2023 4th IEEE Global Conference for Advancement in Technology (GCAT)
Bangalore, India. Oct 6-8, 2023

Optimizing Resource Allocation using Proactive Predictive Analytics and ML-Driven Dynamic VM Placement

Utpal Chandra De
School of Computer Applications
KIIT Deemed to be University
Bhubaneswar, India
deutpal@gmail.com

Rabinarayan Satapathy
Faculty of Emerging Technologies
Sri Sri University
Cuttack, India
rabinarayan.satpathy@gmail.com

Sudhansu Shekhar Patra
School of Computer Applications
KIIT Deemed to be University
Bhubaneswar, India
sudhanshupatra@gmail.com

2023 4th IEEE Global Conference for Advancement in Technology (GCAT) | 979-8-3503-0525-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/GCAT59970.2023.10353234

***Abstract*—Whenever a user needs to work or deliver beyond the capabilities of the physical machine being used, virtual machines come into the picture. This may be in the form of a system and services being provided by clients to be able to work on their platform on subscribed tools and services or to acquire enhanced computational power in sessions using GPUs and whatnot. However, the area of study is the allocation and placement of the Virtual machines based on dynamic requirements so that the resource cost and energy required are minimized. This takes the form of an optimization task, based on various parameters associated with the system. In this paper, we have used a technique where we leverage predictive analytics to predict the demand for resources in the future, followed by which we allocate Virtual Machines on demand using Machine Learning to optimize the computational cost of VM-based systems.**

Keywords—Cloud Computing, Virtual Machine Allocation (VMA), User Monitoring System, Optimization, Machine Learning

I. INTRODUCTION

Physical computer systems have a limitation with respect to computation power and memory availability. To address this issue, we use virtual machines using Virtual Desktop Infrastructures to acquire more resources and perform our desired tasks. These tasks may be client-specific requirements that are beyond the ability of the physical system being used. There might also be a requirement for third-party services, databases, or tools to which the clients subscribe and need users to work with.

Now based on the requirements, virtual machines need to be allocated and used. Oftentimes, the machines are allocated but aren't used during reduced loads. For example, if 10 virtual machines are allocated but only 5 of them are being used, then the rest 5 of them are lying idle but the user needs to pay for it. In addition to allocation expenses, this also leads to increased and unnecessary energy consumption.

The placement and allocation of virtual machines play a significant role as mentioned earlier. In [1] the authors have used a multi-objective PSO algorithm to determine the placement of virtual machines in data center resources. In [2], the authors have proposed a new approach for the placement of virtual machines in a multi-data center cloud environment using a context-aware multi-objective genetic algorithm called aware genetic algorithm first fit (AGAFF). A model for initial virtual machine fault-tolerant placement for cloud data centers based on star topology has been built in [3] on the basis of failure rate, power consumption rate, fault tolerance rate, etc. A redundant VM placement optimization approach has been proposed by the authors in [4] using 3 algorithms – The 1st algorithm selects the appropriate set of servers to host the VM based on network topology. The 2nd algo finds the optimal strategy to place the primary and backup virtual machines. The 3rd one is a heuristic which addresses the task-to-VM reassignment optimization.

The authors in [5] have proposed a model to improve the decision-making process in virtual machine placement based on placement time, power consumption & resource wastage. The proposed fitness function was implemented using three techniques – PSOLF (Particle Swarm optimization with Levy Flight), FPO (Flower Pollination Optimization) & a hybrid algorithm of them called HPSOLF-FPO. A population-based meta-heuristic – TLBO (teaching-learning-based optimization) technique has been used by authors in [6] for cost optimization in VM allocation. Authors in [7] have performed a survey of VM placement approaches in cloud computing environments based on energy & power algorithms, network methodologies, multi-objective optimization, cost optimization, etc. In [8], the authors have performed a comprehensive survey of several challenges being faced while managing virtualized resources which include existing proposals, and problem formulation, followed by advantages and shortcomings of reviewed works.

In [9], the authors have provided a detailed review of preference representations, explained their existing usage & explained the adopted solving approaches, followed by which the authors have discussed key challenges & identified possible research opportunities in the context of VM placement. A Multi-Objective Workflow Scheduling (MOWS) scheme has been proposed by authors in [10] to address the issues of information leaks caused by intermediate data security or data alteration in the environment. A model has been formulated for task scheduling followed by which a heuristic algorithm has been proposed for completion time and security requirements. A VM schedule management system has been proposed for smart grid applications based on the size of solar panels by the authors in [11] to reduce brown energy consumption as much as possible.

