits superior accuracy, CT and GBRT models demonstrate advantages in resource-dependent migration rates. The findings underscore the importance of considering prediction accuracy in conjunction with practical outcomes, prompting the introduction of an enhanced GBRT model (GBRT+). Moving forward, the exploration of urgency-weighted evaluations and adjustments to machine learning processes offer promising avenues for refining disk-failure prediction models.

III. Proposed Method

The host operating system (OS) as shown in Figure 1 H2H (Host to Host) VM Provisioning is the foundation of the Type 2 hypervisor architecture. This is the main operating system that is directly installed on the computer's Physical hardware. It might be Linux, macOS, or Windows.

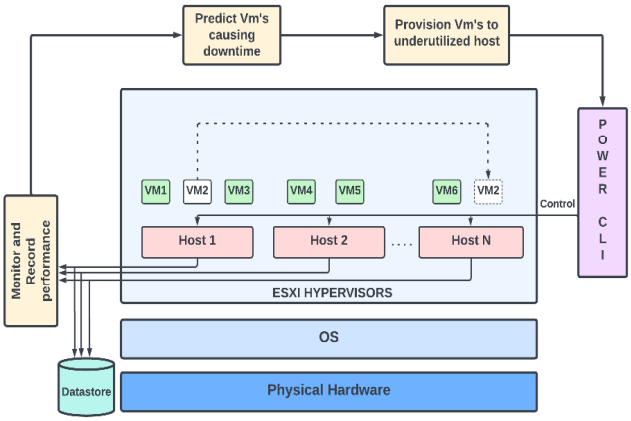

Fig. 1. H2H VM Provisioning

The Type 2 hypervisor operates as an application on top of the host operating system. Multiple guest operating systems can operate concurrently on the same physical hardware by the virtualization layer created by hypervisor software like VMware Workstation Player. One or more guest operating systems may be installed as virtual machines inside the Type 2 hypervisor environment. Both the host OS and these guest OS instances run independently of one another. Different operating systems, including different Windows versions, Linux distributions, and (in certain situations) macOS, can be installed by users.

In order to allocate parts of the hardware resources such as CPU, memory, storage, and network interfaces to each virtual machine as required, the hypervisor manages these resources. As a result, numerous virtual machines (VMs) can share the physical resources without interfering with one another. Datastores are used by guest operating systems to store virtual hard drives (VHDs) or virtual machine disk images. They are commonly represented as disk image files. These files may be kept on local or network storage or on any other storage device that the host operating system may access. Users communicate with the Type 2 hypervisor via a management interface supplied by the hypervisor software. This interface allows users to set up, configure, start, stop, and delete virtual machines, as well as manage their settings and resources.

The proposed system is based on a real-time feed as inputs and follows the operational process shown in Figure 2.

- ESXi Hypervisor Setup
- Datastore Creation
- VM Creation and Provisioning
- Template Creation
- Installation of Power CLI
- Data Collection and Prediction
- VM Migration