In this paper, we have integrated predictive analytics with machine learning-driven dynamic placement, which forms a cohesive approach for efficient resource allocation. The

Google Cluster-Usage Traces Dataset was used for experimentation. Finally, the static allocation method has been compared with our proposed approach with respect to average resource allocation, cost saving & response time.

The paper is organized as follows. Section II gives the methodology, Section III provides the dataset and experimental setup, The implementation is shown in section IV, section V gives the results and discussion and finally the conclusion and future work in this direction is provided in section VI.

II. METHODOLOGY

The proposed method consists of two stages – proactive predictive analysis and ML-driven placement of VM. These are discussed below in detail.

A. Proactive Predictive Analysis

In this stage, we collect data and preprocess that. This helps us acquire the usage patterns and workload characteristics. Forecasting based on deterministic or probabilistic methods is performed, or even we can use LSTM architectures to do that. After training, this model can be employed to forecast future resource demands based on this data.

B. ML-Driven Dynamic VM Placement

This utilizes a carefully selected set of attributes that are derived from real-time workloads, historical data & performance metrics. These features serve as inputs to various ML models which are trained to predict the optimal placement of VMs.

The stages discussed above in subsections A and B above, when integrated together form a cohesive approach for optimal allocation of resources. In Stage A, we get valuable insights into future resource requirements which allow the system in Stage B to allocate resources with respect to varying workloads. In addition to that, ML models adapt to evolving workload patterns which ensure dynamic placement decisions for real-time variations.

This whole system optimizes the allocation of VMs and minimizes under-utilization & also over-provisioning. In this way, predictive analytics informs the ML models and vice versa, it ensures a balance between proactive planning & responsiveness to real-time demands, in turn enhancing the efficiency of resource allocation in cloud environments.

In this work, we aim to work towards resource allocation, cost saving, and response time. The proposed architecture scheme is shown in Fig. 1.

III. DATASET & EXPERIMENTAL SETUP

For experimentation purposes, a representative dataset is required that encompasses a variety of workload patterns and essential resource demands. We need information regarding job execution, resource consumption, and task characteristics. We need some valuable information regarding the dynamics of cloud workloads to simulate a real-time cloud environment.

We used the Google Cluster-Usage traces dataset, which comprises job events and resource consumption from Google's data Centers. It includes information such as task start and end times, resource usage, and task priorities covering diverse workloads from batch processing to interactive tasks, allowing comprehensive experimentation. The dataset size is 2.4 TiB ≈ 2.4 × 1.1 TB ≈ 2.64 TB. The model description has been discussed in the next section.

IV. IMPLEMENTATION

Having fixed the schema for VM placement as mentioned above in Section II, the next task is to identify the techniques to be employed for Stages 1 and 2. This technique might differ based on the data distribution of the dataset being used.

The predictive analysis is usually a univariate analysis with respect to the number of virtual systems being required. In case other supporting attributes correlating with the use of the VM counts are available, one can go ahead with an MVA algorithm. But this is not the case which we are dealing with. Usually, probabilistic models like ARMA, ARIMA, etc., or sometimes even deterministic models like HW smoothening can be used for this. These methods have been implemented in [12] and can be used as it is just by changing the dataset and preprocessing them.

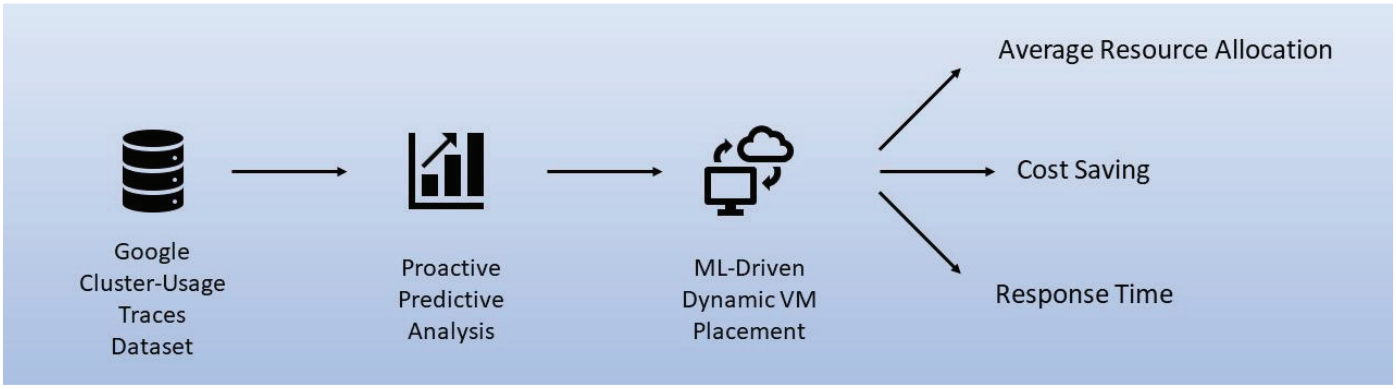

Fig. 1. Proposed Scheme

For the placement task in stage two, we generally employ any classical classification model. It is important to note that since the dimensionality and complexity of data in our case are not too high, we used a simple SVM [13] model for this task. Based on the dimensionality of the data in hand, the complexity may change, and we might need to cross-validate, fine-tune, or even change the model accordingly.

In our case, after enough cross-validations, we used the classical triple-exponential smoothing to predict the count of machines that would be required at a given point in time, followed by which we used a simple SVM model to predict the placement of machines in the system.

V. RESULTS & DISCUSSION

We've compared the output metrics based on three types of allocations – random allocation, static allocation, and ML-driven allocation. Table 1 shows the resource utilization across the three allocation strategies.

TABLE I. RESOURCE UTILIZATION COMPARISON

Allocation Strategy	Average resource Utilization
Random Allocation	68.7 %
Static Allocation	65.2 %
ML-Driven Allocation	**82.3 %**

Table 1 shows that the ML-driven allocation outperforms both static and random allocations with a utilization rate of 82.3%. Hence the ML-driven approach is more effective in adapting to workload changes and utilizing allocated resources. The static allocation is unable to accommodate dynamic workload patterns, hence resulting in lower utilization. Random placement while slightly better, still lacks the intelligence provided by predictive analytics and ML-driven decision which leads to suboptimal resource allocation.

TABLE II. COST SAVINGS ANALYSIS

Allocation Strategy	Cost Savings
Random Allocation	8.2 %
Static Allocation	12.6 %
ML-Driven Allocation	**20.9 %**

Table 2 shows the cost-saving analysis, hence revealing the financial benefits of the ML-driven model. It achieves a remarkable cost saving of 20.9%, compared to static allocation. This shows the ability to allocate resources efficiently according to workload demands. Random and static allocations exhibit lower cost savings due to their lack of adaptability to dynamic situations. Hence, we see that the proposed ML-driven approach not only optimizes resource utilization but also minimizes unnecessary expenditure which makes it a cost-effective solution for cloud resource allocation.

The performance of these allocation strategies in terms of average response time has been mentioned in Table 3. The ML-driven approach shows the lowest response time of 180 ms, which indicates its effectiveness in the delivery of efficient task execution. Static and random allocation strategies take a longer time to respond as they are unable to adapt to variations in workload. This diminished response time of our proposed approach enhances user experience and system efficiency. This makes it valuable for real-time applications and workloads with stringent latency requirements.

TABLE III. RESPONSE TIME PERFORMANCE

Allocation Strategy	Average Response Time
Random Allocation	220 ms
Static Allocation	250 ms
ML-Driven Allocation	**180 ms**

In addition to the above three comparisons, we move ahead with a few other such comparisons and comprehensively compare the CPU and memory utilization.

TABLE IV. COMPREHENSIVE RESOURCE UTILIZATION COMPARISON

Allocation Strategy	Average CPU Utilization	Average Memory Utilization
Random Allocation	63.5 %	72.4 %
Static Allocation	60.2 %	70.8 %
ML-Driven Allocation	**75.1 %**	**78.6 %**

Table 4 provides a detailed comparison of resource CPU and memory utilization. The proposed ML-driven approach consistently outperforms the rest of the two approaches with an average CPU utilization of 75.1% and memory utilization of 78.6%. These two values themselves justify the ML-driven approach's capability to optimize the allocations of both CPU and memory resources. Static allocation falls short in allocating resources efficiently, resulting in lower utilization as indicated in Table 1. Random allocation, while slightly better again lacks the intelligence of the ML-driven approach leading to suboptimal resource allocation, again justifying Table 1.

Table 5 presents a comprehensive analysis of energy efficiency achieved by different allocation strategies. In correlation with Table 2, the proposed approach significantly contributes to energy efficiency by saving 410 kWh of energy compared to static allocation. This approach's ability to allocate resources dynamically according to workload demands minimizes resource wastage, hence causing both environmental and financial benefits. Random and static allocation exhibit lower energy efficiency due to a lack of adaptability.

TABLE V. ENERGY EFFICIENCY ANALYSIS

Allocation Strategy	Energy Efficiency (Saved)
Random Allocation	175 kWh
Static Allocation	245 kWh
ML-Driven Allocation	**410 kWh**

We also measured the response times and throughput by varying the workloads, as mentioned in Table 6.