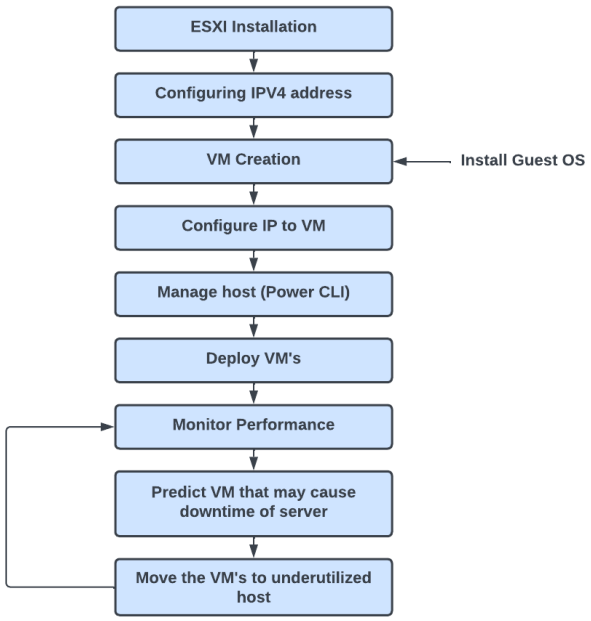

Fig. 2. Proposed Solution for VM Provisioning

ESXi hypervisors are installed on physical hardware. Each hypervisor is configured with IP addresses (IPv4), DNS server addresses, and default gateways. Datastores are established to allocate memory to virtual machines (VMs) and houses the disk files of VMs. This allows for efficient management of resources. Let's consider 2 hosts (ESXi server version 8.0.2) host 1 as remoteServer.localdomain configured with static ipv4 address "192.168.204.2" and host2 as host2.localdomain with static address "192.168.204.8" each with 4GB of RAM and 4 CPU cores as shown in Table 1 host information.

TABLE I. HOST INFORMATION

Host Name	Num CPU	CPU Usage (Mhz)	CPU Total (Mhz)	Memory Usage (GB)	Memory Total (GB)
192.168.204.2	4	462	9980	1.59	3.999
192.168.204.8	4	156	9980	1.578	3.999

When a request for a new VM arises, resources are allocated based on customer demands. VMs are created and configured with unique IP addresses. ISO image files are used to load guest operating systems onto these VMs. Templates are created to expedite the provisioning process. These templates contain pre-configured settings and configurations, enabling quick deployment of new VMs within minutes. Two VMs vm1 and WIN-2019-ser-2 are created under host1, VM named h2_WIN-2019-ser-2 alone is created in host2. The number of CPU cores, RAM and datastore of these VMs are shown in Table 2 VM information.

TABLE II. VM INFORMATION

VM Name	VM id	EXSi Host	Data Store	Num CPU	Memory (GB)
vm1	10	192.168.204.2	Storage_ datastore _1	2	1
WIN-2019-ser-vm-2	9	192.168.204.2	Storage_ datastore _1	2	4
h2_WIN-2019-ser-vm-2	9	192.168.204.8	Storage_ datastore _2	2	4

Power CLI is used to automate and manage the Vsphere, Vcenter and Vmware host client using power shell by logging in with credentials of the server. The usage the of host and VMs are recorded with the use of CLI script for a specific time interval for analyzing high resource usage (Memory, CPU, network, disk Usage) of the VMs in Over-Provisioned Host. Collecting and preprocessing data from diverse sources like CPU usage, memory usage, and disk usage enables the prediction of future resource usage for VMs and the detection of those exceeding 80% resource consumption.

$$obj(y, \hat{y}) = ½ \Sigma (yi - \hat{y}i)^2 + \lambda \Sigma |w\_j|^\gamma \quad (1)$$

Where

- obj objective function in SG model minimizes both the training error and a regularization term to prevent overfitting.
- yi is the actual value of the i-th data point.
- $\hat{y}i$ is the predicted value of the i-th data point by the model.
- λ is the regularization parameter controlling the strength of the penalty term.
- w_j is the weight of the j-th feature in the model
- γ is a hyperparameter that determines the type of regularization.

Let T(x_i) represent the prediction of the i-th data point by a single tree in the ensemble. The ensemble prediction can be formulated as:

$$\hat{y}i = \Sigma f\_m(x\_i) \quad (2)$$

where:

- m iterates over all trees in the ensemble.
- $f\_m(x\_i)$ is the prediction of the m-th tree for the i-th data point.

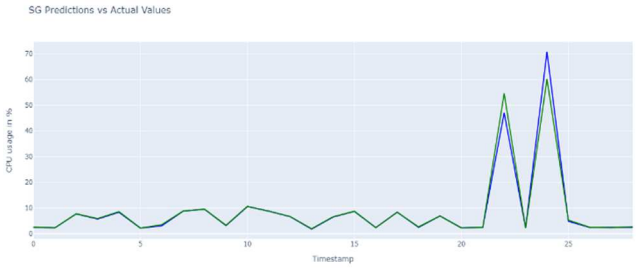

Fig. 3. CPU Usage Actual vs Predicted plot

By analyzing the Figure 3, its observed the accuracy of model is better fit for the dataset. The predicted resource usage of these VM's are collected, analyzed and detected the VMs causing downtime of the Server as shown in Figure 4 sum of resource Usage of CPU, Memory and Disk Usage.