TABLE VI. PERFORMANCE METRICS FOR WORKLOAD TYPES

Workload Type	Allocation Strategy	Average Response Time	Throughput (Tasks/min)
Batch	Random Allocation	290 ms	210
	Static Allocation	320 ms	180
	ML-Driven Allocation	**220 ms**	**300**
Interactive	Random Allocation	210 ms	350
	Static Allocation	180 ms	400
	ML-Driven Allocation	**150 ms**	**480**

From Table 6 above, we see that for both batch and interactive workloads the ML-driven technique yields improved response time and throughput compared to static and random strategies. This shows its adaptability to various workload characteristics. The lower response times and higher throughputs enhance the efficiency of the ML-driven model ensuring optimal performance for diverse workloads.

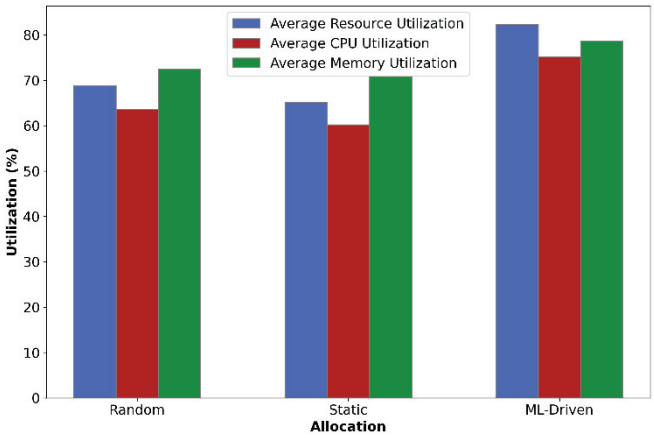

Fig. 2. Comprehensive Resource Utilization

Fig. 2 shows the graphical representation of the average resource utilization, average CPU utilization, and average memory utilization for the three allocation strategies. Fig. 3 shows the throughputs corresponding to batch and interactive workloads for the three allocation strategies. Fig. 4 shows the Average Response Times corresponding to batch and interactive workloads for the three allocation strategies.

The values in Tables 2 and 5 are not directly in proportion but are strongly correlated. This is because the cost is not just dependent on energy consumption but many other factors such as resource utilization, memory allocation, etc.

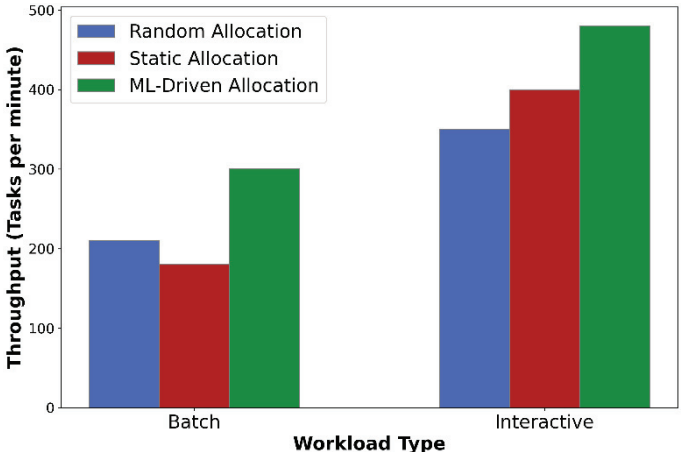

Fig 3. Throughputs

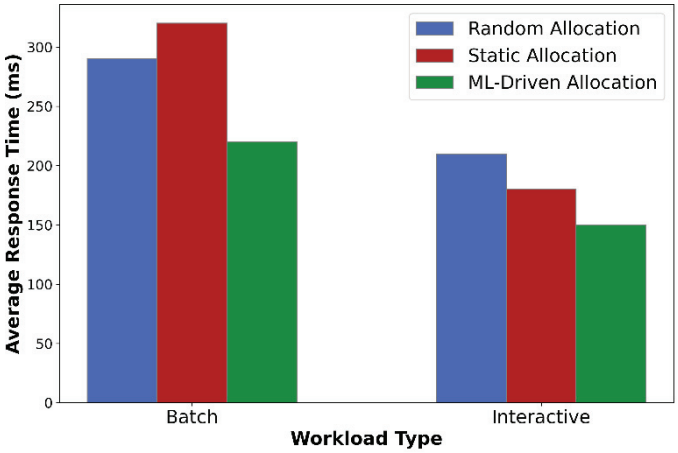

Fig. 4. Average Response Times

VI. CONCLUSION & FUTURE SCOPE

After looking at the outputs obtained after using the proposed ML-driven approach in the traces dataset, we acquired satisfactory results compared to the rest allocation strategies in all aspects whether be it in terms of energy saving, cost cutting, utilization, throughput, or response time. Practically it is impossible to meet the ideal conditions, but one always aims to optimize the model as much as possible.