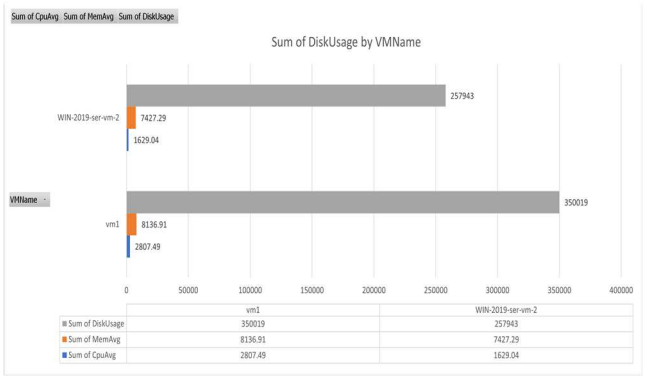

Fig. 4. Sum of Resource Usage

When the resource consumption of a host exceeds 80%, it evaluates the total resource usage (Rs_Usage) across all virtual machines within a specified time interval. This total usage (Rs_Usage) is calculated by summing the resource usage (Rs_Usage) of each VM at various time points (t_i) within the interval.

$$Total\ Rs\_Usage(X) = \sum Rs\_Usage\ (X, ti) \quad (3)$$

The VM with the highest total resource usage (Rs_Usage) is identified as Xm, indicating that it's consuming the most resources.

$$\text{Total Rs\_Usage (Xm)} > \text{Total Rs\_Usage(X1), Total}$$

$$Rs\_Usage(X2), \ldots\ldots, Total\ Rs\_Usage\ (Xn) \quad (4)$$

Where X1, X2, X3, …., Xn are n different virtual machines.

$m \neq 1,2\ldots.\ N$

In this case, when the host's consumption is above 80%, we decide to move or provision the Xm VM to alleviate the strain on the host and maintain optimal performance.

For host (Rs_consumption) <= 80%

$$VM\ to\ be\ moved\ or\ provisioned = Xm \quad (5)$$

Conversely, when the host's resource consumption is below 80%, no action is taken as the resource utilization is within acceptable limits. Identified VMs that are predicted to cause downtime are removed from host and provisioned to underutilized ESXi host. This proactive approach helps optimize resource utilization and mitigate potential disruptions to services.

IV. Results

A. Software Requirements

The system requirements include a minimum of 12 GB RAM and 256 GB storage space. Essential software components comprise VMware Workstation Player for virtualization, a hypervisor like VMware ESXi for resource management, ISO image files for OS installation, VMware VCenter Appliance for centralized management, and Power CLI for automation.

B. Host Workload Analysis

The graph in Figure 5 represents that the CPU utilization consistently exceeds the 80% threshold. CPU utilization refers to the percentage of the CPU's processing capacity that is being used at any given time. In this context, surpassing the 80% threshold indicates that host1's CPU is being heavily utilized, potentially reaching its maximum capacity frequently. Such high CPU utilization poses a significant risk to the server's stability and performance. When the CPU is consistently operating at or near its maximum capacity, it leaves little room for handling additional processing tasks or spikes in workload. As a result, there is an increased likelihood of server downtime or performance degradation. To mitigate this risk and ensure the continued smooth operation of the server, it's advisable to take proactive measures such as virtual machine migration. By migrating some of the virtual machines from host1 to other hosts with lower resource utilization, the burden on host1's CPU can be alleviated, reducing the risk of downtime and ensuring optimal performance for all virtual machines hosted on the server.