Most of the classical time-series models were tested for stage 1, and we finally decided to proceed with triple-exponential smoothing. Exploring neural network-based models or even training pre-built models by transfer learning in our context may or may not yield more efficient results in stage 2. Based on the prebuilt models available, we decided to use SVM in this paper. Exploration of other advanced models will be a topic of discussion in our subsequent works.

REFERENCES

[1] J. Gao and G. Tang, "Virtual Machine Placement Strategy Research", International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, Beijing, China, 2013, pp. 294-297, 2013.

[2] S.M. Seyyedsalehi, & M. Khansari, "Virtual Machine Placement Optimization for Big Data Applications in Cloud Computing", IEEE Access, 10, pp. 96112-96127, 2022.

[3] W. Zhang, X. Chen, & J. Jiang," A multi-objective optimization method of initial virtual machine fault-tolerant placement for star topological data centers of cloud systems", Tsinghua Science and Technology, 26(1), pp. 95-111, 2020.

[4] A. Zhou, S. Wang, B. Cheng, Z. Zheng, F. Yang, R.N. Chang, & R. Buyya,"Cloud service reliability enhancement via virtual machine placement optimization", IEEE Transactions on Services Computing, 10(6), pp.902-913, 2016.

[5] S. Mejahed, M. Elshrkawey, "A multi-objective algorithm for virtual machine placement in cloud environments using a hybrid of particle swarm optimization and flower pollination optimization", PeerJ Computer Science, 8, e834, 2022.

[6] U.C. De, R. Satapathy, & S.S.Patra, " Cost Analysis and Optimization of Virtual Machine Allocation in the Cloud Data Center", International Conference on Inventive Computation Technologies (ICICT) ,pp. 809-813, IEEE, 2023.

[7] Sudhakar, & Saravanan. September, "A Survey and Future Studies of Virtual Machine Placement Approaches in Cloud Computing Environment", 6th International Conference on Cloud Computing and Internet of Things ,pp. 15-21, 2021.

[8] M.C. Silva Filho, C. C. Monteiro, P.R. Inácio & M.M. Freire, "Approaches for optimizing virtual machine placement and migration in cloud environments", A survey. Journal of Parallel and Distributed Computing, 111, pp.222-250, 2018s.

[9] A. Alashaikh, E. Alanazi, & A. Al-Fuqaha, "A survey on the use of preferences for virtual machine placement in cloud data centers", ACM Computing Surveys (CSUR), 54(5), pp.1-39, 2012.

[10] Abazari Farzaneh, Analoui Morteza, Hassan Takabi, Song Fu, "MOWS: Multi-objective workflow scheduling in cloud computing based on heuristic algorithm, Simulation Modelling Practice and Theory", Volume 93, pp.119-132, 2019.

[11] Inès De Courchelle, Tom Guérout, Georges Da Costa, Thierry Monteil, Yann Labit, "Green energy efficient scheduling management", Simulation Modelling Practice and Theory, Volume 93,pp.208-232, 2019.

[12] S. C. Banerjee, S. Banerjee and R. K. Jain, "Vaccine Supply Forecasting and Optimization using Deterministic and Probabilistic Approaches," 2nd International Conference for Innovation in Technology (INOCON), pp. 1-5, 2023,

[13] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt and B. Scholkopf, "Support vector machines," in IEEE Intelligent Systems and their Applications, vol. 13, no. 4, pp. 18-28,1998.

[14] B. B. Dash, R. Satapathy and S. S. Patra, "SDN-Assisted Routing Scheme in Cloud Data Center using Queueing Vacation Policy," 2023 2nd International Conference on Edge Computing and Applications (ICECAA), pp. 1-6,2023.

[15] B. B. Dash, R. Satapathy and S. S. Patra, "Energy Efficient SDN-assisted Routing Scheme in Cloud Data Center," 2023 2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies (ViTECoN), pp. 1-5, 2023.

[16] S. Behera, N. Panda, U. C. De, B. B. Dash, B. Dash and S. S. Patra, "A task offloading scheme with Queue Dependent VM in fog Center," 2023 6th International Conference on Information Systems and Computer Networks (ISCON), pp. 1-5, 2023.

[17] B. B. Dash, S. S. Patra, R. Satpathy and B. Dash, "Improvement of SDN-based Task Offloading using Golden Jackal Optimization in Fog Center," 2023 World Conference on Communication & Computing (WCONF),pp.1-6,2023.