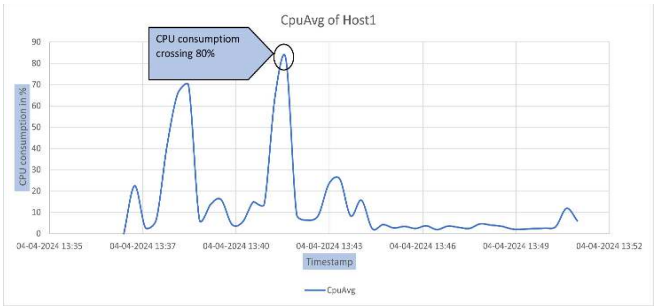

Fig. 5. Host 1 consumption before VM migration

Migrating vm1 to host2 involves relocating its workload and associated resources from host1 to host2. By doing so, the CPU usage on host1 will decrease, mitigating the risk of server downtime and ensuring optimal resource allocation across the infrastructure. Executing this migration is essential for maintaining the stability and performance of the server environment. It allows for better distribution of workload and prevents any single virtual machine from monopolizing resources as shown in Figure 6, thereby promoting overall system efficiency and reliability.

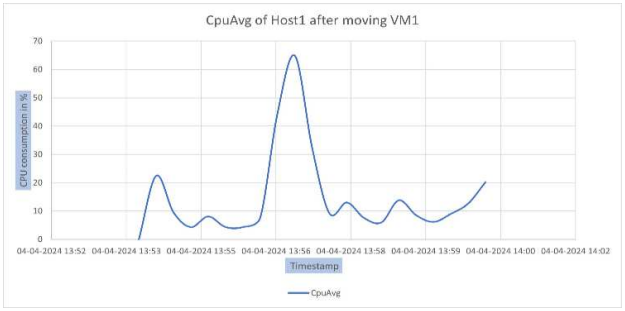

Fig. 6. CPU average of Host 1 after moving vm1

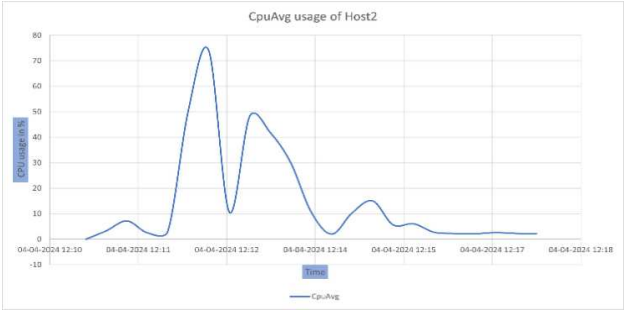

Fig. 7. CPU Average of Host 2 after moving vm1

As depicted in Figure 6, the average CPU utilization on host 1 post-migration has indeed decreased to below 80%. Subsequently, Figure 7 illustrates the CPU usage post-migration of VM1 to host 2, revealing that despite the migration, CPU utilization remains below the 80% threshold. Consequently, it is evident that migrating vm1 is the optimal solution to effectively manage CPU usage within acceptable parameters.

C. Discussion

The performance of various forecasting models, including ARIMA, SARIMA, SVM, Prophet, TensorFlow, and SG, was evaluated based on metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) shown in Table 3.

TABLE III. ACCURACY OF MODELS

Models	MAE	MSE	RMSE
Arima	0.78196	6.81183	2.44604
Sarima	14.2945	247.0878	15.71902
SVM	1.631639	5.892534626	2.724808
Prophet	6.177789	6.07989	7.1269
TensorFlow	8.566368	10.21722	11.7898
SG	0.7133	5.87738	2.424332

SG model, with a MAE of 0.7133, MSE of 5.87738, and RMSE of 2.424332, showed competitive performance similar to ARIMA. Despite its effectiveness, the proposed system has certain limitations. One limitation is the reliance on historical data for predictive modeling, which may not always capture sudden or unforeseen changes in workload patterns. Moreover, the accuracy of the predictions may be influenced by factors such as data quality, modeling assumptions, and the complexity of the underlying system architecture. Furthermore, the automated migration or provisioning of VMs may introduce additional overhead and complexity, requiring careful monitoring and management to ensure smooth operation.

The proposed system utilizes real-time data feeds to predict resource usage for virtual machines (VMs) hosted on ESXi hypervisors. By collecting and preprocessing data on CPU, memory, and disk usage, the system predicts future resource needs and detects VMs exceeding 80% resource consumption. The system then automatically migrates or provisions VMs to maintain optimal performance and prevent downtime. Through the analysis, it's evident that the predictive modeling approach based on historical resource usage data can effectively forecast future resource needs. By identifying VMs with high resource consumption, the system can proactively manage host resource allocation, thus ensuring efficient resource utilization and minimizing service disruptions.

The proposed system offers a proactive and automated approach to resource management in virtualized environments. By leveraging predictive modeling and automation technologies, the system effectively addresses the challenges of resource allocation and optimization, thereby improving the reliability, scalability, and efficiency of ESXi hypervisor deployments. However, ongoing monitoring, evaluation, and refinement are essential to ensure the system's continued effectiveness and adaptability to evolving workload demands and system requirements.

V. CONCLUSION

The experiments conducted in this study aimed to assess the effectiveness of a predictive resource provisioning approach in preventing server downtimes due to resource overutilization in ESXi servers. The results demonstrate that leveraging historical workload data and deploying a predictive analytics model for forecasting critical periods, coupled with a dynamic provisioning mechanism, significantly enhances the stability and resilience of virtualized environments. The proactive nature of the proposed approach not only prevents server downtimes but also contributes to increased overall system reliability and performance. Future research could explore additional refinements to the predictive analytics model, considering evolving workload patterns and incorporating real-time adjustments to the provisioning strategy. Overall, this aims to enhance the stability, performance, and reliability of virtualized environments, thereby improving the overall service delivery and user experience. With proper implementation and ongoing monitoring, this solution can help organizations effectively manage their virtual infrastructure while meeting the demands of their customers.

REFERENCES

[1] R. S. Alhumaima, R. K. Ahmed and H. S. Al-Raweshidy, "Maximizing the Energy Efficiency of Virtualized C-RAN via Optimizing the Number of Virtual Machines," in IEEE Transactions on Green Communications and Networking, vol. 2, no. 4, pp. 992-1001, Dec. 2018, doi: 10.1109/TGCN.2018.2859407.

[2] F. Farahnakian, T. Pahikkala, P. Liljeberg, J. Plosila, N. T. Hieu and H. Tenhunen, "Energy-Aware VM Consolidation in Cloud Data Centers Using Utilization Prediction Model," in IEEE Transactions on Cloud Computing, vol. 7, no. 2, pp. 524-536, 1 April-June 2019, doi: 10.1109/TCC.2016.2617374.

[3] F. -H. Tseng, X. Wang, L. -D. Chou, H. -C. Chao and V. C. M. Leung, "Dynamic Resource Prediction and Allocation for Cloud Data Center Using the Multiobjective Genetic Algorithm," in IEEE Systems Journal, vol. 12, no. 2, pp. 1688-1699, June 2018, doi: 10.1109/JSYST.2017.2722476.

[4] L. Abdullah, H. Li, S. Al-Jamali, A. Al-Badwi and C. Ruan, "Predicting Multi-Attribute Host Resource Utilization Using Support Vector Regression Technique," in IEEE Access, vol. 8, pp. 66048-66067, 2020, doi: 10.1109/ACCESS.2020.2984056.

[5] X. Fu and C. Zhou, "Predicted Affinity Based Virtual Machine Placement in Cloud Computing Environments," in IEEE Transactions on Cloud Computing, vol. 8, no. 1, pp. 246-255, 1 Jan.-March 2020, doi: 10.1109/TCC.2017.2737624.

[6] B. Xia, T. Li, Q. Zhou, Q. Li and H. Zhang, "An Effective Classification-Based Framework for Predicting Cloud Capacity Demand in Cloud Services," in IEEE Transactions on Services Computing, vol. 14, no. 4, pp. 944-956, 1 July-Aug. 2021, doi: 10.1109/TSC.2018.2804916.

[7] H. -W. Li, Y. -S. Wu, Y. -Y. Chen, C. -M. Wang and Y. -N. Huang, "Application Execution Time Prediction for Effective CPU Provisioning in Virtualization Environment," in IEEE Transactions on Parallel and Distributed Systems, vol. 28, no. 11, pp. 3074-3088, 1 Nov. 2017, doi: 10.1109/TPDS.2017.2707543

[8] S. B. Melhem, A. Agarwal, N. Goel and M. Zaman, "Markov Prediction Model for Host Load Detection and VM Placement in Live Migration," in IEEE Access, vol. 6, pp. 7190-7205, 2018, doi: 10.1109/ACCESS.2017.2785280.

[9] L. Li, J. Dong, D. Zuo and J. Wu, "SLA-Aware and Energy-Efficient VM Consolidation in Cloud Data Centers Using Robust Linear Regression Prediction Model," in IEEE Access, vol. 7, pp. 9490-9500, 2019, doi: 10.1109/ACCESS.2019.2891567.

[10] Singh, D., Vishnu, C. and Mohan, C.K., 2020, September. Real-time detection of motorcyclist without helmet using cascade of cnns on edge-device. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)* (pp. 1-8). IEEE.

[11] Moreno-Vozmediano, R., Montero, R.S., Huedo, E. et al, "Efficient resource provisioning for elastic Cloud services based on machine learning techniques," J CloudComp 8, 5 (2019).

[12] Devi, K.L., Valli, S, "Time series-based workload prediction using the statistical hybrid model for the cloud environment," Computing 105, 353–374 (2023).

[13] Souhei Yanase, Graduate Student Member, Fujun He and Eiji Oki, "Approximation Algorithms to Distributed Server Allocation With Preventive Start-Time Optimization Against Server Failure," in IEEE Networking Letters, Vol. 3, No. 4, December 2021

[14] Antra Malhotra, AMR Elsayed, Randolph Torres and Srinivas Venkatraman, "Evaluate Solutions for Achieving High Availability or Near Zero Downtime for Cloud Native Enterprise Applications," in IEEE Access vol. 11, doi: 10.1109/ACCESS.2023.3303430.

[15] A. U. Rehman Rui L. Aguiar, and JOÃO Paulo Barracca, "Fault-Tolerance in the Scope of Cloud Computing," in IEEE Access vol. 10, 2022, doi: 10.1109/ACCESS.2022.3182211.

[16] Zhinan Cheng, Shujie Han, Patrick P. C. Lee, Xin Li, Jiongzhou Liu and Zhan Li, "An In-Depth Correlative Study Between DRAM Errors and Server Failures in Production Data Centers," in 2022 41st International Symposium on Reliable Distributed Systems (SRDS) IEEE, DOI: 10.1109/SRDS55811.2022.00032

[17] Song Wu, Yusheng Yi, Jiang Xiao, Hai Jin, and Mao Ye, "A Large-Scale Study of I/O Workload's Impact on Disk Failure," in IEEE Access Vol. 6, 2018, doi: 10.1109/ACCESS.2018.2866522.

[18] Jargalsaikhan Narantuya, Hannie Zang, and Hyuk Lim, "Service-Aware Cloud-to-Cloud Migration of Multiple Virtual Machines," in IEEE Access Vol. 6, 2018, doi: 10.1109/ACCESS.2018.2882651.

[19] Congfeng Jiang, Guangjie Han, Jiangbin Lin, Gangyong Jia, Weisong Shi, and Jian Wani, "Characteristics of Co-Allocated Online Services and Batch Jobs in Internet Data Centers: A Case Study from Alibaba Cloud," in IEEE Access Vol. 7, 2019, doi: 10.1109/ACCESS.2019.2897898.

[20] Jing Li, Rebecca J. Stones, Gang Wang, Zhongwei Li, Xiaoguang Liu, and Jianli Ding, "New Metrics for Disk Failure Prediction That Go Beyond Prediction Accuracy," in IEEE Access Vol. 6, 2018, doi: 10.1109/ACCESS.2018.2884004.